

Руководство по Flussonic

2020-02-21

Быстрый старт

Live — потоковое вещание

Варианты источников

Переключение источников

Публикация видео на сервер

Прием мультикаста

Серверные плейлисты

Миксер

Decklink SDI

Отправка потока на другие серверы

Субтитры

Наложение логотипа

Проигрывание

Воспроизведение HLS

Вставка видео на сайт (embed.html)

HTML5 (MSE-LD) воспроизведение с низкой задержкой

MSE Player

Рассылка мультикаста

Рассылка UDP с постоянным битрейтом (CBR)

Использование WebRTC для проигрывания видео с Flussonic Media Server

Воспроизведение H265

Транскодер

Аппаратное транскодирование на Nvidia NVENC

Intel Quick Sync Video

Скриншоты

Мозаика

VOD — вещание файлов

Кэш

Облачное хранение

Транскодирование файлов

Плагин Dynafile

Мультибитрейтный плейлист из файлов

Кластер
Ретрансляция потоков
Кластерный захват потоков
Балансировщик нагрузки
Пиринг
Организация CDN
Запись видеопотоков (Digital Video Recording)
Настройка DVR (Digital Video Recording)
Таймшифт в другой часовой пояс
Запись передач (Catchup TV)
Проигрывание архива
Экспорт в mp4
Доступ к DVR по различным протоколам
Timelapse
DVR API
Кластеризация DVR
Репликация
DVR в облаке
Flussonic RAID для DVR
Авторизация
Конфигуратор бэкендов
Сервис статистики
Domain lock
Авторизация в Flussonic через Middleware
Защита доступа к потокам (Авторизация по токену)
Ограничение количества сессий на пользователя (защита от кражи)
Как настроить два авторизационных бекенда
Как ограничить доступ по IP адресам
Авторизация доступа к архиву
Псевдонимы для имен потоков
HTTP API к Flussonic Media Server
Events API
SQL API
Управление кластерами при помощи SQL API
SNMP
Защита контента с помощью DRM
Conax DRM

BuyDRM KeyOS
Widevine DRM
PallyCon DRM
EzDRM
Веб-интерфейс
Инсталляция
Обновление Flussonic
Конфигурирование Flussonic Media Server
Мониторинг
Оптимизация Flussonic Media Server и операционной системы
Использование лицензионного ключа
Скрипты на Lua
Безопасность Flussonic Media Server
Let's Encrypt и Flussonic Media Server
Миграция
Захват спутникового видео
Транскодирование
Middleware в IPTV OTT
Получение EPG из MPEG-TS потоков
Группы потоков
Врезка рекламы
How-to руководства

Быстрый старт

Эта статья познакомит вас с Flussonic Media Server. Прочитав ее, вы сумеете:

- Установить Flussonic Media Server

- Загрузить и проиграть файл

- Настроить и посмотреть поток

- Опубликовать видео на сервере

Чтобы попробовать Flussonic Media Server, нужен компьютер с Linux, подключенный к Интернету, и лицензионный либо триальный ключ. Триальный ключ можно запросить на этой странице

Мы рекомендуем операционные системы Debian или Ubuntu, но можно попробовать и CentOS. Основное требование — 64-разрядная операционная система. Если Линукса или свободного сервера под рукой нет, то можно арендовать сервер в Digital Ocean на время, чтобы попробовать Flussonic Media Server.

В результате вам надо иметь доступ к консоли Linux под пользователем root.

Замечание. Далее в документации мы будем приводить IP адрес и URL Flussonic Media Server (например, FLUSSONIC-IP). Вам необходимо заменять их на реальные IP адрес или URL вашего сервера.

Основные решения Flussonic

Flussonic Media Server

Flussonic Media Server — это серверное программное обеспечение для видео стриминга, способное решать широкий ряд задач от захвата, транскодирования, записи архива и мультипротокольной раздачи видеоконтента (live и on-demand) по всему миру, до управления потреблением контента и видео потоками.

Flussonic Watcher

Flussonic Watcher — это отдельный продукт, который используется вместе с Flussonic Media Server для организации систем видеонаблюдения. Это пользовательский интерфейс к серверу Flussonic, созданный для удобной работы с камерами, абонентами и событиями.

Flussonic Watcher работает только вместе с Flussonic Media Server. О том, как установить Watcher, рассказано в документации на Flussonic Watcher.

Установка Flussonic Media Server

Здесь приводится краткое описание установки, достаточное для быстрого старта Flussonic. Более подробно об установке Flussonic Media Server читайте в разделе про установку.

Чтобы установить Flussonic, выполните в командной строке Linux команду:

```
curl -sSf https://flussonic.com/public/install.sh | sh
```

Затем запустите Flussonic Media Server:

```
/etc/init.d/flussonic start
```

Теперь откройте в браузере веб-интерфейс администратора.

Первое открытие веб-интерфейса Flussonic (UI)

Веб-интерфейс Flussonic доступен по адресу `http://FLUSSONIC-IP:8080/` (замените FLUSSONIC-IP на адрес вашего сервера).

Логин и пароль по умолчанию: `flussonic` и `letmein!`

На стартовой странице `http://FLUSSONIC-IP:8080/` Flussonic просит ввести полученный ключ.

На этой же странице необходимо сразу же изменить пароль администратора для управления Flussonic Media Server.

Замечание. Лицензионный ключ хранится в файле `/etc/flussonic/license.txt`, его можно вписать туда вручную перед запуском Flussonic, тогда его не нужно будет указывать в веб-интерфейсе.

Проверка установки

Проверить правильность установки Flussonic Media Server можно по адресу `http://FLUSSONIC-IP:8080/`, где FLUSSONIC-IP — адрес того сервера, на который вы поставили ПО. Откроется главная страница веб-интерфейса к Flussonic.

Другой способ проверить правильность установки — выполнить команду:

```
/etc/init.d/flussonic status
```

Ответ системы должен быть похож на такой:

```
# /etc/init.d/flussonic status
```

```
Flussonic 19.05 is running with streams:
```

```
...
```

Проигрывание файлов

Попробуем проиграть файл через Flussonic Media Server. Для этого необходимо:

- настроить место хранения файлов

- загрузить файл

- проиграть файл

Настройка места хранения файлов

Flussonic Media Server не диктует жестко, в каком месте на диске должны находиться ваши файлы. Более того, файлы могут находиться в разных местах на диске, поэтому надо явно указать, откуда брать файлы.

Для этого Flussonic Media Server надо сообщить, какой путь в запросах на проигрывание файла будет соответствовать файлу на диске или в HTTP хранилище. Исходя из этих настроек, Flussonic сам формирует путь к файлу – этот путь используется в запросах на проигрывание файла по разным протоколам. Например, URL для проигрывания по протоколу HLS – `http://FLUSSONIC-IP:80/vod/bunny.mp4/index.m3u8`.

В веб-интерфейсе можно посмотреть готовые URL адреса для запроса файла. Их можно копировать и использовать для проигрывания файла.

Чтобы указать, откуда брать файлы, в конфигурационном файле `/etc/flussonic/flussonic.conf`

добавьте строки:

```
file vod {  
    path /storage;  
}
```

Теперь Flussonic Media Server будет знать, что при обращении к `vod/movies/bunny.mp4` надо будет взять файл `/storage/movies/bunny.mp4`. Другими словами, всё после совпавшего префикса `vod` будет отрезано и "подклеено" к указанному пути на диске.

Примените новые настройки, выполнив команду:

```
/etc/init.d/flussonic reload
```

Другой способ сконфигурировать доступ к файлам — через веб-интерфейс:

Откройте Media > Files (VOD) > кликните add > введите VOD name (vod) и File directory path (`/storage`) > кликните create.

Загрузка файла на сервер

Теперь можете закачать файл, который вы уже подготовили или скачали подготовленным, в каталог `/storage`.

На сервере есть готовый файл `/opt/flussonic/priv/bunny.mp4`.

Если файла ещё нет, можете взять свободно доступный Big Buck Bunny:

```
mkdir -p /storage  
cd /storage  
curl -o bunny.mp4 http://download.blender.org/peach/bigbuckbunny_movies/big_buck_bunny_480p_h264.mov
```

Проигрывание

Теперь зайдём на `http://FLUSSONIC-IP:8080/vod/bunny.mp4/embed.html` и посмотрим видео.

Подробнее о файлах см. в разделе про работу с видеофайлами

Прямой эфир

Flussonic Media Server может получать потоковое видео двумя основными способами: выступая в роли клиента или сервера.

В первом случае Flussonic Media Server сам обращается к источнику для получения с него видео. Во втором – ожидает подключения, чтобы принять видео для публикации.

Источником видео может быть видеокамера, другой видеостриминговый сервер, специализированная программа, работающая с DVB-картой, и вообще любая программа, умеющая передавать видео по сети. Flussonic поддерживает все основные протоколы передачи видео.

Также Flussonic Media Server может сам генерировать поток fake, который можно использовать, например, для проверки работы сервера.

Откроем конфигурационный файл `/etc/flussonic/flussonic.conf` и добавим туда описание потока:

```
stream demo {  
    url fake://fake;  
}
```

Здесь:

stream — ключевое слово, за ним идёт имя потока: demo. Далее в фигурных скобках описание параметров потока.

url — источник видео задаётся с помощью параметра url.

fake://fake — специальный адрес, по которому можно получить демонстрационный поток видео — цифровые часы на сером фоне.

После того, как мы изменили конфигурационный файл, необходимо применить настройки. Выполним команду:

```
/etc/init.d/flussonic reload
```

Увидеть результат можно открыв страницу: <http://FLUSSONIC-IP:8080/demo/embed.html>.

Подробнее о прямом эфире см. в разделе о потоковом вещании.

Публикация видео

Публикацией называется ситуация, когда к Flussonic Media Server подключается другая программа и иницирует передачу ему потокового видео. Чтобы это было возможно, в Flussonic Media Server должно быть сконфигурировано место на сервере, в которое разрешена публикация.

Это можно сделать двумя способами.

Публикация в поток с известным именем

Для того, чтобы разрешить публикацию в отдельный конкретный поток, используется опция url publish://:

```
stream publishdemo {  
    url publish://;  
}
```

Имя потока (publishdemo) заранее известно серверу Flussonic, так как вы сконфигурировали поток с этим именем.

В этом случае публиковать надо в URL:

```
rtmp://FLUSSONIC-IP:1935/static/publishdemo
```

а смотреть можно на странице:

```
http://FLUSSONIC-IP:8080/publishdemo/embed.html
```

Публикация в поток с неизвестным именем по префиксу

Вам нужно будет настроить место публикации (publishing location, или префикс) на сервере Flussonic, куда вы разрешите публикацию потоков. В одно место публикации может быть опубликовано несколько потоков.

Директива live задаёт префикс (в примере: mylive), разрешая публикацию в любой поток с таким префиксом в адресе:

```
live mylive {  
}
```

Префикс mylive добавлен в конфигурационном файле по умолчанию после установки.

Проверьте его наличие в /etc/flussonic/flussonic.conf. Если такой директивы нет в вашем файле, добавьте ее и примените настройки:

```
/etc/init.d/flussonic reload
```

Опубликовать видео можно с помощью утилиты, которая устанавливается в пакете rtmp_push и находится в директории /opt/flussonic/contrib. Для публикации файла по RTMP в префикс mylive используйте синтаксис:

```
/opt/flussonic/contrib/rtmp_push FILE.mp4 rtmp://FLUSSONIC-IP:1935/mylive/STREAM_NAME
```

Пример

Для передачи используем протокол RTMP, источником видео будет файл bunny.mp4 (о файле см. выше в разделе про прямой эфир):

```
/opt/flussonic/contrib/rtmp_push /opt/flussonic/priv/bunny.mp4 rtmp://127.0.0.1/mylive/bunny
```

Просмотреть публикуемое видео можно в браузере:

```
http://FLUSSONIC-IP:8080/mylive/bunny/embed.html
```

Важно. Если вы указали в конфигурации префикс публикации mylive, то в URL вы должны указывать имя потока, начинающееся с mylive/, например, mylive/bunny. Имя потока, которое идет после префикса, задаёт клиентское приложение, которое публикует видео. В конфигурации Flussonic Media Server имя потока не указывается.

Подробнее о публикации можно узнать в разделе Публикация видео.

Live — потоковое вещание

Flussonic Media Server умеет ретранслировать потоковое видео, перепаковывая его на лету в разные форматы. Это означает, что вы можете захватить MPEG-TS поток и раздавать его одновременно тысячам получателей, например, в DASH или HLS, и публиковать в RTMP на YouTube.

Flussonic Media Server поддерживает три типа потоков:

- static — постоянно живущие (транслируемые).
- ondemand — потоки, транслируемые по запросу.
- live – публикуемые пользователем. См. Публикация

Содержание:

- Статические потоки
- Потоки по запросу (ondemand)
- Проигрывание потоков
- Мультибитрейтный поток
- Скриншоты потока
- Заглушка для потока
- Подстановки
- Запись потоков (DVR)
- Сдвиг по часовой зоне (Timeshift)
- Выдача потока в UDP multicast
- Настройки потоков для IP камер наблюдения
- Включение audio-only варианта HLS
- Захват потока с другого сервера Flussonic Media Server
- DRM в live-потоках
- Обнаружение тишины в потоке
- Настройки потока или группы

Статические потоки

Статические потоки запускаются при старте сервера и Flussonic Media Server постоянно следит за ними. Если источник пропадает (выключился транскодер, авария на антенне), то Flussonic Media Server будет пытаться переподключиться к источнику и ни при каких обстоятельствах не перестанет этого делать.

IPTV канал или IP камера обычно объявляются именно как статический поток.

Flussonic Media Server поддерживает различные типы источников, которые указываются в виде URL-адресов.

Пример конфигурации потоков из /etc/flussonic/flussonic.conf:

```
stream ort {  
    url tshttp://10.0.4.5:9000/stream;
```

```
}  
stream ipcam {  
    url rtsp://192.168.0.100/channel/101;  
}
```

В этой конфигурации:

ort и ipcam — это имена потоков, по которым можно обратиться к Flussonic Media Server и получить их.

tshhttp://10.0.4.5:9000/stream и rtsp://192.168.0.1/channel/101 — URL источников.

Важно. Имя канала должно состоять из латинских букв, чисел, точки (.), символов минуса (-) и нижнего подчеркивания (_). Если в имени будет что-то кроме этих символов, то работоспособность DVR и вещания мы гарантировать не можем.

Чтобы добавить поток через веб-интерфейс:

Перейдите на вкладку Media.

Нажмите add рядом со Streams.

Введите название потока и URL адрес источника. После этого нажмите create.

По умолчанию создается статический поток. Чтобы поменять тип потока на on demand, кликните по static рядом с названием потока.

Поток создан. Теперь можно перейти на страницу потока для проверки захвата:

Потоки по запросу (ondemand)

Если поток нужен не всё время, а только по требованию пользователя, можно указать Flussonic Media Server отключать его при неиспользовании и включать по запросу.

Для указания такого типа потока надо заменить stream на ondemand:

```
ondemand ipcam {  
    url rtsp://192.168.0.1/channel/101;  
}
```

Важно. Если между источником и репитером используется RTMP, RTSP или HTTP MPEG-TS, то снимать с репитера HDS или HLS не получится, потому что эти протоколы требуют наличия 10-30 секундного буфера видео в памяти. Плеер не начнет проигрывание пока этот буфер не накопится, и, следовательно, первый пришедший пользователь будет ждать это время. Единственный источник, который не подвержен этой проблеме – это другой Flussonic Media Server по протоколу HLS. Во Flussonic Media Server используются собственные расширения, позволяющие моментально стартовать видео на iPhone.

Можно регулировать время жизни потока после отключения клиента:

```
ondemand cam1 {  
    url rtsp://192.168.0.1/channel/101;  
    retry_limit 10;  
    client_timeout 20;  
}
```

В конфигурации выше указано следующее: пытаться переподключиться к сбойному источнику не больше 10 раз и после ухода последнего клиента гонять поток вхолостую не дольше 20 секунд.

Проигрывание потоков

Проигрывание описано подробно в отдельном разделе.

Мультибитрейтный поток

Для объединения двух независимых источников одного и того же контента для создания мультибитрейтного потока используется опция

```
mbr=1
```

.

Подробнее в разделе про объединение источников в один поток

JPEG-скриншоты потока

Flussonic Media Server может делать JPEG-скриншоты потока. Для этого надо указать опцию `thumbnails` в настройках потока:

```
stream ort {  
  url udp://239.255.0.100:1234;  
  thumbnails;  
}
```

А можно указать Flussonic Media Server, где забирать скриншоты. Это поможет не тратить ресурсы CPU. Многие IP-камеры имеют специальный URL со скриншотами:

```
stream cam0 {  
  url rtsp://10.0.4.3:554/h264;  
  thumbnails http://10.0.4.3/cgi-bin/snapshot.cgi  
}
```

URL с потоком скриншотов можно найти в документации к вашей модели камеры.

Последний скриншот потока доступен по адресу <http://flussonic:8080/ort/preview.jpg>

MJPEG поток скриншотов доступен по адресу <http://flussonic:8080/ort/preview.mjpeg>

Читайте также:

Скриншоты о JPEG-скриншотах.

Видео-скриншоты об MP4-скриншотах, экономящих ресурсы.

Файл-заглушка для потока

Если поток недоступен, можно запустить на проигрывание файл — резервный источник потока. Это можно делать для любых live-потоков, включая публикуемые.

```
stream rtr {  
  url tshttp://10.0.4.5:9000/channel/5;  
  backup vod/backup.mp4;  
}
```

Необходимо указывать путь к файлу-заглушке относительно VOD-локации, например vod/backup.mp4, при этом нельзя указывать абсолютный путь.

Важно. Если оригинальный поток идет без звука (например, с IP камеры), то и файл-заглушка тоже должен быть без звука.

По умолчанию, файл-заглушка не пишется в архив и не транскодируется.

Опции файла-заглушки

dvr=true — если основной поток пишется в архив, то заглушка будет тоже записываться в архив, если указать эту опцию:

```
stream backup_dvr {  
  backup vod/backup-file.mp4 dvr=true;  
  dvr movies;  
}
```

timeout=10 — через какое время (в секундах) Flussonic переключится на заглушку, если от основного источника видеопотока перестанут поступать кадры. Важно, что источник при этом остается запущенным, что позволяет, например, оставаться на сожете клиенту-публикатору.

Если не указывать timeout специально для заглушки, то в случае отсутствия кадров будет использоваться source_timeout основного источника.

Комбинируя настройки timeout и source_timeout, можно:

задать более долгое время для того, чтобы мобильные клиенты успели начать публикацию и не были отключены
при этом переключаться за заглушку как можно скорее.

Пример:

```
stream backup_timeout {  
  url publish:// source_timeout=20;  
  url fake://fake;  
  backup vod/backup-file.mp4 timeout=1;  
}
```

В этом примере:

До начала публикации будет воспроизводиться поток fake. Затем подключается клиентское приложение, которое будет публиковать видео на Flussonic. Если после подключения от клиента-публикатора потока так и не пришли кадры в течение 20 секунд, то клиент принудительно отключается и начинает проигрываться fake.

После начала публикации файл-заглушка backup-file.mp4 начнет проигрываться, если в течение 1 секунды не поступит ни одного кадра от публикуемого потока. Источник публикации при этом пока не отключается.

При возобновлении кадров от источника поток переключается на клиента-публикатора. Если же в течение 20 секунда кадры не приходят, то публикатор принудительно отключается и начинает проигрываться fake.

задать более долгое время для того, чтобы мобильные клиенты успели начать публикацию и не были отключены

при этом переключаться за заглушку как можно скорее.

Подстановки

Иногда заранее неизвестны имена потоков на удаленном сервере, и хочется получать все эти потоки. В этом случае используется специальный тип потока (rewrite):

```
rewrite nsk/* {  
    url rtsp://nsk-origin:554/%s;  
}  
  
rewrite ams/* {  
    url hls://ams-origin:8080/%s/index.m3u8;  
}
```

Если опция rewrite сочетается с использованием звездочки на конце имени потока, то префикс перед звездочкой отрезается от имени потока и результат подставляется в URL на место "%s".

Запись потоков (DVR)

Во Flussonic Media Server встроена система записи потоков в так называемый архив. Архиватор потоков умеет записывать видео, предоставлять доступ к произвольному фрагменту, экспортировать части архива в виде MP4 файлов, очищать старые файлы и поддерживать заполнение хранилища на приемлемом уровне.

Для включения архива достаточно указать опцию dvr в конфиге потока:

```
stream foxlive {  
    url tshttp://trancoder-5:9000/;  
    dvr /storage 90% 5d;  
}
```

Подробнее в разделе про архив и методов работы с ним.

Сдвиг по часовой зоне (Timeshift)

Flussonic Media Server умеет воспроизводить записанный в архив поток с фиксированным отставанием.

Важно! Flussonic Media Server очень четко поддерживает фиксированное отставание. Если в архиве были "дырки", то пользователи не получают никакого видео за это время.

Для таймшифта есть отдельный протокол источника:

```
stream ort {  
    url tshttp://trancoder-5:9000/;  
    dvr /storage 90% 5d;  
}
```

```
stream ort-4h {  
  url timeshift://ort/14400;  
}
```

Отставание указывается в секундах.

Выдача потока в UDP multicast

Flussonic Media Server умеет ретранслировать поток из источника в локальную сеть.

Flussonic Media Server пытается выдавать UDP максимально монотонно во времени, чтобы не создавать скачкообразной нагрузки на сеть.

```
stream ort {  
  url tshttp://trancoder-5:9000/;  
  udp 239.0.4.4:1234;  
}
```

Настройки потоков для IP-камер видеонаблюдения

Можно указать Flussonic Media Server запрашивать поток с камеры только по UDP. Это бывает нужно с камерами, в которых проблемная реализация TCP.

```
stream cam1 {  
  url rtsp://10.0.4.3:554/h264 rtp=udp;  
}
```

Важно! Обратите внимание, что из десктопных браузеров показывать H265 сейчас фактически умеет только Microsoft Edge (версия 16 и выше) и Safari (версия 11 и выше). Из мобильных браузеров — Safari и Chrome для iOS (версия 11.0 и выше). Подробнее в статье [Воспроизведение H265](#).

Если с камеры не надо забирать звук (например, он в G.726) то можно указать Flussonic Media Server забирать только одну дорожку. Ее номер необходимо указать в конфигурации:

```
stream cam1 {  
  url rtsp://10.0.4.3:554/h264 tracks=1;  
}
```

Для того, чтобы автоматически транскодировать аудио с камеры из G.711a или G.711u в AAC, укажите другой протокол:

```
stream cam1 {  
  url rtsp2://10.0.4.3:554/h264;  
}
```

Включение audio-only варианта HLS

Apple при валидации программ в AppStore может потребовать, чтобы поток был с audio only вариантом. Если добавить директиву в конфигурацию:

```
stream cam1 {  
    url rtsp://10.0.4.3:554/h264;  
    add_audio_only;  
}
```

и при этом в потоке есть и видео, и аудио, то Flussonic Media Server будет генерировать мультибитрейтный вариантный плейлист из двух потоков: один обычный, второй только со звуком.

Захват потока с другого сервера Flussonic Media Server

Детально вопросы передачи видео между разными Flussonic Media Server описаны в разделе про кластеризацию видеопотоков в Flussonic.

DRM в live-потоках

Детально вопрос использования таких DRM как AES-128, SAMPLE-AES и Conax описан в разделе про DRM.

Обнаружение тишины в потоке

Flussonic умеет обнаруживать низкий уровень звука (отсутствие звука) на источнике входного потока и оповещать об этом. Подробнее в разделе

Обнаружение тишины

Настройки потока или группы

Эти настройки можно указывать внутри директив stream, ondemand, rewrite и live. Мы называем их опции.

auth

auth http://backend/; Включение авторизации для потока. Более подробно в разделе про авторизацию.

domains

domains host1.ru *.host1.ru; Указание доменов, на которых можно проигрывать это видео. Не работает для клиентов, которые не передают Referer. Для правильной работы в WEB в списке доменов должен присутствовать домен, с которого загружается embed плеер.

allowed_countries

allowed_countries RU UA KZ; Список стран (двухбуквенные коды как в БД maxmind) с которых разрешен доступ.

url

url tshttp://transcoder:port/; Адрес источника. Можно перечислить несколько директив url для выбора источника по кругу.

Важно! Если используется UDP-источник, то в файле конфигурации этот UDP-адрес должен

использоваться только один раз. Если несколько потоков будут использовать один и тот же адрес, то, скорее всего, это просто не будет работать.

urls

`urls source1 source2;` Список адресов источников. Подробнее про механизм переключения источников.

url_prefix

`url_prefix prefix` При использовании протокола HLS, адреса отдельных сегментов и плейлистов в вариантном плейлисте будут начинаться с указанного префикса. Эта опция может быть использована не только как часть настроек конкретного потока, но и в глобальной части файла конфигурации. Прописанная глобально опция применяется ко всем потокам на этом сервере. Пример: `url_prefix http://my.domain.address.com:8080`

dvr

`dvr /storage 1d 50% schedule=8:00-16:00;` `dvr @my_raid 1d 50% schedule=8:00-16:00;`
Включение архива. Flussonic Media Server будет писать в каталог `/storage/streamname` или дисковый массив `@my_raid`. Очищать архив он будет либо через день, либо когда диск, на который пишут, заполнится на 50%. Вместо дней можно указывать часы: `20h`. Параметр `schedule` позволяет задать расписание для записи архива в виде интервалов. Время указывается в UTC в часах и опционально минутах, интервал может перехлёстываться через полночь: `22-1:30`. Расписание может содержать несколько интервалов, разделенных запятой: `8:00-16:00,22-1:30`.

dvr_offline

`dvr_offline /storage 1d 50%;` При указании этой опции, поток не будет включать запись на старте. Её надо будет включать явно через API. Эта опция указывается вместо опции ``dvr``.

udp

`udp 239.0.0.1:4501 multicast_loop;` `udp 239.0.0.1:4501;` Flussonic Media Server будет посылать поток по UDP MPEG-TS. Чтобы задать UDP сокету параметр `MULTICAST_TTL`, используйте следующий синтаксис: `udp 239.0.0.1:4501?ttl=8;`. Чтобы задать постоянный битрейт (CBR), используйте следующий синтаксис: `udp 239.0.0.1:4501?cbr=2000;`, где 2000 — битрейт в kbit/sec.

thumbnails

`thumbnails;` Включает генерацию JPEG-скриншотов потока..

retry_limit

`retry_limit 10;` Количество раз, которое Flussonic Media Server будет пытаться подключиться к источникам, прежде чем отключить не статический поток.

clients_timeout

`clients_timeout 10;` Количество секунд которое Flussonic Media Server будет держать нестатический стрим после последнего обращения клиента.

source_timeout

`source_timeout 10;` Количество секунд, которое Flussonic Media Server будет ожидать кадры от источника. По прошествии этого времени Flussonic Media Server переподключится к источнику.

По умолчанию значение source_timeout равно 60 секундам.

frames_timeout

frames_timeout 3; Время в секундах, в течение которого Flussonic Media Server ожидает кадры от источника прежде чем генерировать событие frames_timed_out. Это время должно быть меньше, чем source_timeout. Событие frames_timed_out уведомляет вас о том, что источник, возможно, скоро отключится. Если от источника снова начали поступать кадры, до наступления source_timeout, Flussonic генерирует событие frames_restored.

password

password secret; Пароль, который должен передаваться в query string (http или rtmp) для публикации в поток или в группу.

push

push rtmp://destination-server/name; При указании этой опции Flussonic Media Server будет публиковать поток на другой сервер.

backup

backup vod/blank.mp4; Указание этой опции в потоке будет запускать указанный файл vod/blank.mp4 пока нет видео от источника.

url_publish://

url_publish://; Указание этой опции в потоке позволяет публиковать в него видео. Не используется для группы потоков.

on_publish

on_publish http://host/publish.php; on_publish /etc/flussonic/publish.lua; При публикации в этот поток или группу потоков будет вызывать скрипт или делаться HTTP запрос с параметрами в виде названия потока, IP адреса публикующего и т.п. В ответ можно либо отказать в публикации, либо разрешить: — HTTP бэкенд должен вернуть либо 200 OK, либо 403 Forbidden; — .lua скрипт должен вернуть либо {true, {}}, либо {false, {}}.

max_sessions

max_sessions 1000; Лимит по количеству сессий на поток.

settings_rtp

rtp udp; Включает принудительное использование UDP для работы с RTSP камерами.

add_audio_only

add_audio_only; Добавляет в HLS плейлист ссылку на поток только с аудио. Требуется для прохождения валидации приложения в Apple.

prepush off

prepush off; Отключается [prepush](#prepush), т.е. более быстрый старт. Может быть нужно для сохранения риа тайма в потоке.

prepush

prepush 10; Включает буфер заданной продолжительности. Если соединение клиента с

сервером прерывается или замедляется, то он проигрывает видео из буфера, что позволяет плееру стартовать быстрее, но с отставанием.

max_bitrate

max_bitrate 1000; Лимит битрейта публикуемого потока.

logo

В версии 4.6.15 и выше.

logo path=flu/embed-logo.png height=100 width=100 left=0 top=0; Добавить логотип при воспроизведении потока. Такой логотип не будет отображаться на мобильных устройствах и в DVR плеере. Чтобы добавить логотип к видео используйте транскодер. path (Обязательный параметр) — путь относительно директории wwwroot. height, width — размер изображения логотипа в пикселях. Если задан только один из этих параметров, то второй будет изменен пропорционально. Не указывайте эти параметры, чтобы оторазить логотип в исходном размере. left, top, right, bottom — положение логотипа заданное в виде смещения в пикселях от левого, верхнего, правого и нижнего края видео. Например, чтобы отобразить логотип в правом нижнем углу: `right=0, bottom=0`. Не используйте одновременно параметры left, top, right и bottom.

mpegts_pids

mpegts_pids pmt=4095 sdt=0x12 v1=211 v2=212 a0=220 t0=16#fb; С помощью этого параметра задаются значения PID исходящему MPEG-TS потоку. Можно задать PID для PMT, SDT, дорожек с видео, аудио и субтитрами. Дорожки нумеруются начиная с 1: `a1=123` установит PID для первой аудио дорожки. Можно задать базовый PID для дорожек определённого типа с помощью индекса 0. Например, `t0=100` установит первой дорожке субтитров PID равный 101, второй - 102 и т.д. Числа могут быть заданы в десятичном виде (по умолчанию) или префиксом `0x` для шестнадцатеричных чисел.

segment_duration

segment_duration 4; Регулирует длительность сегмента в HLS и HDS потоках (в секундах). Не для всех получаемых потоков Flussonic сможет применить указанную длительность сегмента. Всё зависит от длины GOP у потока. Нужно, чтобы сегмент был кратен GOP, потому что GOP нельзя порезать на части. Например, для потока с GOP в 4 секунды допустима длина сегмента 4 секунды, 8 секунд, 12 секунд и т.д. Если это не так, то Flussonic создаёт сегменты равные GOP.

segment_count

segment_count 4; Регулирует количество сегментов в HLS и HDS плейлистах.

group

group sport; Используется только на сервере-источнике, чтобы отнести поток к одной или нескольким группам ТВ-каналов. Подробнее

disabled

disabled; Отключить поток.

Варианты источников

Источник — это то, откуда Flussonic получает поток видеоданных. Источником может быть IP камера, спутник, HTTP сервер и т.д. Для указания источника нужно знать его адрес и видео-протокол, по которому Flussonic получает данные, т.е. нужно знать URL.

Для передачи видеоданных между серверами Flussonic Media Server мы рекомендуем использовать внутренний протокол Flussonic — M4F. Например, он подходит для репликации DVR архива, для рестриминга отдельных потоков в кластере.

Важно. Flussonic Media Server не поддерживает тип источника http, и адрес источника вида `http://host/path` не будет работать. Требуется более детально указывать, что это — HTTP MPEG-TS, HLS или другой протокол. Ниже на этой странице показаны варианты URL для разных протоколов.

Важно. В качестве источника не указывайте файл на сервере. Раздача файлов настраивается по-другому. Подробнее читайте в статье VOD — вещание файлов.

Содержание

- M4F
- MPEG-TS по HTTP
- MPEG-TS по UDP
- MPEG-TS по TCP
- HLS
- RTSP
- RTMP
- File
- SHOUTcast
- H323
- Таймшифт
- Миксер
- Мозаика
- MPEG-TS опции
- Передача closed captions в формате WebVTT
- HTTP опции подключения

M4F

Используется для передачи между серверами Flussonic.

`m4f://hostname:port/stream` — синтаксис

`m4f://flussonic.example.com/channel_01` — пример

О протоколе M4F

MPEG-TS по HTTP

tshttp://hostname:port/stream — синтаксис (захват по HTTP)
tshttps://hostname:port/stream — синтаксис (захват по HTTPS)
tshttp://192.168.100.1/worldnews.ts — пример настоящего URL

MPEG-TS по UDP

udp://239.255.1.1:5500 — мультикаст URL
udp://239.255.1.1:5500/192.168.10.1 — захват мультикаста через указанный интерфейс
udp://192.168.0.1:12345 — юникаст URL

MPEG-TS по TCP

tstcp://192.168.0.1:54321

HLS

hls://HOSTNAME/PATH/TO/MANIFEST.m3u8 — синтаксис
hls://10.0.0.1/news.m3u8 — пример настоящего URL
hls://example.com/news/index.m3u8 — пример настоящего URL.
hlss://HOSTNAME/PATH/TO/MANIFEST.m3u8 — захват по HTTPS.
hls2://example.com/news/index.m3u8 — захват мультибитрейтного HLS источника.
hlss2://example.com/news/index.m3u8 — захват мультибитрейтного HLS источника по HTTPS.

По умолчанию, используя hls://, Flussonic Media Server будет захватывать только одно из доступных качеств.

Для захвата мультибитрейтного источника укажите hls2://.

При запуске потока Flussonic Media Server проверяет, что источник не завис, скачивая несколько раз индекс.

Проверка может быть отключена с помощью опции skip_stalled_check:

```
stream ort {  
  url hls://source:8080/stream/index.m3u8 skip_stalled_check=true;  
}
```

RTSP

Одного IP адреса камеры недостаточно для получения с нее видео. Всегда нужно указать ещё один путь. Он не всегда приводится в документации, поэтому, возможно, придется обращаться к продавцу или производителю камеры.

rtsp://hostname/path — синтаксис
rtsp://user:password@ip/path — URL с указанием авторизации
rtsp2://hostname/path — включает транскодирование звука в AAC. Подробнее
rtsp://192.168.0.100/h264 — пример настоящей ссылки

RTMP

rtmp://hostname/application/stream

rtmp://10.0.0.1/live/news

Протокол требует, чтобы в URL было не меньше двух сегментов. Первый сегмент, по умолчанию, используется для указания имени RTMP приложения (application). Если название RTMP приложения на сервере состоит больше чем из одного сегмента, то в адресе надо указать два слеша для явного разделения на application и stream name.

File

file://vod-location/file.mp4 синтакс

file://vod/bunny.mp4 пример реального потока, подробнее о VOD путях читайте в статье

Flussonic может создать зацикленный поток из любого .mp4 или .ts файла, укажите адрес потока используя протокол file://.

SHOUTcast

shout://example.com/radio

Схема shout:// позволяет захватить SHOUTcast и ICEcast радио-потоки.

H323

url h323://192.168.100.150

Схема h323:// позволяет захватить видео по VoIP протоколу H323. Подробнее

Таймшифт

timeshift://ipcam/3600

timeshift://news/10600

Схема timeshift:// позволяет создать поток, идентичный исходному (ipcam, news в примере), который будет вещаться из архива с заданным отставанием (в секундах).

Миксер

mixer://stream1,stream2

Схема mixer:// позволяет склеить видеодорожку из первого потока с аудиодорожкой из второго. Подробнее в отдельной статье.

Мозаика

mosaic://cam1,cam2,cam3,cam4?fps=20&preset=ultrafast&bitrate=1024k&size=340x240&mosaic_size=4

Схема mosaic:// позволяет собрать несколько потоков на один экран (мозаика). Подробнее в отдельной статье.

MPEG-TS опции

Для tshttp://, udp://, tstcp:// источников можно указать дополнительные опции.

Захват конкретной MPEG-TS программы и PID-ы:

```
stream ort {  
    url tshttp://source:8080/stream program=21 pids=45,46,47;  
}
```

Управление субтитрами

Вы можете управлять DVB-субтитрами в выходном потоке с помощью опции subtitles, которая имеет следующие значения:

drop — в выходном потоке не будет дорожки субтитров. Пример см. ниже.

accept — в выходном потоке будет дорожка с субтитрами в DVB формате, без преобразования в текст.

ocr_replace — в выходном потоке будет дорожка, содержащая субтитры, преобразованные в текстовый формат (WebVTT).

ocr_add — выходной поток будет иметь две дорожки субтитров: исходную дорожку с субтитрами в DVB и новую дорожку с текстовыми субтитрами.

Принятие субтитров

Flussonic Media Server по умолчанию всегда принимает субтитры от публикуемых потоков. Это справедливо как для публикаций в статический поток, так и по динамическому имени. Чаще всего не нужно явно указывать subtitles=accept.

Отклонение субтитров

Если субтитры в потоке не нужны, опцию subtitles следует установить в drop:

```
stream ort {  
    url tshttp://SOURCE:8080/STREAM subtitles=drop;  
}
```

Замечание. Для live-потоков субтитры всегда включены. Опция subtitles не поддерживается для live публикаций, и выключить субтитры в этом случае не получится.

Устаревшая опция allow_subtitles поддерживается только для обратной совместимости.

Конвертация субтитров

О распознавании и преобразовании DVB-субтитров см. Конвертация субтитров из DVB в WebVTT

NIT-таблица

Включить пропуск NIT-таблицы:

```
stream ort {  
    url tshttp://SOURCE:8080/STREAM bypass_eit=true;  
}
```

Включите эту опцию, если хотите получить EPG на выходе Flussonic.

HTTP опции подключения

При подключении к источнику по HTTP-протоколам может потребоваться указать определенный User-Agent:

```
stream example {  
  url tshttp://source:8080/stream user_agent="VLC"  
}
```

Чтобы указать HTTP заголовок, например, Referer:

```
stream example {  
  url tshttp://source:8080/stream header.Referer=https://flussonic.com;  
}
```

Передача Closed Captions в формате WebVTT для HLS и в TTML для DASH

Flussonic позволяет передавать скрытые субтитры CEA-608/708 из входного MPEG-TS:

в формате WebVTT для HLS

в формате TTML для DASH.

Чтобы использовать эту функцию, добавьте параметр closed_captions=PID к URL-адресу потока.

Для MPEG-TS:

```
stream tv2 {  
  url tshttps://HOST/STREAM/mpegts closed_captions=211;  
}
```

Для HLS:

```
stream tv2 {  
  url hlss://HOST/STREAM/index.m3u8 closed_captions=211;  
}
```

Для обозначения расположения субтитров на видео используйте опцию substyle valign=top|middle|bottom align=left|center|right, например:

```
stream tv2 {  
  url tshttps://HOST/STREAM/mpegts closed_captions=211;  
  substyle valign=top align=left;  
}
```

После преобразования в выходном HLS потоке присутствуют скрытые субтитры в формате WebVTT:

Переключение источников

Резервные источники видео

Для обеспечения резервирования источников можно указать несколько разных источников видео и Flussonic Media Server будет переключаться между ними, если выбранный источник стал недоступен.

Под термином «стал недоступен» подразумевается либо немедленное отключение, либо отсутствие кадров в течение 10 секунд.

```
stream rtr {  
  url tshttp://10.0.4.5:9000/channel/5;  
  url tshttp://10.2.4.5:9000/channel/5;  
}
```

Если пришлось переключиться на второй источник, то Flussonic Media Server будет периодически перепроверять первый источник на работоспособность.

```
stream rtr {  
  url tshttp://10.0.4.5:9000/channel/5 priority=1 source_timeout=60;  
  url tshttp://10.2.4.5:9000/channel/5 priority=2 source_timeout=60;  
}
```

source_timeout

Для каждого источника можно указать количество секунд, которое Flussonic Media Server будет ожидать кадры от этого источника. Если был задан source_timeout для всего потока, то в этом случае он не применяется. По умолчанию значение source_timeout равно 60 секундам.

priority

Для источников можно указать приоритет. По умолчанию, высший приоритет имеет первый источник в списке, низший приоритет — последний источник в списке. Если для каких-то источников в списке приоритет не указан, то применяется нумерация по умолчанию. Если приоритет недоступного источника такой же как и текущего, то Flussonic Media Server не будет периодически проверять такой недоступный источник.

Проверки источника

Flussonic Media Server следит только за тем, чтобы от источника были кадры, и переключается на другой источник, если кадры не поступают определенное количество секунд.

Он не переключит источник при пропадании звука или видео, или при росте количества MPEG-TS CC ошибок.

Запись

Если на потоке настроен архив, то Flussonic Media Server пишет видео из активного источника в

архив.

В случае, если последним адресом указан локальный файл:

```
stream cnn {  
  url udp://239.1.2.3:1234;  
  url file://vod/backup.mp4;  
  dvr /storage;  
}
```

то видео из этого файла также будет писаться в архив.

Чтобы пользователям показывать заглушку вместо потока, но не писать её в архив, надо использовать заглушку

Публикация видео на сервер

Flussonic Media Server может принимать видео от программ и устройств, которые являются инициаторами начала видеотрансляции. Это называется

публикация

Публикация может использоваться в тех случаях, когда устройство не имеет статического IP адреса или вообще находится за NAT и Flussonic Media Server не сможет обратиться к этому устройству или программе за видео.

Что мы называем публикацией:

Передачу видео с любого мобильного устройства на сервер Flussonic.

Передачу видео из OBS (Open Broadcaster Software) или vMix на сервер Flussonic. Узнать больше

Передачу видео с HTML страницы в браузере через WebRTC на сервер Flussonic. Узнать больше

Также Flussonic может публиковать видео в социальные сети.

Что мы не называем публикацией:

Получение мультикаста

Прием потока из какого-либо источника

В этих ситуациях Flussonic неким образом подключается к источнику. Но ситуация, когда Flussonic не делает ничего для того, чтобы инициировать соединение, называется публикацией: например, публикация — это когда мобильное устройство подключается к Flussonic для передачи видео на сервер.

Если быть точными, публикация видео в социальные сети не подходит под определение публикации, которое мы используем в документации.

Протоколы

Flussonic Media Server может принимать запросы на публикацию видео по протоколам RTMP, RTSP, HTTP MPEG-TS и WebRTC.

Содержание:

Публикация в статический поток

Настройка публикации в статический поток через UI

Настройка публикации в статический поток через UI

Публикация по динамическому имени

Настройка публикации по динамическому имени через UI

Настройка публикации по динамическому имени через UI

Публикация по RTMP

Публикация по RTSP

Публикация по MPEG-TS

Публикация по WebRTC

Авторизация при публикации потока

Расширенная валидация публикации

Архив и динамические имена потоков

Перепубликация

Публикация в статический поток

Если вы точно знаете под каким именем поток должен появиться на сервере, то вы можете создать поток и указать ему, что вы разрешаете в него публиковать, путём указания особого источника `url publish://`:

```
stream published {  
  url publish://;  
}
```

Для применения настроек не забудьте выполнить команду:

```
/etc/init.d/flussonic reload
```

Важно! Начиная с версии Flussonic 19.01 вместо `publish_enabled` используется `url publish://`. Конфигурационный файл версии 19.01 не получится использовать в предыдущих версиях Flussonic. Новая директива `'url publish://'` предлагает гораздо больше гибкости в управлении публикациями — вы можете использовать `timeout`, транскодирование и поддержку WebRTC.

Подробнее о том, как настроить публикацию

URL для публикации по разным протоколам

При публикации в статический поток вы можете публиковать видео используя следующие адреса:

`rtsp://flussonic-ip/published`

`http://flussonic-ip/published/mpegts`

`rtmp://flussonic-ip/published`

`rtmp://flussonic-ip/static/published`

Особенности URL для RTMP

Важно! Если при публикации по RTMP вы укажете простое имя потока, то у стороннего ПО может возникнуть вопрос — какое имя приложения использовать. Flussonic Media Server выкидывает `rtmp` из имени приложения, поэтому, если ПО обязательно требует какое-то имя приложения, укажите ему `static`. Это будет выглядеть так:

server URL: `rtmp://flussonic-ip/static`

stream name: `published`

Если вы используете сложное имя стрима, например `client15/published1`, то вы можете использовать:

server URL: rtmp://flussonic-ip/client15

stream name: published1

При такой конфигурации вы можете использовать все настройки потока.

Настройка публикации в статический поток через UI

Добавить параметр "url publish://" вы можете и через веб-интерфейс администратора Flussonic.

Чтобы создать статический поток с публикуемым источником:

В административном интерфейсе создайте поток: Media > Stream > add.

Заполните Stream name.

Укажите publish:// в качестве Source URL. Либо, сохранив настройки, перейдите на вкладку Input и нажмите акцепт под Published input.

Нажмите create.

Чтобы удалить публикуемый источник из настроек потока:

Откройте созданный поток и перейдите во вкладку Input.

2. В разделе Published input нажмите ignore. Другой способ — нажать remove напротив URL, содержащего источник publish://.

Нажмите save для применения настроек.

Публикация по динамическому имени

Зачем использовать динамическое имя и publishing location

Указывать статическое имя для публикуемых потоков может быть непрактично в следующих случаях:

Публикация длится ограниченный период времени (по сравнению с трансляцией телеканала, например) и происходит лишь один раз.

Может быть слишком много публикаций, чтобы создавать отдельный поток для каждой из них. Кроме того, вам пришлось бы прописывать настройки отдельно для каждого потока.

В некоторых случаях неизвестно, под каким именем внешнее приложение-клиент будет публиковать поток. Flussonic не знает имя потока, пока не придет запрос на публикацию видео на сервер. Зачастую имя выбирается произвольно (например, в случае с веб-чатами).

Flussonic позволяет решить эти проблемы с помощью создания префикса публикации (publishing location), где можно указать настройки сразу для множества потоков.

Динамическое имя означает, что полное имя потока формируется из предварительно настроенного префикса публикации и заранее не известного имени, определенного во внешнем приложении.

Таким образом, если вы заранее не знаете под каким именем будет публиковаться поток или этих потоков будет много, то настройте префикс публикации:

```
live chats {  
}
```

Здесь chats — это префикс публикации. Все потоки, опубликованные под префиксом chats, будут иметь настройки, которые вы укажете в директиве live.

О всех опциях потока см. в разделе про настройку потоков.

Публикация через WebRTC в live локации

Вы можете использовать следующие настройки транскодера с публикуемыми live потоками по WebRTC:

`output_audio=(aac|aac_opus|opus)`. Указывает, как транскодировать аудио. Можно получить результирующий поток в AAC, AAC+Opus или Opus. По умолчанию используется `aac_opus` — входные потоки, которые обычно имеют аудио в Opus, на выходе получают также дорожку в AAC.

`prefer_codec=(h264|vp8|vp9)`. Позволяет на старте публикации предложить один из перечисленных видеокодеков. По умолчанию используется кодек H264. Видео в VP8 и VP9 проходит без транскодирования, его поддержка будет реализована в будущем.

Пример:

```
live mylive {  
    output_audio aac;  
    prefer_codec vp8;  
}
```

URL для публикации по разным протоколам

В случае публикации по динамическому имени вам надо публиковать стримы под именами с префиксом, например:

```
rtsp://flussonic-ip/chats/tempname  
http://flussonic-ip/chats/tempname/mpegts  
rtmp://flussonic-ip/chats/tempname
```

Что именно идет после `chats` — это дело клиента. Flussonic Media Server заранее не знает, какое именно это будет имя.

Настройка публикации по динамическому имени через UI

Создать префикс для публикации по динамическому имени (publishing location) можно через веб-интерфейс.

Откройте вкладку Media:

Найдите раздел Publishing locations и нажмите add:

Введите имя префикса и сохраните конфигурацию.

Имя должно быть уникальным, состоять из английских букв, цифр, знака подчеркивания и слэша (/).

Крайне не рекомендуется использовать другие символы — они могут быть восприняты различными программами на клиентах как служебные (сам Flussonic ограничений на имя не накладывает).

Префикс для публикации создан.

О настройке других опций потока см. в разделе про настройку потоков.

Публикация по RTMP

С настройкой публикации по RTMP есть ряд сложностей. Дело в том, что протокол RTMP не имеет как таковых адресов. При подключении к серверу по RTMP клиент указывает имя сервера + имя приложения, а потом уже указывает в какой поток публиковать или какой поток проигрывать.

Обычно для сокращенной записи имя приложения и имя потока склеивают. Например, пара из `rtmp://rtmp.myhosting.com/chats` и `chat-15` превращается в `rtmp://rtmp.myhosting.com/chats/chat-15`.

Стандартная практика при интерпретации RTMP адреса `rtmp://rtmp.myhosting.com/chats/chat-15` — разделить путь после имени сервера по первому слешу и взять первую часть имени в качестве приложения.

В Flussonic Media Server нет концепции приложений, они существуют только в протоколе RTMP и отсутствуют в других протоколах, поэтому при публикации по RTMP используется следующая логика:

Сервер склеивает имя приложения с публикуемым путем. Так пары `rtmp://rtmp.myhosting.com/chats/my`, `chat-15` и `rtmp://rtmp.myhosting.com/chats`, `my/chat-15` превратятся в публикуемое имя потока `chats/my/chat-15`.

Ищется первый префикс публикации, который подходит под это имя. В нашем случае будет выбран префикс `chats`.

Дальше во всех интерфейсах авторизации и т.п. имя потока будет полное: `chats/my/chat-15`.

Опубликовать по RTMP можно, например, с помощью `ffmpeg`:

```
ffmpeg -re -i /opt/flussonic/priv/bunny.mp4 -vcodec copy -acodec copy -f flv rtmp://localhost/chats/my/chat-15
```

При этом в веб-интерфейсе появится новый поток:

Публикация по RTSP

Некоторые клиенты могут публиковать видео по RTSP.

Flussonic Media Server поддерживает автоматический выбор между UDP и TCP транспортом: как захочет клиент, так и будет принимать.

Имя потока должно быть полным: `chats/my/chat-15`

```
ffmpeg -re -i /opt/flussonic/priv/bunny.mp4 -vcodec copy -acodec copy -f rtsp rtsp://localhost/chats/my/chat-15
```

Публикация по MPEG-TS

При транскодировании потока с помощью `ffmpeg` можно опубликовать видео по HTTP, добавив суффикс `mpegs`:

```
ffmpeg -re -i /opt/flussonic/priv/bunny.mp4 -vcodec copy -vbsf h264_mp4toannexb -acodec copy -f mpegs http://localhost:8080/chats/my/chat-15/mpegs
```

Авторизация при публикации потока

Flussonic Media Server может проверять пароль при публикации потока. Укажите пароль в конфигурационном файле следующим образом:

```
live live {  
    password mypass;  
}  
  
stream s1 {  
    password secure;  
    url publish://;  
}
```

RTMP

Чтобы опубликовать в защищенную паролем зону поток по RTMP, укажите данные следующим образом:

```
rtmp application: rtmp://192.168.2.3/live  
stream name: mystream?password=mypass
```

HTTP MPEG-TS

Чтобы опубликовать поток по HTTP MPEG-TS, укажите данные следующим образом:

```
http://192.168.2.3:8080/s1/mpegts?password=secure  
ffmpeg -re -i video.mp4 -vcodec copy -acodec copy -  
f flv rtmp://192.168.2.3/live/mystream?password=mypass  
ffmpeg -re -i video.mp4 -vcodec copy -bsf h264_mp4toannexb -acodec copy -  
f mpegts http://192.168.2.3:8080/s1?password=secure
```

Расширенная валидация публикации

Flussonic Media Server позволяет указать HTTP URL или пользовательский скрипт, которые будут проверять расширенную информацию об источнике публикации перед тем, как разрешить или запретить публикацию.

```
live chats {  
    on_publish http://my-site.com/publish-check.php;  
}
```

или

```
live chats {  
    on_publish /etc/flussonic/publish_check.lua;  
}
```

В HTTP запросе Flussonic Media Server передает по методу POST JSON объект с полями, описанными ниже. В lua скрипт передается объект req с такими же полями.

Передаются следующие поля:

- name — имя публикуемого потока (например chats/15), включающее префикс публикации
- location — префикс публикации
- proto — протокол публикации (rtmp, rtsp, mpegts)
- args — параметры из query string запроса. Для RTMP берется query string из имени

публикуемого потока

ip — IP адрес источника

HTTP обработчик должен вернуть код 200, что бы публикация началась. Также может вернуть заголовок x-record, чтобы сообщить в какой файл записывать публикуемый поток.

lua скрипт должен вернуть пару true и объект, в котором допустимо поле record_path с путем для записи.

Важно! Если обработчик сообщает, куда записывать файл, то Flussonic Media Server будет писать бесконечный flv файл, не отслеживая ни его длину, ни возможную перезапись имеющегося файла.

Архив и динамические имена потоков

Вы можете сконфигурировать архив для live префикса:

```
live recorded {  
    dvr /storage 3d 500G;  
}
```

В этом случае публикуемое видео будет записываться и будет доступно даже если публикация остановилась.

Когда клиент перестает публиковать видео, то стрим через какое-то время пропадает и Flussonic Media Server про него почти ничего не знает. Почти — означает, что в индексе архива информация об этом видео потоке существует и Flussonic Media Server не потеряет эти файлы на диске.

Система зачистки архива удалит их по расписанию.

Перепубликация

Когда вы используете префиксы публикации, вы не можете использовать push в UDP:

```
live pushed {  
    udp 239.0.0.1:1234;  
}
```

Точнее можете, но в этом случае все потоки будут отправляться в одну мультикаст группу с печальными последствиями.

Правильнее использовать push с шаблоном (%s) по другим протоколам:

```
live pushed {  
    push rtmp://cdn-server/client43/%s;  
}
```

При такой настройке для перепубликации стрима pushed/mystream будет использоваться адрес rtmp://cdn-server/client43/pushed/mystream

Прием мультикаста

Flussonic Media Server умеет захватывать видео, передаваемое мультикастом по протоколам UDP MPEG-TS и UDP RTP.

Для захвата мультикаста необходимо, чтобы Flussonic Media Server отправил IGMP запрос на присоединение к группе (к мультикаст адресу) в нужный интерфейс и это самый тонкий момент.

В самом простом варианте надо создать поток, дать ему какое-то имя и добавить источник вида `udp://239.0.0.1:1234`

```
stream ort {  
    url udp://239.0.0.1:1234;  
}
```

Содержание:

- Выбор интерфейсов

- Захват MPTS

- Тюнинг ОС

- Проблемы с захватом

- Проблемы с коммутаторами

- Проблемы с головными станциями

Выбор интерфейсов

Как правило сервер, захватывающий мультикаст, имеет больше одного интерфейса: одним интерфейсом он смотрит в локальную сеть, где ходит видео, другим интерфейсом в интернет, куда он это видео отдает по HLS или HTTP MPEGTS.

Внешний интерфейс часто идет первым и Flussonic Media Server по умолчанию будет слать IGMP запросы туда, и не будет получать видео.

Чтобы явно указать через какой интерфейс надо получать мультикаст, можно указать IP адрес текущего сервера на этом интерфейсе. Например, если на интерфейсе eth2 сервер имеет IP адрес 10.100.200.3, то конфигурация будет иметь вид:

```
stream ort {  
    url udp://239.0.0.1:1234/10.100.200.3;  
}
```

Захват MPTS

Можно захватить мультипрограммный транспортный поток (MPTS), для этого создаем потоки, даем им необходимые имена и в качестве источника указываем мультикаст-группу и ID программы.

```
stream 1 {  
    url udp://239.0.0.1:1234 program=2001;  
}
```

```
stream 2 {  
    url udp://239.0.0.1:1234 program=2002;  
}
```

```
stream 3 {  
    url udp://239.0.0.1:1234 program=2003;  
}
```

Подробнее об MPTS

Тюнинг ОС

Настройки Linux по умолчанию не позволяют захватывать видео по UDP без потерь, поэтому надо серьезно увеличивать размеры сетевых буферов.

Подробная информация об этом есть в статье про настройку производительности.

Важно также отметить, что для HD каналов рекомендуется размер буферов порядка 16 мегабайт.

Проблемы с захватом

С мультикастом часто бывают разнообразнейшие проблемы. Если у вас есть проблемы с качеством захватываемого мультикаста, можете попробовать проверить, в чём именно проблема.

Во-первых, вам надо полностью убрать все настройки firewall. iptables -F. Сначала надо сделать чтобы работало, потом всё остальное. В некоторых дистрибутивах Linux (например, CentOS) по умолчанию идут жесткие правила для iptables.

Также отключите rp filter:

```
sysctl -w 'net.ipv4.conf.eth0.rp_filter=0'
```

и

```
sysctl -w 'net.ipv4.conf.all.rp_filter=0'
```

При необходимости измените eth0 на нужный интерфейс.

Далее, следует понимать, что когда вы настроите мультикаст в Flussonic Media Server и будете смотреть видео с Flussonic Media Server, то на качество видео влияет множество факторов: качество сигнала, качество захвата, работа сервера и качество вашей сети. Проблемы, которые вы увидите в такой конфигурации будут говорить только о том, что у вас проблемы, но никак не о работе Flussonic Media Server. Особенно это проявляется при отладке работы HD каналов: смотреть без затыков 10 мегабит видео могут считанные проценты пользователей.

Например, если вы запустите:

```
/opt/flussonic/contrib/multicast_capture.erl udp://239.0.0.1:1234/10.100.200.3 output.ts
```

запишете секунд 30 видео, скопируете себе на компьютер и посмотрите видео в VLC, то вы получите неискаженную картину того, как мультикаст приходит на сервер. Этот скрипт не распаковывает MPEG-TS, а пишет сырой мультикаст на диск.

Если на этом этапе вы получили хорошее ровное видео, то можно идти дальше и запускать на самом сервере:

```
curl -o output.ts http://127.0.0.1:8080/ort/mpegts
```

Таким образом, вы получите видеопоток, который был захвачен Flussonic Media Server, распакован и упакован обратно в MPEG-TS. Этот файл надо скачать себе на компьютер и посмотреть локально, что бы убедиться в том, что ваше качество канала не влияет на эксперименты.

Если на этом этапе видео тоже хорошее, а при просмотре с Flussonic Media Server дергается, проблема скорее всего в том, что канала не хватает на передачу видео с Flussonic Media Server к вам.

Проблемы с коммутаторами

Иногда проблемы возникают с настройками коммутаторов. Например, у одного клиента возникла проблема с ограничением на количество принимаемых каналов. Оказалось, что в нём стоит лимит на количество подписок на одном порту. Это можно было выяснить командой:

```
#debug igmp snooping all
```

При этом появляются сообщения:

```
%Jun 25 15:12:18 2015 SrcIP is 192.168.121.2, DstIP is 226.2.1.16
```

```
%Jun 25 15:12:18 2015 Groups joined have reached the limit, failed to add more groups
```

В данном конкретном случае получилось починить с помощью команды:

```
#ip igmp snooping vlan XX limit group <1-65535>
```

Проблемы с головными станциями

В определенных случаях бывают проблемы с номерами групп на головных станциях.

Рекомендуется использовать группы от 239.1.1.1 и выше — всё, что ниже, иногда может не работать.

Серверные плейлисты

Серверные списки воспроизведения создаются провайдером на сервере Flussonic. Они позволяют запускать ваш собственный телеканал, где вы выбираете, какие файлы и потоки транслировать.

Серверные плейлисты имеют ряд недостатков при их использовании в Интернете для вставки видео на сайт:

- невозможность таргетировать рекламу;
- невозможность учитывать рекламу через adriver и другие подобные сети;
- сложность сделать мультибитрейтной доставки: разные файлы могут иметь разное количество разных битрейтов;
- технически неоправданно сложно делать отмотку назад, а это одно из основных преимуществ интернет-доставки по сравнению с эфирной;
- пауза слишком сложна в реализации.

Невозможность реализации адекватной системы учета рекламы сводит на нет все желание использовать серверные плейлисты. Поэтому вместо серверных плейлистов рекомендуется использовать клиентские плейлисты, когда абонент IPTV сам формирует плейлист из доступных каналов.

Однако серверные плейлисты можно использовать в других случаях. Практика показывает, что пользователям приятнее смотреть то, что им предлагают, а не искать самим.

Применение серверных плейлистов

Серверные плейлисты разумно использовать для:

- одновременного вещания канала на множество абонентов в локальной сети;
- переключения между потоками с разных камер, например, раз в минуту.

Создание серверных плейлистов

Приведем здесь обзорно процесс создания плейлиста в Flussonic Media Server.

Прежде всего, поместите видео файлы на сервер и настройте VOD локацию (см. ниже).

Затем надо составить файл плейлиста. Пусть он будет в каталоге /tmp/playlist.txt. Можно расположить и на другом сервере. В плейлисте перечислите файлы для проигрывания:

```
vod/video1.mp4  
vod/video2.mp4
```

В конфигурационном файле нужно указать VOD-локацию и поток, который будет проигрывать плейлист.

```
http 80;
stream playlist1 {
    url playlist:///tmp/playlist.txt;
}
stream playlist2 {
    url playlist://http://HOST/playlist.txt;
}
file vod {
    path /var/movies;
}
```

Если в каталоге /var/movies есть файлы video1.mp4 и video2.mp4, то их проигрывание начнется сразу же и будет зациклено.

В плейлист можно добавить управляющие команды:

```
#EXT-X-MEDIA-SEQUENCE:20
#EXT-X-PROGRAM-DATE-TIME:2013-02-12T12:58:08Z
vod/video1.mp4
#EXTINF:64
vod/video2.mp4
```

Полный пример того, как создать список воспроизведения на стороне сервера, с наложением логотипа и мультикаст-рассылкой, см. в разделе Плейлист оператора.

Управляющие команды

Плейлист поддерживает управляющие команды:

#EXT-X-MEDIA-SEQUENCE — номер первого элемента плейлиста. Используется для правильной ротации и обновления плейлиста.

#EXTINF — продолжительность в секундах проигрывания элемента плейлиста. Может использоваться для вставки прямого эфира.

#EXT-X-UTC — Unix Timestamp того времени, когда надо начать проигрывать элемент плейлиста

#EXT-X-PROGRAM-DATE-TIME — время начала проигрывания элемента плейлиста в формате ISO 8601: 2013-02-12T12:58:38Z по GMT.

После завершения каждого файла плейлист перечитывается.

Надо учитывать следующие правила обработки плейлистов:

Если указана опция EXT-X-MEDIA-SEQUENCE, то запоминается последний проигранный номер и после перечитывания

проигрывание продолжается со следующего номера. То есть содержимое нового плейлиста может быть любым, синхронизация

будет осуществляться со следующего номера. Если в новом плейлисте все номера меньше, чем последний проигранный, то

плейлист будет каждую секунду перечитывать файл, ожидая появления правильного номера;

Если опция EXT-X-MEDIA-SEQUENCE не указана и сам файл плейлиста не менялся, то проигрывается следующий элемент.

Если менялся, то проигрывание начинается с начала.

Использование потоков в плейлисте

Например, у вас на Flussonic есть такие два потока: cam1 и cam2:

```
stream cam1 {  
  url rtsp://192.168.1.21:554/user=admin&password=&channel=0&stream=0;  
}  
stream cam2 {  
  url rtsp://192.168.1.22:554/user=admin&password=&channel=0&stream=0;  
}
```

В плейлисте playlist.txt запишем имена потоков:

```
#EXTINF:60  
cam1  
#EXTINF:60  
cam2
```

Информация о состоянии плейлиста (playlist)

HTTP API позволяет запросить текущее состояние серверного плейлиста и получить JSON ответ с данными.

URL: /flussonic/api/playlist/STREAM_NAME

Пример: http://example.flussonic.com:8080/flussonic/api/playlist/example_stream

Параметры:

STREAM_NAME — имя потока.
Обязательный параметр

Ответ: JSON вида

```
{  
  "current_entry": "vod/ir.mp4", //Идентификатор текущего элемента в плейлисте  
  "current_type": "file", //Тип текущего элемента в плейлисте  
  "duration": null, //Длительность текущего элемента в миллисекундах (null - неизвестна)  
  "position": 5.22e4 //Позиция проигрывания внутри текущего элемента, в миллисекундах  
}
```

Миксер

Flussonic Media Server умеет создавать новый поток, используя видео и аудио из других live потоков. Это можно использовать, например, чтобы наложить музыку поверх камеры наблюдения.

Настройка

Создайте новый поток и укажите в качестве источника протокол `mixer://` и имя двух потоков: откуда взять видео и откуда взять аудио:

```
stream mix {  
    url mixer://stream1,stream2;  
}
```

где:

`stream1` — имя live-потока из которого Flussonic Media Server возьмет видеодорожку

`stream2` — только звук.

Внимание! Миксер работает только с live-потоками, уже заведенными во Flussonic Media Server. Не используйте миксер с VOD файлами и не указывайте источник прямо в строке с `mixer://`.

Пример применения

Например, у нас есть поток `cam1` с камеры видеонаблюдения (h264 video + рсти звук), но камера расположена высоко на столбе и ничего кроме шума ветра не слышно.

```
stream cam1 {  
    url rtsp://cam1.local/h264;  
}
```

Логично выключить звук совсем, захватив только видео:

```
stream cam1 {  
    url rtsp://cam1.local/h264 tracks=1;  
}
```

А можно создать новый поток с помощью миксера, который наложит аудио с другого источника. Например, радио:

```
stream cam1 {  
    url rtsp://cam1.local/h264;  
    dvr /storage 7d;  
}  
stream radio {  
    url shout://example.com/fm;  
}  
stream cam1radio {  
    url mixer://cam1,radio;  
}
```

В такой конфигурации мы получаем поток sam1radio, который можно вставить на сайт. Зрителям будет интереснее смотреть на камеру, слушая новости, а в архив будет сохраняться оригинальный поток sam1, включая оригинальный звук с камеры. Это может быть полезно, если произойдет ЧП.

Decklink SDI

Flussonic Media Server может как захватывать видео с плат захвата Decklink SDI или Decklink HDMI, так и передавать видео на Decklink SDI или HDMI.

Захват видео с платы захвата Decklink SDI или HDMI

Установите драйверы Blackmagic, обновите прошивку и настройте поток таким образом:

```
stream sdi {  
  url decklink://0 enc="vb=1024k preset=superfast tune=zerolatency ab=32k";  
}
```

Flussonic Media Server подключится к указанному первому устройству (0) и запустит на нем автоконфигурацию для поиска активного разрешения.

Некоторые модели Decklink не поддерживают автоматический поиск активного разрешения и для них необходимо указывать режим вручную с помощью опций mode и vinput. Например, для Intensity Pro с подключенным к нему по HDMI источником 720p и 50 fps поток нужно настроить так:

```
stream sdi {  
  url decklink://0 mode=hp50 vinput=3 external=true enc="vb=3096k ab=64k";  
}
```

Параметры:

mode - режим входящего потока

vinput - видео интерфейс

- 1 - Composite
- 2 - Component
- 3 - HDMI
- 4 - SDI
- 5 - Optical SDI
- 6 - SVideo

- 1 - Composite
- 2 - Component
- 3 - HDMI
- 4 - SDI
- 5 - Optical SDI
- 6 - SVideo

ainput - аудио интерфейс

- 0 - Analog
- 1 - Embedded

2 - AESEBU

0 - Analog

1 - Embedded

2 - AESEBU

external - запустить захват во внешнем процессе

true

false - по умолчанию

true

false - по умолчанию

vpts - режим синхронизации для видео

2 - по аудио

3 - по эталонному сигналу

2 - по аудио

3 - по эталонному сигналу

apts - режим синхронизации для аудио

1 - по видео

3 - по эталонному сигналу

1 - по видео

3 - по эталонному сигналу

Устранение чересстрочности в прогрессивных потоках

Flussonic может устранять чересстрочность в прогрессивных потоках для улучшения качества видео. Для этого следует использовать метод деинтерлейсинга CUDA yadif:

```
stream test {
```

```
  url decklink://1 enc="vb=4000k hw=nvenc preset=slow fps=50 deinterlace=yadif ab=128k" vinput=4;
```

```
}
```

Вывод на плату Decklink SDI или Decklink HDMI

Flussonic Media Server может не только захватывать, но и передавать видео на плату захвата и вывода Decklink SDI или HDMI.

Для вывода на Decklink укажите параметр push decklink://:

```
stream test {
```

```
  url udp://239.0.0.1:1234;
```

```
  push decklink://0 size=720x576 fps=50 volume=0.01 deinterlace=true;
```

```
}
```

Flussonic передает раскодированный поток на указанный номер устройства или порт на самой карте (например, 0). При необходимости можно указать опцию deinterlace=true для устранения чересстрочности. Максимальный уровень громкости (volume) — 1.0.

Отправка потока на другие серверы

Копирование потока на другие серверы (push)

Flussonic Media Server может принудительно копировать поток на другие серверы при помощи директивы push. Например, можно копировать поток в CDN:

```
stream breakingnews {  
    url publish://;  
    push rtmp://cdn1/myapp/breakingnews;  
    push tshttp://cdn2/breakingnews;  
    push hls://cdn3/breakingnews;  
}
```

Директива push поддерживает протоколы RTMP, HTTP MPEG-TS и HLS.

Копирование потока в CDN Akamai

```
stream breakingnews {  
    url publish://;  
    segment_count 10;  
    segment_duration 10;  
    push hls://post.[HOSTNAME].akamaihd.net/[STREAM ID]/[ANY STREAM NAME];  
}
```

Субтитры

Flussonic Media Server распознает субтитры, читает телетекст и Closed Captions из MPEG-TS и передает по протоколам HLS и DASH.

Содержание:

- Распознавание DVB субтитров и конвертация в WebVTT

- Передача телетекста в формате WebVTT для HLS и в TTML для DASH

- Передача скрытых субтитров в формате WebVTT для HLS и в TTML для DASH

Наложение логотипа

Используя Flussonic Media Server, вы можете наложить изображение поверх видео двумя способами:

С помощью плеера в HTML. Поверх плеера накладывается прозрачный слой с изображением. Этот способ отлично подходит для вставки видео на сайт.

С помощью транскодера. Более ресурсоемкий способ, но он позволяет «вшить» изображение в видеодорожку. В результате логотип никак не получится удалить или скрыть. Подходит для использования на ТВ-приставках.

Наложение логотипа с помощью плеера

Такой логотип не будет отображаться на мобильных устройствах и в DVR плеере, зато он не создает дополнительной нагрузки на сервер.

```
stream ort {  
  url udp://239.0.0.1:5000;  
  logo path=flu/erly-small.png height=100 width=100 left=0 top=0;
```

В строке `logo path=flu/erly-small.png height=100 width=100 left=0 top=0` использованы опции:

`path` (обязательный параметр) — путь относительно директории `wwwroot`.

`height`, `width` — размер изображения логотипа в пикселях. Если задан только один из этих параметров, то второй будет изменен пропорционально. Не указывайте эти параметры, чтобы отразить логотип в исходном размере.

`left`, `top`, `right`, `bottom` — положение логотипа заданное в виде смещения в пикселях от левого, верхнего, правого и нижнего края видео. Например, чтобы отобразить логотип в правом нижнем углу: `right=0`, `bottom=0`. Не используйте одновременно параметры `left` и `right`, `top` и `bottom`.

Наложение логотипа с помощью транскодера

Такой логотип будет «вшит» в видеодорожку и отображаться на всех устройствах и в архивных записях.

Пример конфигурации:

```
stream ort {  
  url udp://239.0.0.1:5000;  
  transcoder vb=2048k preset=fast logo=/path/to/erly-small.png@10:10 ab=128k;  
}
```

Здесь `10:10` — это координаты от левого верхнего угла экрана.

Для размещения в других частях экрана нужно написать более сложную формулу. Например, для размещения в центре:

```
stream ort {  
  url udp://239.0.0.1:5000;  
  transcoder vb=2048k preset=fast logo=/path/to/erly-small.png@(main_w-overlay_w-10)/2:(main_h-
```

```
overlay_h-10)/2 ab=128k;  
}
```

Для размещения в левом нижнем углу:

```
stream ort {  
    url udp://239.0.0.1:5000;  
    transcoder vb=2048k preset=fast logo=/path/to/erly-small.png@10:(main_h-overlay_h-10) ab=128k;  
}
```

Для размещения в правом верхнем углу:

```
stream ort {  
    url udp://239.0.0.1:5000;  
    transcoder vb=2048k preset=fast logo=/path/to/erly-small.png@(main_w-overlay_w-10):10 ab=128k;  
}
```

Для размещения в правом нижнем углу:

```
stream ort {  
    url udp://239.0.0.1:5000;  
    transcoder vb=2048k preset=fast logo=/path/to/erly-small.png@(main_w-overlay_w-10):(main_h-overlay_h-10) ab=128k;  
}
```

Подробнее про настройку транскодера см. в отдельном разделе.

Проигрывание

Проигрывание потоков

Flussonic Media Server позволяет проигрывать потоки по разным протоколам.

На этой странице вы найдете:

- URL-адреса, которые вы следует использовать в плеерах для воспроизведения видео по различным протоколам

- ссылки на разделы, которые содержат больше информации о настройке воспроизведения через каждый определенный протокол

- способы настройки выходных протоколов в пользовательском интерфейсе и в файле.

URL-адреса для проигрывания видео

embed.html

Адрес: <http://FLUSSONIC-IP/STREAMNAME/embed.html>

В Flussonic Media Server есть специальная страница — [embed.html](#), которая предназначена для вставки видео на сайт или просмотра видео через браузер. Страница автоматически определяет браузер и выбирает поддерживаемый протокол. Для большинства устройств на сегодня — это HLS. Подробнее в статье [Вставка видео на сайт \(embed.html\)](#).

HLS

Адрес для плеера: <http://FLUSSONIC-IP/STREAMNAME/index.m3u8>

Подробнее в статье «Воспроизведение HLS». Для вставки на сайт используйте [\(embed.html\)](#) или любой сторонний плеер. Например, [hls.js](#) или [clappr](#).

HLS для мультязыковых потоков, просмотр на приставке и в VLC

Адрес для плеера: <http://flussonic-ip/STREAMNAME/video.m3u8>

Подробнее в статье «Мультязычный HLS».

HTML5 (MSE-LD)

Поток доступен по адресу: <http://FLUSSONIC-IP/STREAMNAME/embed.html?realtime=true>

Подробнее в статье «HTML5 (MSE-LD) воспроизведение с низкой задержкой».

DASH

Поток доступен по адресу <http://FLUSSONIC-IP/STREAMNAME/index.mpd>

А так же есть специальный плейлист "rewind-N.mpd" с большим «скользящим» окном, позволяющий перематывать и ставить на паузу потоки на долгие часы: <http://FLUSSONIC-IP/STREAMNAME/rewind-7200.mpd>

Подробнее в статье «Воспроизведение DASH».

MSS

Поток доступен по адресу: <http://FLUSSONIC-IP/STREAMNAME.isml/manifest>

HTTP MPEG-TS

Поток доступен по адресу: `http://FLUSSONIC-IP/STREAMNAME/mpegts`

HDS

Адрес для плеера: `http://FLUSSONIC-IP/STREAMNAME/manifest.f4m`

RTMP

Поток доступен по адресу: `rtmp://FLUSSONIC-IP/static/STREAMNAME`

RTSP

Поток доступен по адресу: `rtsp://FLUSSONIC-IP/STREAMNAME`

Если у потока есть несколько аудио- и видеодорожек, то по умолчанию Flussonic выбирает первую дорожку (a1v1).

Можно указать, какие именно дорожки следует отдавать. Для этого укажите номера дорожек, добавив параметр `tracks` к URL потока .

Примеры:

`rtsp://FLUSSONIC-IP/STREAMNAME?tracks=a2v1`

`rtsp://FLUSSONIC-IP/STREAMNAME?tracks=a2v1&from=1` — DVR.

`rtsp://FLUSSONIC-IP/vod/file?tracks=a2v1` — VOD.

`rtsp://FLUSSONIC-IP/STREAMNAME2 = rtsp://FLUSSONIC-IP/STREAMNAME1?tracks=v1a1`

Можно выбрать одну дорожку:

`rtsp://FLUSSONIC-IP/STREAMNAME?tracks=a1` — выбрать только аудио.

`rtsp://FLUSSONIC-IP/STREAMNAME?tracks=v1` — выбрать только видео.

Если указать больше двух дорожек или указать в неверном формате, то будет использоваться параметр по умолчанию (a1v1).

WebRTC

Поток доступен по адресу: `ws://FLUSSONIC-IP/STREAMNAME/webrtc`

Подробнее в статье «WebRTC Проигрывание».

SHOUTcast

Поток доступен по адресу: `http://FLUSSONIC-IP/STREAMNAME/shoutcast`

Flussonic Media Server умеет отдавать SHOUTcast, ICEcast радио поток.

Как настроить протоколы для проигрывания потока

Приведенные выше URL-адреса можно найти в пользовательском интерфейсе Flussonic на вкладке Output в настройках потока.

По умолчанию разрешено проигрывание по всем протоколам, но вы можете запретить проигрывание по протоколам, которые укажете.

Начиная с версии 20.02, вы можете использовать более короткую нотацию для указания выходных протоколов для потока. Пользовательский интерфейс поддерживает новую нотацию, а в файле можно перечислить все протоколы с помощью опции `protocol`.

Прежняя нотация тоже поддерживается.

Пример

Например, для потока channel_01 вы хотите разрешить проигрывание по всем протоколам, кроме MPEG-TS и HLS.

В файле конфигурации эти настройки выглядят следующим образом:

```
stream channel_01 {  
    protocols -mpegts -hls;  
}
```

Для потока channel_02 вы хотите разрешить только DASH и RTMP и запретить все остальные:

В файле конфигурации эти настройки выглядят следующим образом:

```
stream channel_02 {  
    protocols dash rtmp;  
}
```

Воспроизведение HLS

Flussonic Media Server поддерживает раздачу видео по протоколу HLS. Многие из доступных возможностей нестандартны для HLS, но мы поддерживаем их для вашего удобства.

Поддерживаемые кодеки: H264, H265, MPEG2 video, AAC, MP3, MPEG2 audio, AC-3.

Flussonic Media Server позволяет получать по HLS прямой эфир, видео по запросу, видео из архива (catchup и timeshift).

Если входной поток содержал DVB субтитры или телетекст, то они будут переданы в выходной поток, проигрываемый по HLS, если настроить Flussonic для этого. Если поток пишется в архив, то и субтитры сохраняются в архиве.

Содержание:

- Простое воспроизведение HLS

- Мультиязычный HLS

- Добавление «Audio only» для Apple

- Отдельные битрейты для приставок

- DVR catchup playback

- Rewind playlist

- DVR timeshift playback

- Воспроизведение отдельных дорожек

Простое воспроизведение HLS

Если у вас есть простой live поток или файл (одно видео, один звук), то URL для воспроизведения через HLS очень простой:

`http://FLUSSONIC-IP/STREAMNAME/index.m3u8`

```
<video src="http://192.168.2.3:8080/ort/index.m3u8" controls></video>
```

где FLUSSONIC-IP — это пример адреса + порта вашего Flussonic Media Server.

Flussonic Media Server также принимает playlist.m3u8 в конце URL для обратной совместимости с другими серверами.

Когда вы начинаете работать с мультиязычными или мультибитрейтными контентом, все становится сложнее.

Мультиязычный HLS

Если вы хотите воспроизвести ваш мультиязычный поток на iPhone, вам нужно использовать тот же `http://192.168.2.3:8080/STREAMNAME/index.m3u8`

Но если вы хотите посмотреть мультиязычный поток используя VLC или приставку, нужно включать `video.m3u8`.

URL для плеера: `http://flussonic-ip/STREAMNAME/video.m3u8`

```
<video src="http://flussonic-ip/STREAMNAME/video.m3u8" controls></video>
```

Это связано с тем, что, согласно требованиям Apple HLS, для каждого отдельного языка нужно указывать отдельный плейлист с audio only вариантом. В MPEG-TS другой механизм: рядом с видео укладываются все аудиодорожки, и плеер сам выбирает, что будет проигрывать. Чтобы видео можно было посмотреть на iPhone, оно должно соответствовать требованиям Apple. Но VLC и приставки, в нарушение стандарта HLS, ожидают старый вариант MPEG-TS, преобразованный в HLS. Поэтому нужно включать video.m3u8.

Добавление «Audio only» для Apple

Apple требует, чтобы у всех ваших потоков был вариант без видео, только звук.

Они считают, что если пользователь смотрит видео по 3G и, оказавшись в зоне неуверенного приема, лучше у него будет только звук, чем буферизация.

Вы можете включить эту опцию в Flussonic Media Server следующим образом:

```
stream ort {  
    url udp://239.0.0.1:1234;  
    add_audio_only;  
}
```

Замечание. Такая настройка может сделать ваш index.m3u8 адрес невоспроизводимым в VLC или на приставке. В этом случае используйте video.m3u8 (см. выше).

Отдельные битрейты для приставок

Когда у вас есть мультибитрейтный мультязычный контент и вы хотите проиграть его на приставке, которая не поддерживает мультибитрейтный HLS плейлисты, вы можете запросить с Flussonic Media Server отдельные плейлисты с одним видео и всеми звуковыми дорожками, как с опцией mono:

<http://flussonic-ip/STREAMNAME/video1.m3u8>

Этот плейлист не мультибитрейтный, в нем URL до сегментов, в которых первая видео дорожка и все доступные звуковые дорожки.

Если вы хотите доставлять мультязычные мультибитрейтные потоки на приставки, не понимающие стандарт Apple для мультязычности, используйте video.m3u8:

<http://flussonic-ip/STREAMNAME/video.m3u8>

Это мультибитрейтный плейлист, который отдает список плейлистов с разными качествами: video1.m3u8, video2.m3u8 и т.д.

DVR catchup playback

Когда ваш поток уже записан на сервере нашим DVR, вы можете воспроизвести видео через HLS, используя время начала и конца передачи (например, из EPG).

URL будет:

<http://flussonic-ip/STREAMNAME/archive-1508403742-3600.m3u8>

Этот плейлист будет так называемый variant, если в потоке будет больше одной звуковой дорожки или больше одного битрейта. Он будет отдавать список сегментов начиная с UTC 1362504585 (2013, Март, 5, 17:29:45 GMT) и один час вперед.

URL mono даст список сегментов, содержащих все дорожки, в MPEG-TS:

<http://flussonic-ip/STREAMNAME/mono-1362504585-3600.m3u8>

Более конкретный videoN плейлист даст список сегментов с N видео дорожкой и всеми звуковыми дорожками:

<http://FLUSSONIC-IP/STREAMNAME/video1-1362504585-3600.m3u8>

и variant видео плейлист со списком videoN плейлистов:

<http://FLUSSONIC-IP/STREAMNAME/video-1362504585-3600.m3u8>

Перемотка плейлиста

Есть специальный плейлист "rewind-N.m3u8" с большим «скользящим» окном, позволяющий перематывать и ставить на паузу HLS потоки на долгие часы.

<http://FLUSSONIC-IP/STREAMNAME/rewind-7200.m3u8>

7200 — длина HLS плейлиста в секундах. Это означает, что ваши клиенты могут поставить эфир на паузу на 2 часа или перемотать на начало футбольного матча без обращения по специальными архивным ссылкам.

DVR timeshift playback

Если у вас есть поток, записываемый на диск, но вы не настроили отложенный поток, вы можете проиграть видео со сдвигом по времени с помощью правильно построенного HLS адреса.

Список URL для относительного таймшифта:

/timeshift_rel-3600.m3u8

/mono-timeshift_rel-3600.m3u8

/video-timeshift_rel-3600.m3u8

/video1-timeshift_rel-3600.m3u8

и для абсолютного таймшифта:

/timeshift_abs-1508403742.m3u8

/mono-timeshift_abs-1508403742.m3u8

/video-timeshift_abs-1508403742.m3u8

/video1-timeshift_abs-1508403742.m3u8

Воспроизведение отдельных дорожек

Если у потока есть несколько аудио- и видеодорожек, то можно указать, какие именно дорожки следует отдавать. Для этого укажите номера дорожек, добавив строку запроса с параметром filter к URL потока.

Примеры:

<http://flussonic-ip/STREAMNAME/index.m3u8?filter=tracks:v2a1> — выбрать первую аудио- и вторую видеодорожки.

<http://flussonic-ip/STREAMNAME/index.m3u8?filter=tracks:v1> — выбрать только видео.

<http://flussonic-ip/STREAMNAME/archive-1362504585-3600.m3u8?filter=tracks:v2a1> —
проигрывание отрывка длиной 3600 секунд из DVR архива, начиная со времени UTC
1362504585.

Вставка видео на сайт (embed.html)

У сервера Flussonic Media Server есть специальная страница —

embed.html

, которая предназначена для вставки видео на сайт и просмотра видео через браузер.

Технически

embed.html

— тот же плеер, что используется в административном интерфейсе Flussonic Media Server.

Страница с плеером доступна по ссылке:

<http://HOSTNAME/STREAMNAME/embed.html>

Страница автоматически определяет браузер и выбирает поддерживаемый видео-протокол. Для большинства устройств на сегодня — HLS (используется плеером по умолчанию).

Важно. Проигрывание видео в браузере может начаться без звука по причине политики, принятой разработчиками браузера. По ссылке объясняется политика и условия, при которых звук включится сам. Политика автопроигрывания на примере Chrome

Вызывается плеер двумя способами:

При открытии embed.html напрямую (указав ссылку в адресной строке) видео развернется на размер окна браузера и автоматически начнет воспроизведение.

Также embed.html удобен для вставки видео на сайт как части веб-страницы. HTML-код для вставки доступен на странице Overview в свойствах каждого потока в административном интерфейсе.

Пример:

html

```
<iframe style="width:640px; height:480px;" allowfullscreen  
src="http://hostname/streamname/embed.html"></iframe>
```

Код вставляет на страницу окно с плеером фиксированного размера — 640x480 пикселей. Воспроизведение начинается автоматически.

Дополнительные параметры

Для большинства задач не требуется дополнительная настройка, но embed.html имеет параметры, которые можно задать с помощью URL. Дополнительные параметры задаются в адресной строке:

http://HOSTNAME/STREAMNAME/embed.html?autoplay=false&play_duration=600

autoplay — автозапуск воспроизведения при открытии страницы. По умолчанию значение true. Чтобы выключить автозапуск, установите false. Показывает скриншоты перед просмотром. Проигрывание со звуком определяется политикой браузера.

mute - управление звуком, доступные значения true или false.

play_duration — через сколько секунд остановить воспроизведение видео. По умолчанию

выключен. Полезен для экономии трафика.

realtime — включает вещание через протоколы с низкой задержкой. Автоматически выбирается между MSE-LD, RTMP или WebRTC. По умолчанию выключен, установите значение true для воспроизведения видео с низкой задержкой.

Внимание! Вещание с низкой задержкой сильнее загружает сеть и процессор, не используйте его без необходимости.

dvr — открывает архивный плеер. Для доступа к архиву укажите значение true. Подробности ниже.

ago — разрешает пользователям перематывать видео назад. Значение задается в секундах. По умолчанию выключен. Удобнее DVR плеера для просмотра видео за последние несколько минут или часов. Идеально подходит для организации паузы и перемотки живого видео на сайте. Пример, час перемотки задается так: `embed.html?ago=3600`.

from — Unix timestamp начала записи. Указав этот параметр, плеер перейдет к воспроизведению timeshift-abs плейлиста с указанного времени.

to — Unix timestamp окончания записи. Используется только в связке с from. Плеер откроет HLS VOD, и будет доступна перемотка в рамках указанного интервала.

Пример доступа к видео из архива

Например, ссылка для доступа к записи телепередачи будет содержать параметры from и to:

`http://HOSTNAME/STREAMNAME/embed.html?from=1511300552&to=1511300852`

Такие ссылки лучше формировать с помощью серверных скриптов на основе программы передач (EPG) для организации CatchUp сервиса.

DVR плеер

Чтобы просмотреть архив записи для потока, откройте DVR плеер при помощи ссылки:

`http://HOSTNAME/STREAMNAME/embed.html?dvr=true`

Плеер позволяет проиграть видео из архива, для больших архивов доступен календарь, а не только временная шкала. Интерфейс плеера позволяет задать масштаб временной шкалы, включить ускоренное воспроизведение и сохранить указанный интервал в виде MP4 файла.

Для DVR плеера доступны все дополнительные параметры адреса, кроме ago.

Интерфейс плеера позволяет автоматически генерировать ссылки формата `embed.html?dvr=true&from=1511300552` без использования дополнительных утилит. Просто откройте нужный момент в архиве и нажмите на часы, чтобы открыть ссылку с параметром from.

HTML5 (MSE-LD) воспроизведение с низкой задержкой

Долгое время Flash Player был лучшим и единственным способом доставлять видео на веб-страницу с относительно низкой задержкой.

Это требуется для вебинаров, спортивных мероприятий(ставок), видеонаблюдения, удаленного управления.

В настоящее время запланировано постепенное удаление Flash из современных браузеров. WebRTC был добавлен в браузеры, но он имеет отграниченную поддержку аудио и видео кодеков (поддерживаются не все разновидности H.264, нет поддержки AAC).

Мы предлагаем новый способ решения этой проблемы, который позволит смотреть видео с помощью встроенного HTML5 механизма и с действительно низкой задержкой.

Воспроизведение с низкой задержкой в HTML5

Откройте в браузере:

<http://flussonic-ip/STREAMNAME/embed.html?realtime=true>

Если все в порядке (правильные кодеки, работающий поток, работающие вебсокеты), то вы получите видео с довольно низкой задержкой (около 1 секунды).

Детали

Мы используем MSE для доставки и проигрывания видео. Следовательно, поддерживаемые кодеки будут теми же, что и в браузере. Обычно H.264 и AAC поддерживаются, а остальное не считается пригодным.

Вам не нужно ничего кроме HTTP или HTTPS чтобы запустить MSE плеер, это может быть хорошим способом для проигрывания видео в условиях ограниченного окружения.

Вы также можете использовать наш плеер внутри своего приложения без iframe. Читайте о нашем MSE JavaScript плеере

MSE Player

Вы можете использовать наш low latency плеер с открытым исходным кодом.

Быстрый старт без NPM

Модуль для поддержки MSE можно скачать по адресу: <http://flussonic-ip:8080/flu/assets/FlussonicMsePlayer.js>

Добавьте файл в секцию скриптов вашего html-файла:

```
<script type="text/javascript" src="/flu/assets/FlussonicMsePlayer.js"></script>
```

Затем инициализируйте плеер и прикрепите его к `<video/>`. Далее начните воспроизведение:

```
...
<body>
  <video id="player"/>
...
<script type="text/javascript">
  window.onload = function() {
    var element = document.getElementById('player');
    window.player = new FlussonicMsePlayer(element, streamUrl);
    window.player.play();
  }
</script>
</body>
...
```

— Запустите:

```
npm install --save @flussonic/flussonic-mse-player
```

— и импортируйте его в JS:

```
import FlussonicMsePlayer from '@flussonic/flussonic-mse-player'
...
const player = new FlussonicMsePlayer(element, url, opts)
```

Пример приложения с webpack и нашим плеером MSE.

Вы можете найти исходный код плеера MSE на github: [flussonic/mse-player](https://github.com/flussonic/mse-player)

```
var player = new FlussonicMsePlayer(element, streamUrl)
```

element — `<video>` DOM элемент

streamUrl — адрес видео потока

Методы:

`play()` — начать воспроизведение

`stop()` — остановить воспроизведение

`setTracks([videoTrackId, audioTrackId])` — установить видео, аудио треки воспроизведения

`getVideoTracks()` — получить доступные видео треки(использовать в `onMediaInfo`)

`getAudioTracks()` — получить доступные аудио треки(использовать в `onMediaInfo`)

События

`onProgress(currentTime)` — срабатывает при воспроизведении видео и отдает текущее проигрываемое время

`onMediaInfo(metadata)` — срабатывает при получении данных потока. Содержит общую информацию о воспроизводимом видео. А также, в случае мультитрейт видео, информацию о всех его треках. Внутри этой функции или после срабатывания доступны методы `getVideoTracks()/getAudioTracks()`

Установка треков в «мультитрейт» видео

Например, есть видео с тремя видео треками: `v1(800k)`, `v2(400k)`, `v3(200k)` и двумя аудио: `a1(32k)`, `a2(16k)`.

Чтобы по-умолчанию игрались `v2`, `a1`, укажем в конструкторе `FlussonicMsePlayer` такой URL-адрес:

```
'ws://flussonic-ip/s/mse_id?tracks=v2a1'
```

Получить все доступные треки можно с помощью присвоения атрибуту `onMediaInfo` функции-колбеку, которая при загрузке видео потока вернет его метаданные:

```
{  
  width: ...,  
  height: ...,  
  streams: [  
    {
```

```

    track_id: "v1", bitrate: ..., codec: ..., content: "video", fps: ..., ...
    ...

    track_id: "a1", bitrate: ..., codec: ..., content: "audio", fps: ..., ...
  }
]
}

```

Или с помощью методов `getVideoTracks()/getAudioTracks()`, которые после срабатывания `onMediaInfo` вернут доступные видео/аудио треки соответственно.

С помощью метода `setTracks([videoTrackId, audioTrackId])` вы можете установить нужные треки для воспроизведения.

```

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<style>
  .player-container {
    border: 1px solid black;

    #player {
      position: relative;
      width: 100%;

      .mbr-controls {
        display: none;
      }
    }
  }
</style>
<body>
  <div class="player-container">
    <video id="player"/>
  </div>
  <div class="mbr-controls">
    <div>
      <label for="videoTracks">video tracks</label>
      <select name="videoTracks" id="videoTracks"></select>
    </div>
    <div>
      <label for="audioTracks">audio tracks</label>
      <select name="audioTracks" id="audioTracks"></select>
    </div>
    <button onclick="window.setTracks()">set tracks</button>
  </div>

  <button onclick="window.player.play()">Play</button>
  <button onclick="window.player.stop()">Stop</button>
  <script type="text/javascript" src="/flu/assets/FlussonicMsePlayer.js"></script>
</script>

```

```

window.onload = onLoad();

function onLoad() {

    var element = document.getElementById('player');
    var videoTracksSelect = document.getElementById('videoTracks');
    var audioTracksSelect = document.getElementById('audioTracks');
    var mbrControls = document.querySelector('.mbr-controls');

    var url = (window.location.protocol == "https:" ? "wss:" : "ws:") + '//' + window.location.host + '/clock/mse
_id';

    window.player = new FlussonicMsePlayer(element, url);

    window.player.onProgress = function(currentTime) {
        console.log(currentTime);
    };

    window.player.onMediaInfo = (rawMetaData) => {
        var videoTracks = window.player.getVideoTracks()
        var audioTracks = window.player.getAudioTracks()
        var videoOptions = videoTracks.map((v, i) => (
            `<option value="${v['track_id']}">${v['bitrate']} ${v['codec']} ${v['fps']} ${v['width']}x${v['height
']}</option>`
        ))

        var audioOptions = audioTracks.map(v => (
            `<option value="${v['track_id']}">${v['bitrate']} ${v['codec']} ${v['lang']}</option>`
        ))

        videoTracksSelect.innerHTML = videoOptions.join("")
        audioTracksSelect.innerHTML = audioOptions.join("")

        mbrControls.style.display = 'block'
    }

    window.setTracks = () => {
        var videoTrackId = videoTracksSelect.options[videoTracksSelect.selectedIndex].value
        var audioTrackId = audioTracksSelect.options[audioTracksSelect.selectedIndex].value

        window.player.setTracks([videoTrackId, audioTrackId])
    }
}
</script>
</body>
</html>

```

Рассылка мультикаста

При работе с IPTV часто приходится иметь дело с видео, передающимся мультикастом. В подавляющем большинстве случаев, в мультикасте передается MPEG-TS контейнер (по 7 188-байтных пакетов в одном UDP пакете). Реже в сеть передается RTP протокол, внутри которого идет тот же MPEG-TS. RTP нужен для того, чтобы можно было отслеживать потери. В RTP пакете есть 16-битный счетчик, используемый для отслеживания порядкового номера.

Краткие основы мультикаста

Мультикаст — это UDP пакеты, передающиеся от одного источника группе подписчиков. Адрес, по которому посылаются такие пакеты, обычно находится в диапазоне от 224.0.0.0 до 239.255.255.255, однако 224.0.0.0/8 не рекомендуется из-за большого количества специализированных адресов.

В правильно настроенной сети мультикаст-трафик идет до ближайшего роутера, а роутер уже сам выбирает, какому клиенту какой трафик послать на основании пожеланий клиентов. Пожелания передаются по протоколу IGMP, по которому передаются сообщения о включении в группу рассылки какого-то адреса или исключения из группы.

Таким образом для того, чтобы Flussonic рассылал мультикаст клиентам, надо чтобы он слал пакеты в нужный интерфейс (локальная сеть оператора), а роутер был настроен на правильную работу с мультикастом.

Настройка Flussonic

Для настройки мультикаст-рассылки достаточно указать опцию push в стриме:

```
stream ort {  
  url hls://provider.iptv/ort/index.m3u8;  
  push udp://239.0.0.1:1234;  
}
```

Создать такой поток можно и в веб-интерфейсе, указав ему адрес источника (на вкладке Input) и добавив URL для рассылки мультикаста `udp://239.0.0.1:1234` в свойствах потока (на вкладке Output в разделе Push live video to certain URLs).

Можно выбрать дорожки на отправку:

```
stream ort {  
  url hls://provider.iptv/ort/index.m3u8;  
  push udp://239.0.0.1:1234?tracks=v2a4;  
}
```

Где:

v2 — это вторая видеодорожка;

a4 — это четвертая аудиодорожка.

Настройка сервера

После того, как вы настроили мультикаст-вещание потока, с большой вероятностью ничего не заработает, потому что часто из-за настроек сервера мультикаст-трафик пойдет в первый интерфейс, который как правило смотрит в интернет. Таким образом, необходимо, чтобы Flussonic начал отправлять трафик в тот интерфейс, который смотрит в локальную сеть.

```
route add -net 239.0.0.0/8 dev eth2
```

Где eth2 — это название интерфейса, к которому подключена локальная сеть. После такого прописывания роутинга мультикаст с Flussonic польется в нужный интерфейс и его можно будет увидеть на роутере, а следовательно и на клиенте.

Обратная ситуация с захватом мультикаста описана в статье: [Захват мультикаста](#)

Рассылка UDP с постоянным битрейтом (CBR)

Мы предполагаем, что вы ознакомились со статьёй о рассылке мультикаста, потому что эта страница представляет собой дополнение к ней.

Здесь вы можете узнать, как настроить Flussonic Media Server для рассылки UDP мультикаста пригодного для мультиплексирования DVB-C и DVB-S стримерами.

Требования

Если вам необходимо отправить видео в DVB-C сеть или на спутник, то у вас могут быть жёсткие аппаратные требования:

- постоянный битрейт, (действительно постоянный, а не усредненно);
- частая вставка PCR.

Вам может встретиться высокотехнологичный DVB мультиплексор, который сможет объединить потоки вместе, допуская переменный битрейт. Но обычно вы должны заполнить выделенную полосу MPEG-TS пакетами.

Flussonic Media Server умеет добавлять нулевое заполнение, которое не может быть добавлено головной станцией.

Также устройство будет требовать частую посылку PCR: до 1 раза в 20 миллисекунд. Таким образом, может случиться, что вам будет нужно вставить 2 PCR на протяжении одного кадра.

Это также возможно сделать в Flussonic Media Server.

Настройка

Добавьте параметры cbr и pcr:

```
stream ort {  
  url hls://provider.iptv/ort/index.m3u8;  
  push udp://239.0.0.1:1234?cbr=2200&pcr=20;  
}
```

С такими значениями Flussonic Media Server будет заполнять полосу пропускания до 2200 килобит и вставлять PCR каждые 20 миллисекунд.

Если входной поток имеет слишком неравномерный или высокий битрейт (выше, чем указано в параметре cbr), это может привести к переполнению буфера отправки по UDP. Чтобы избежать ошибок, подпишитесь на событие `udp_pusher_does_not_fit_cbr`. Также сообщения о неподходящем битрейте входного потока можно будет увидеть в логах.

Идентификаторы таблиц MPEG-TS потока

Вы можете задать произвольные ID таблиц MPEG-TS потока используя опцию `mpegts_pids`:

```
stream ort {  
  url hls://provider.iptv/ort/index.m3u8;
```

```
mpegs_pids pmt=4095 sdt=0x12 v1=211 v2=212 a0=220 t0=16#fb;  
push udp://239.0.0.1:1234?cbr=2200&pcr=20;  
}
```

Можно задать PID для PMT, SDT, дорожек с видео, аудио и субтитрами.

Дорожки нумеруются начиная с 1: a1=123 установит PID для первой аудио дорожки.

Можно задать базовый PID для дорожек определённого типа с помощью индекса 0. Например, t0=100 установит первой дорожке субтитров PID равный 101, второй - 102 и т.д.

Числа могут быть заданы в десятичном виде (по умолчанию) или префиксом 0x для шестнадцатеричных чисел.

Использование WebRTC для проигрывания видео с Flussonic Media Server

О WebRTC

WebRTC — это P2P протокол общения между двумя клиентами, регламентирующий передачу данных по заранее установленному соединению. Например, для связи двух браузеров по протоколу WebRTC необходимо зайти на один и тот же сайт в интернете. Также, можно использовать посредника — сигнальный сервер.

Прежде чем начать передачу видео-данных, клиенты должны установить соединение. Процесс установления соединения между двумя клиентами заключается в обмене данными о соединении двух видов:

- текстовыми описаниями медиапоток в формате SDP

- ICE Candidates как часть SDP

Сигнальный сервер — посредник, позволяет передать данные о соединении от одного клиента к другому.

О проигрывании по WebRTC с Flussonic

Flussonic Media Server использует WebRTC для проигрывания видео с сервера Flussonic (источника) на клиентском устройстве или в программе (приемник). При этом Flussonic также играет роль сигнального сервера в процессе установления соединения, через который происходит обмен данными о соединении.

Почему мы используем именно WebRTC для обмена медиа-данными между двумя клиентами? Данный механизм позволяет нам достигнуть сверхнизкой задержки.

Таким образом, обмен видео по WebRTC через Flussonic Media Server нельзя считать peer-to-peer, правильнее будет называть это публикацией видео на Flussonic Media Server по WebRTC и проигрыванием видео с Flussonic сервера по WebRTC.

На схеме ниже показан процесс обмена данными для установления соединения между Flussonic и клиентским приложением, для проигрывания видео:

Как организовать проигрывание публикуемых потоков по WebRTC

Стороны должны обмениваться SDP через посредника (а это сигнальный сервер - Flussonic), после чего начать передачу данных напрямую. В случае проигрывания видео инициировать процесс и отправлять приглашение (SDP offer) нужно со стороны сервера Flussonic (источника).

Соединение устанавливается по WebSocket, а видео передается по RTP.

На клиенте нужно исполнить код для проигрывания видео из публикуемого потока. Для создания кода используйте библиотеку Flussonic WebRTC player. Описание классов библиотеки и код примера доступны на [nrm](#).

На сервере Flussonic должен быть сконфигурирован публикуемый поток, т.е. поток с источником `publish://`, в этот поток клиенты публикуют видео, а нам потребуется забрать его с сервера. URL этого потока нужно использовать в коде. Поддерживаются протоколы `ws` и `wss`.

```
stream STREAMNAME {  
  url publish://;  
}
```

Воспроизведение H265

H.265 (High Efficiency Video Compression — HEVC) — это новый формат видеосжатия, который постепенно приходит на смену H.264. Уменьшение размера файла по сравнению с предыдущим стандартом H.264 может достигать 25-50%. При этом качество изображения остается высоким. Поддерживаются форматы кадра до 8K (UHDTV) с разрешением 8192×4320 пикселей.

Вместо макроблоков, которые применялись в H.264, в HEVC используются блоки с древовидной структурой кодирования. Выигрыш кодера HEVC — в применении блоков большего размера. Применение таких блоков повышает эффективность кодирования при одновременном сокращении времени декодирования.

H.265 подается как решение для размеров экранов выше чем FullHD и поддерживается на большом количестве энкодеров: софтверных, обычных аппаратных (Nvidia NVENC, Intel QSV) и железных аппаратных. Также, H.265 можно встретить на спутниковом телевидении и IP камерах.

Поддержка H.265 в браузерах

Важно! Из десктопных браузеров показывать H.265 сейчас фактически умеет только Microsoft Edge (версия 16 и выше) и Safari (версия 11 и выше). Из мобильных браузеров — Safari и Chrome для iOS, (версия 11.0 и выше).

InternetExplorer	Microsoft Edge	MozillaFirefox	GoogleChrome	Safari	iOSSafari					
OperaMini	ChromeforAndroid	UC BrowserforAndroid	SamsungInternet			+	+	-	-	+
+	-	-	-	-						

На смартфонах в данный момент H.265 скорее всего будет проигрываться на процессоре, таким образом сильно нагружая аккумулятор устройства.

H.265 также проигрывается на телевизионных приставках и SmartTV.

Поддержка H265 в протоколах

В протоколе HLS формат H.265 поддерживается уже очень давно.

В протоколе MPEG-TS формат H.265 поддерживается.

В протоколе RTSP H.265 поддерживается. Есть упаковка и в SDP, и в RTP. Остается старый нюанс с передачей bframes по RTSP, но это отдельная головная боль.

Поддержка H265 в плеерах

Последние версии VLC Media Player имеют встроенную поддержку формата H.265.

Транскодер

Транскодирование необходимо для того, чтобы:

- создать мультибитрейтный поток
- изменить параметры видео — кодек и битрейт потока, размер картинки
- наложить логотип.

У Flussonic Media Server есть встроенный транскодер. Он поддерживает транскодирование на графическом процессоре или же с использованием центрального процессора.

Содержание:

- Установка транскодера
- Настройка транскодера через файл
- Настройка транскодера в веб-интерфейсе
- Примеры настройки параметров транскодирования потока
- Список опций транскодирования
- Аппаратное ускорение
- Наложение логотипа
- Добавление пресета
- Предотвращение моргания при смене размера изображения

Поддерживаются протоколы RTMP, RTSP, MPEG-TS (по HTTP и UDP).

Протокол HLS поддерживается частично — некоторые источники могут не транскодироваться. Работоспособность вашего источника HLS с транскодером необходимо каждый раз проверять самостоятельно.

При транскодировании на NVENC транскодер может обрабатывать потоки с 10-битной глубиной цвета.

Важно

Транскодирование является крайне ресурсоемким процессом (по CPU) и включает в себя следующие этапы:

- Декодирование исходного потока.
- Обработка и кодирование сырого потока в соответствии с заданными параметрами.

Рассчитывайте от 5 до 20 каналов на один сервер в зависимости от настроек.

Установка транскодера

Для транскодирования на GPU Nvidia NVENC никаких дополнительных пакетов устанавливать не требуется (в актуальных версиях Flussonic).

Но если вы используете для транскодирования центральный процессор (CPU), то нужно установить пакет flussonic-transcoder:

```
apt-get -y install flussonic-transcoder
```

Пакет устанавливается из того же репозитория, что и пакет flussonic.

Настройка транскодирования через файл

Настройки транскодирования задаются в настройках потока в конфигурационном файле. Их можно также указать в веб-интерфейсе.

Рассмотрим, как указать их в виде текстовой строки в конфигурационном файле `/etc/flussonic/flussonic.conf`.

Важно! Для транскодера важно соблюдать последовательность опций. Сначала указываются все параметры видео, затем глобальные опции и аудио. В примерах мы покажем правильную компоновку опций.

Транскодер для входящего потока включается следующим образом:

```
stream ort {  
url udp://239.0.0.1:5000;  
transcoder vb=2048k size=1280x720 preset=fast ab=128k;  
}
```

Для применения настроек не забудьте выполнить

```
/etc/init.d/flussonic reload
```

Настройка транскодирования с помощью веб-интерфейса

Чтобы включить транскодирование и задать его параметры через веб-интерфейс Flussonic Media Server:

В меню Media выберите канал, который необходимо транскодировать.

Откройте вкладку Transcoder. Поля для задания настроек станут активными после того, как вы добавите выходной видеопоток (см. следующий шаг).

Нажмите кнопку Add video track и (опционально) выберите битрейт, чтобы появился диалог настройки параметров видео:

Появятся поля для выбора свойств видео дорожек.

Если вы используете специальные аппаратные средства для ускорения транскодирования, укажите это:

Hardware acceleration — включает аппаратное кодирование. Аппаратное кодирование позволяет транскодировать значительно больше потоков на одном сервере.

Deinterlace mode — устраняет чересстрочность. Деинтерлейсинг необходим для комфортного просмотра ТВ на ПК/мобильных устройствах.

В этом поле вы видите методы, доступные для выбранного типа транскодера (CPU, Nvidia NVENC, и др.)

Device ID — задаёт номер видеокарты NVENC.

Seamless source change — предотвращает мигание при смене источника видеопотока.

Hardware acceleration — включает аппаратное кодирование. Аппаратное кодирование

позволяет транскодировать значительно больше потоков на одном сервере.

Deinterlace mode — устраняет чересстрочность. Деинтерлейсинг необходим для комфортного просмотра ТВ на ПК/мобильных устройствах.

В этом поле вы видите методы, доступные для выбранного типа транскодера (CPU, Nvidia NVENC, и др.)

Device ID — задаёт номер видеокарты NVENC.

Seamless source change — предотвращает мигание при смене источника видеопотока.

Чтобы оставить параметры видео, которые были у входного потока, отметьте Copy from origin.

Чтобы задать свои уникальные параметры для выходных видео дорожек, заполните форму на экране. Ниже описание опций.

В случае отсутствия требуемых опций на экране, можно переключиться в режим текстовой строки и указать опции вручную (см. далее на этой странице).

Параметры кодирования видео:

Video bitrate — видео-битрейт дорожки.

Video width — ширина в пикселах.

Video height — высота в пикселах.

Video codes (H.264 | H.265 | MP2V) — видео-кодек.

Preset — влияет на качество и скорость доставки. О пресетах см. в Presets на этой странице.

Чтобы получить на выходе видео с теми же характеристиками, что и на входе, отметьте Copy from origin.

Advanced options:

Profile (baseline | main | high) — ограничивает выходной поток одним профилем H.264.

Level — может использоваться для совместимости с устаревшими устройствами проигрывания.

Frames per second — частота кадров.

Reference frames — референсные кадры, использующиеся при межкадровой компрессии для ссылок на последующие кадры. Лучшее качество видео будет при большем количестве референсных кадров.

B-frames — значения 0 | 1 | 2 | 3 соответствуют последовательностям кадров: IP | IBP | IBBP | IBVBP.

Frames in GOP — количество кадров в GOP. Flussonic транскодирует поток, создавая каждый GOP точно такого размера, как указано в этой настройке.

Disable CGOP — разрешает транскодеру разбивать выходной поток на GOP с немного различающимся количеством кадров, но в районе заданного в опции Frames in GOP. Эта настройка применима только к транскодированию на процессоре. Иногда это помогает уменьшить трафик.

Logo options:

Укажите путь к файлу с логотипом (сделанному с учетом размера выходного видео) в поле The path to the track's alogo. Для каждого видео трека подготовьте индивидуальный файл с логотипом для предотвращения растягивания логотипа в результирующем видео. Описание alogo см. в опциях

Затем выберите место размещения логотипа на видео.

Audio settings:

Audio bitrate — битрейт аудио дорожки.

Sample rate (bypass|0|8000|16000|32000|44100|48000)

Codec (aac|opus|mp2a) — аудио-кодек (по умолчанию используется кодек AAC).

Channels — количество аудио-каналов в выходном потоке.

Чтобы получить на выходе аудио с теми же характеристиками, что и на входе, отметьте Copy from origin.

Other settings:

Speed up the transcoder by running it as part of Flussonic (use with caution). По умолчанию транскодер на сервере выполняется в отдельном от Flussonic процессе. Такое поведение более надежно для бесперебойной работы Flussonic Media Server. Если вы включаете опцию Speed up the transcoder by running it as part of Flussonic (external=false), то транскодер будет выполняться в одном процессе с Flussonic Media Server. Это ускоряет кодирование, особенно при кодировании аудио или при кодировании с использованием Nvidia. Однако ошибка транскодера может привести к остановке работы Flussonic Media Server.

Для кодирования с использованием центрального процессора отображаются дополнительные настройки:

Decoder threads — количество потоков на декодере. Эта настройка применима только к транскодированию на процессоре. Используйте её только после рекомендации от нашей службы поддержки клиентов.

Большинства типов транскодера позволяет делать обрезку видео. Обрезка позволяет получить на выходе только часть площади изображения.

Crop-X и Crop-Y — координаты верхнего левого угла результирующего изображения, относительно исходного (т.е. если (0,0) — это левый верхний угол исходного изображения).

Crop-Width — ширина изображения.

Crop-Height — высота изображения.

Для Nvidia NVENC можно удалять дублирующие кадры после устранения чересстрочности:

Drop second (field) — удаляет второе поле, получившееся после устранения чересстрочности, предотвращая тем самым повышенный битрейт. По умолчанию выключено. В конфигурационном файле этот параметр называется deinterlace_rate. Если вы отметили Drop second, то deinterlace_rate = frame и fps остается прежним, в противном случае deinterlace_rate = field и fps увеличивается после транскодирования в два раза.

Сохранение настроек

Чтобы сохранить новые значения, нажмите Save.

Чтобы удалить все настройки и отключить транскодер для этого потока, нажмите Disable transcoder.

Примеры настройки транскодирования потока

В примерах ниже используются иллюстрации, на которых группы настроек выделяются разными цветами:

синим — параметры кодирования видео, которые обязательно нужно указать.

желтым — параметры кодирования аудио, которые обязательно нужно указать.

белым — необязательные параметры.

Пример настройки параметров одного потока:

Пример:

`vb=2048k preset=fast ab=128k`

Примеры настройки параметров мультибитрейтного потока:

Пример:

`vb=2048k preset=fast vb=700k size=720x576 preset=fast vb=300k size=320x240 preset=fast ab=128k`

Пример:

`vb=copy vb=1600k size=1280x720 preset=fast vb=400k size=640x480 preset=fast ab=copy`

Опции

Опции видео

vb

vb (video bitrate) — параметр, задающий битрейт видео дорожки. Задаётся в виде числового значения (1000k, 1500k, 2000k и т.д.). Значение должно обязательно заканчиваться на k. Каждое указание опции vb создает новую видео дорожку в выходном потоке.

Пример:

`vb=2048k ab=128k`

Опция vb=copy сохраняет параметры оригинального потока, то есть просто копируется в исходящий поток.

preset

preset — аналог encoder preset из стандартного набора ffmpeg. Влияет на качество и скорость загрузки. Указывается отдельно для каждого видеопотока. По умолчанию — medium.

Пример:

`vb=2048k preset=fast ab=128k`

Подробнее о пресетах ниже.

size

size — задает размеры видео. Указывается отдельно для каждого видеопотока.

Пример:

`vb=2048k size=1280x720 ab=128k`

logo

logo — позволяет наложить логотип. Логотип добавляется до изменения размера изображения видео.

Пример:

`vb=2048k size=1280x720 logo=/path/to/file.png@10:10 ab=128k`

Подробнее о параметрах наложения логотипа ниже.

alogo

`alogo` — позволяет наложить логотип. Логотип добавляется после изменения размера изображения видео. Это предотвращает растягивание логотипа, которое может произойти при добавлении логотипа при помощи опции `logo`. Для каждого результирующего разрешения видео нужно подготовить отдельный файл логотипа.

Пример:

`vb=2048k size=1280x720 alogo=/path/to/file.png@10:10 ab=128k`

Подробнее о параметрах наложения логотипа ниже.

vcodec

`vcodec` — позволяет задать видео кодек. По умолчанию используется H.264. Flussonic Media Server позволяет кодировать в H.265 (hevc) или mp2v. Указывается отдельно для каждого видеопотока.

Кодек mp2v недоступен при использовании аппаратного кодирования.

Пример:

`vb=2014k vcodec=mp2v ab=128k`

fps

`fps` — задает частоту кадров. Указывается отдельно для каждого видеопотока.

Пример:

`vb=2048k fps=25 ab=128k`

deinterlace

`deinterlace` — устраняет чересстрочность. При интенсивной смене кадров в некоторых видеоплеерах могут наблюдаться продольные горизонтальные полосы. Деинтерлейсинг необходим для комфортного просмотра ТВ на ПК/мобильных устройствах. Указывается один раз и действует сразу на все видеопотоки.

Пример:

`vb=2000k deinterlace=1 ab=128k`

refs

`refs` — количество референсных кадров. Указывается отдельно для каждого видеопотока.

Пример:

`vb=2000k refs=6 ab=128k`

bframes

`bframes` — позволяет отключить b-frames. Это может понадобиться, например, при вещании в RTSP. Указывается один раз и действует сразу на все видеопотоки.

Пример:

vb=2000k bframes=0 ab=128k

hw

hw — включает аппаратное кодирование. Указывается в настройках потока.

Пример:

vb=2048k hw=nvenc ab=128k

crop

crop — позволяет обрезать видео. Доступно только при аппаратном транскодировании на NVENC: hw=nvenc. Указывается отдельно для каждого видеопотока.

Использование: crop=x:y:width:height, где:

x:y — координаты левого верхнего угла выходного видео в пределах размеров входного видео,

width — ширина выходного видео

height — высота выходного видео.

Пример:

vb=2048k hw=nvenc crop=0:0:100:100 ab=128k

aspect

aspect — изменяет соотношение сторон видео. Указывается отдельно для каждого видеопотока.

Важно! В данный момент изменение соотношения сторон не поддерживается для аппаратного кодирования с использованием Intel QuickSync (hw=qsv).

Пример:

vb=2048k hw=nvenc aspect=12:5 ab=128k

force_original_aspect_ratio

force_original_aspect_ratio=true — сохраняет соотношение сторон видео путем добавления черных полос. Опция полезна если вы хотите сохранить разрешение на выходе при переключении между разными источниками.

Пример:

vb=2048k size=1280x720 force_original_aspect_ratio=true

g

g=150 — устанавливает количество кадров в одном GOP. Flussonic транскодирует поток, создавая каждый GOP точно такого размера, как указано в этой настройке.

Пример:

vb=2048k fps=25 size=1280x720 g=150

Чтобы разрешить делать GOP не обязательно строго одного размера, используйте в дополнение к этой опции опцию disable_cgop.

disable_cgop

disable_cgop=1 — разрешает транскодеру разбивать выходной поток на GOP с немного

различающимся количеством кадров, но в районе заданного в опции g. Эта настройка применима только к транскодированию на процессоре (а не на аппаратном транскодере). Иногда это помогает немного уменьшить трафик.

Пример:

```
vb=2048k fps=25 size=1280x720 g=150 disable_cgor=1
```

Любые параметры видео

Например, -profile:v high -level 4.1 в конфигурации Flussonic Media Server будет выглядеть так: profile=high level=4.1.

Опции аудио

ab

ab — задает битрейт аудио. Указывается только один раз (даже если у вас мультибитрейтное видео). Значение должно обязательно заканчиваться на k.

Пример:

```
vb=2048k ab=128k
```

acodec

acodec — задает аудио кодек. Доступные значения: ac3, mp2a. По умолчанию, все аудиопотоки пережимаются в AAC.

Пример:

```
vb=2014k vcodec=mp2v ab=128k acodec=mp2a
```

ar

ar — задает sample rate, частоту дискретизации. Пример: ar=44100.

Пример:

```
vb=2014k ab=128k ar=44100
```

ac

ac — задает количество аудио-каналов.

Пример:

```
vb=2014k ab=128k ac=1
```

Другие опции

seamless=true

seamless=true — предотвращает мигание, которое может возникать при изменении источника потока, особенно если размер изображения из нового источника не такой, как из предыдущего. Подробнее

Пример:

```
transcoder vb=3000k size=1280x720 ab=copy seamless=true
```

Аппаратное ускорение

С помощью аппаратного транскодера можно серьезно увеличить количество транскодируемых потоков на одном сервере.

Flussonic Media Server поддерживает Nvidia NVENC и Intel Quick Sync технологии.

Подробнее про установку и настройку аппаратного ускорения описано в разделах Аппаратное транскодирование на Nvidia NVENC и Intel Quick Sync Video.

Наложение логотипа

Flussonic Media Server может накладывать логотип на видео:

Пример:

```
vb=2048k preset=fast logo=/path/to/file.png@10:10 ab=128k
```

10:10 — это координаты от левого верхнего угла экрана. Для размещения в других частях экрана может понадобиться написать немного более сложную формулу.

Для размещения в центре:

```
vb=2048k preset=fast logo=/path/to/file.png@(main_w-overlay_w-10)/2:(main_h-overlay_h-10)/2 ab=128k
```

Для размещения в левом нижнем углу:

```
vb=2048k preset=fast logo=/path/to/file.png@10:(main_h-overlay_h-10) ab=128k
```

Для размещения в правом верхнем углу:

```
vb=2048k preset=fast logo=/path/to/file.png@(main_w-overlay_w-10):10 ab=128k
```

Для размещения в правом нижнем углу:

```
vb=2048k preset=fast logo=/path/to/file.png@(main_w-overlay_w-10):(main_h-overlay_h-10) ab=128k
```

Внимание! Flussonic Media Server может накладывать логотип только при кодировании на CPU и NVENC.

Выбор пресета

Preset — это набор оптимальных параметров, влияющих скорость кодирования и качество. Чем лучше качество, тем дольше по времени кодируется файл, и наоборот.

Это означает, что лучшего качества при кодировании можно достичь используя более медленный preset, но кодирование займет больше времени. Используйте «медленные» пресеты, если для вас важнее качество, а не скорость.

Для транскодирования с использованием ЦП поддерживаются следующие значения (в порядке убывания скорости):

- ultrafast
- superfast
- veryfast
- faster
- fast
- medium

slow
slower
veryslow
placebo

По умолчанию используется medium. Пресет placebo не слишком полезен, его не нужно использовать.

При кодировании с использованием Nvidia NVENC поддерживается следующий набор пресетов:

default, slow, medium, fast, hp, hq, bd, ll, llhq, llhp, lossless, losslesshp.

При кодировании с использованием Intel QuickSync набор пресетов следующий:

veryfast, faster, fast, medium, slow, slower, veryslow.

Плавная смена источника

Опция "seamless=true" улучшает поток на выходе транскодера. Она предотвращает мигание, которое иногда может возникать при изменении источника потока, особенно если размер изображения у нового источника не такой, как у предыдущего. Например, когда в телеканал врезана реклама с другой шириной и высотой изображения или когда вы транскодируете серверный плейлист, состоящий из разных видео-поток и файлов.

С помощью этой опции транскодер "бесшовно" переключается на другой исходный формат без потери кадров. Это достигается за счет транскодирования с одинаковым размером изображения в результирующем видеопотоке. Все источники будут транскодированы с одним размером, который вы указали в опции size. Если size не указан, то размер будет как у картинки из самого первого источника, который был получен транскодером на вход.

Пример:

```
stream STREAM_NAME {  
    udp://239.0.0.1:5000;  
    transcoder vb=3000k size=320x200 ab=copy hw=nvenc seamless=true;  
}
```

Аппаратное транскодирование на Nvidia NVENC

Транскодирование видео на NVIDIA Nvenc

Flussonic Media Server умеет кодировать видео, используя GPU на видеокартах NVIDIA. Список поддерживаемых видеокарт можно найти на сайте Nvidia.

Также в системе должен быть установлен драйвер Nvidia версии не ниже 352.

Важно. При использовании NVIDIA Nvenc наш транскодер может обрабатывать 10-битные H.265 (HEVC) потоки.

Установка драйвера

Драйвер устанавливается из соответствующего пакета.

Ubuntu 16.04:

```
apt-get install nvidia-375
```

Для транскодирования потоков с 10-битной глубиной цвета потребуется драйвер не ниже 385.130.

Debian 9:

```
apt-get install nvidia-driver libnvidia-encode1
```

В sources.list должен быть включен компонент non-free.

Для других системах можно установить драйвер с официального сайта Nvidia. Инструкция по установке драйвера для Debian

Для работы с большим количеством транскодируемых потоков может потребоваться увеличение лимита на открытые файлы. Сделать это можно командой: `ulimit -n 4096`

Добавьте в файл следующие строки в файл: `/etc/security/limits.conf`:

```
* hard nofile 4096
```

```
* hard nofile 4096
```

Включение транскодера

Настройки транскодера можно задать:

В конфигурационном файле Flussonic `/etc/flussonic/flussonic.conf` в опциях потока, используя директиву `transcoder` с разными опциями.

В интерфейсе администратора в Media > выбрать поток > Transcoder.

Для включения аппаратного кодирования с использованием Nvenc необходимо прописать опцию `hw=nvenc`:

```
transcoder vb=2048k hw=nvenc ab=128k
```

Выбор кодека

По умолчанию используется H.264. При кодировании на Nvenc вы можете использовать H.265

(HEVC):

```
transcoder vb=2048k hw=nvenc vcodec=hevc ab=128k
```

Поддержка потоков с 10-битной глубиной цвета

Транскодер Flussonic может работать с 10-битными H.265 (HEVC) потоками при использовании NVIDIA Nvenc. Для этого необходимо использовать опцию `pix_fmt` и указать необходимый пиксельный формат.

Возможны следующие преобразования (на входе -> на выходе):

10-bit HEVC -> 8-bit HEVC

```
vb=3000k vcodec=hevc pix_fmt=yuv420p ab=128k
```

10-bit HEVC -> 8-bit H.264

```
vb=3000k pix_fmt=yuv420p ab=128k
```

10-bit HEVC -> 10-bit HEVC

```
vb=3000k vcodec=hevc ab=128k
```

8-bit HEVC или H.264 -> 10-bit HEVC

```
vb=3000k vcodec=hevc pix_fmt=yuv420p10 ab=128k
```

Необходимо использовать драйвера NVIDIA - версии 384.130 или выше, и версию Ubuntu - 16.04 или выше.

Выбор видеокарты

Если в системе установлено несколько видеокарт, то можно выбрать какую из них использовать для транскодирования. Для этого используется опция `deviceid`:

```
transcoder vb=2048k hw=nvenc deviceid=2 ab=128k
```

Номер видеокарты можно узнать при помощи команды `nvidia-smi`.

По умолчанию используется первая видеокарта `deviceid=0`.

Обрезка видео

Опция `crop` позволяет обрезать видео. Указывается отдельно для каждого видеопотока.

Использование: `crop=x:y:width:height`, где:

`x:y` — координаты левого верхнего угла выходного видео в пределах размеров входного видео,

`width` — ширина выходного видео

`height` — высота выходного видео.

Пример:

```
transcoder vb=2048k hw=nvenc crop=0:0:100:100 ab=128k
```

Декодирование на CPU

По умолчанию, декодирование также происходит на GPU. Чтобы использовать для декодирования центральный процессор, вместо `hw=nvenc` укажите `hw=nvenc2`:

```
transcoder vb=2048k hw=nvenc2 ab=128k
```

Деинтерлейсинг

Деинтерлейсинг (устранение чересстрочности) при использовании nvenc происходит по умолчанию. Но можно указать определенный метод при помощи опции deinterlace. Например, добавьте deinterlace=yadif, чтобы применить метод CUDA yadif:

```
stream test {  
    url file://vod/test.ts;  
    transcoder vb=4000k ab=128k deinterlace=yadif hw=nvenc;  
}
```

Все доступные для NVIDIA Nvenc методы можно увидеть в UI в настройках транскодера для потока, в поле Deinterlace mode.

В случае nvenc2 (использование CPU) деинтерлейсинг следует включить с помощью опции deinterlace=yes.

Для отключения дорогостоящего деинтерлейса укажите deinterlace=0.

Прочие параметры, такие как size, preset, bframes, level, используются аналогично CPU транскодеру.

Возможные значения параметра preset: hq, hp, bd, ll, llhq, llhp. По умолчанию — hp.

Статистика производительности Nvidia

Вы можете собирать статистику о работе GPU Nvidia, если включите сохранение статистики в базе данных Pulse. Чтобы начать сохранять данные, добавьте в файл конфигурации Flussonic следующую директиву:

```
nvidia_monitor true;
```

Чтобы прекратить сохранять статистику по Nvidia, обновите конфиг:

```
nvidia_monitor false;
```

Чтобы посмотреть данные на графике, составьте запрос и выполните его в интерфейсе администратора в разделе Pulse, поле Custom query.

В запросе используйте основные метрики (метрики температуры могут не поддерживаться видеокартой):

gpu_pwr — энергопотребление (в Вт)

gpu_temp — температура графического процессора (в градусах Цельсия)

gpu_sm — использование SM (streaming multiprocessor) в %

gpu_mem — использование памяти в %

gpu_enc — использование энкодера в %

gpu_dec — использование декодера в %

gpu_usedmem — количество используемой видеопамяти в байтах или килобайтах.

Пример запроса:

```
sum:1m-avg:gpu_dec{from=-2h,gpu=nv0}
```

Другие метрики:

gpu_mclk, gpu_pclk — частота памяти и процессора (в МГц)

gpu_pviol, gpu_tviol — превышение энергопотребления (в %) и превышение температурного режима (в виде булева значения)

gpu_fb, gpu_bar1 — использование буфера кадров и памяти Bar1 (в Мбайтах)

gpu_sbecc, gpu_dbecc — ECC (количество агрегированных single-bit и double-bit ошибок ECC) и количество ошибок воспроизведения PCIe

gpu_pci, gpu_rxpci, gpu_txpci — PCIe Rx и Tx пропускная способность в Мбайт/с (для поколений Maxwell и выше).

Сколько каналов можно транскодировать на одной видеокарте?

Видеокарты серии GeForce, как правило, имеют ограничение в количестве одновременно кодируемых потоков равных двум. Профессиональные линейки QUADRO и TESLA не имеют подобных ограничений, однако максимальное количество транскодируемых потоков зависит от множества факторов, включая качество входных и выходных потоков (битрейт, разрешение видео, число кадров в секунду и т.д.), а также аппаратные ресурсы самого сервера.

При выборе модели видеокарты используйте сравнительную таблицу [nvidia](#)

Intel Quick Sync Video

Информацию о поддерживаемых платформах вы можете посмотреть в официальном GitHub аккаунте Intel:

<https://github.com/intel/media-driver#supported-platforms>

После установки QSV транскодер доступен через опцию `hw=qsv`:

```
stream ort {  
    url udp://239.1.1.10:5500;  
    transcoder vb=3000k hw=qsv ab=64k;  
}
```

Подробнее про настройку транскодера

Установка в Ubuntu

Мы подготовили .deb пакет, позволяющий легко установить поддержку QSV в вашу систему. Совместимость проверена только с Ubuntu 16.04.

```
apt install linux-base flussonic-qsv  
apt install linux-headers-4.15.0-20-generic linux-image-4.15.0-20-generic linux-modules-extra-4.15.0-20-generic libpciaccess0  
reboot
```

Установка в CentOS

Воспользуйтесь официальной инструкцией Intel:

<https://github.com/Intel-Media-SDK/MediaSDK#system-requirements>

Скриншоты

Flussonic Media Server умеет вырезать скриншоты из видео. С их помощью вы сможете:

- показать предпросмотр текущего транслируемого видео на веб-странице;
- оценить качество потока;
- захватить нужный момент времени;
- быстро искать в архиве нужный момент видео по скриншотам;
- создать страницу со скриншотами потока, чтобы быстро просмотреть сутки архивной записи;
- делать что вам угодно с маленькими статическими картинками из большого видео потока.

Flussonic Media Server делает скриншоты двумя принципиально различными способами:

Извлекает видео-кадры в виде JPEG-изображений и может сохранять их в архиве. Создание и хранение JPEG скриншотов потребляет много ресурсов. Подробнее.

Создает экономичные MP4 видео-скриншоты. В H.264-потоке с ключевыми кадрами уже есть сжатые изображения, для получения которых не нужна ресурсоемкая обработка потока.

Flussonic Media Server берет первый ключевой кадр из каждого сегмента и показывает его как MP4 видеофайл, состоящий из одного кадра. Подробности в разделе Видео-скриншоты.

О JPEG скриншотах

Flussonic Media Server делает операцию, которая достаточно требовательна к ресурсам процессора: берётся первый ключевой кадр из сегмента, раскодируется и кодируется обратно в JPEG изображение. Выглядит просто, но когда у вас 300 потоков, то этот процесс требует много процессорного времени.

Flussonic позволяет оптимизировать нагрузку - меняя длительность сегмента, вы можете менять количество JPEG скриншотов. Тот факт, что Flussonic Media Server берёт только первый ключевой кадр из сегмента, означает, что если вы настроили длительность сегмента равной трём секундам, у вас будет 20 JPEG картинок в минуту. Если длительность сегмента установлена в 6 секунд, получится 10 JPEG изображений в минуту. Если вы получаете поток с IP камеры, у вас может быть 60 ключевых кадров в минуту, но Flussonic Media Server создаст меньшее количество JPEG-изображений.

Замечание. Когда вы включаете для потока запись архива, все эти JPEG-скриншоты пишутся на диск.

Можно облегчить нагрузку на процессор, используя загрузку скриншотов по заданному URL. Такой способ применяется с IP камерами, так как IP камеры создают свежий JPEG скриншот для показываемого видео. В этом случае Flussonic Media Server будет загружать JPEG изображения каждый раз, когда начинается новый сегмент.

Настройка генерации JPEG скриншотов

Для создания JPEG скриншотов Flussonic Media Server использует свой встроенный пакет.

Добавьте опцию `thumbnails` в конфиг потока:

```
stream ort {  
    url udp://239.255.0.100:1234;
```

```
thumbnails;  
}
```

Это запустит отдельный процесс flussonic-thumbailer. Возможно, он будет потреблять значительное количество ресурсов, но, к сожалению, это неотъемлемая черта процессов сжатия видео и изображений.

Настроить получение скриншотов можно также и в панели администратора в настройках потока (Media > выбрать поток > Output).

Настройка загрузки JPEG скриншотов по URL

Вы можете указать URL, по которому Flussnic Media Server будет загружать скриншоты, что уменьшит использование процессора. Многие камеры имеют специальный URL для скриншотов:

```
stream cam0 {  
    url rtsp://10.0.4.3:554/h264;  
    thumbnails url=http://10.0.4.3/cgi-bin/snapshot.cgi  
}
```

Вы можете попробовать найти этот URL в документации на вашу камеру, но в 99% случаев этой инструкции нет и вам придётся искать URL для скриншотов в Интернете.

Получение live скриншота

После того как вы включили скриншоты в настройках, нужно получить их.

URL для доступа к live-скриншоту такой:

<http://flussonic:8080/ort/preview.jpg> — последний скриншот пототока.

<http://flussonic:8080/ort/preview.mjpeg> — MJPEG поток скриншотов.

Мы рекомендуем никогда не использовать MJPEG, потому что это неконтролируемый способ передачи видео с очень высоким битрейтом. Вы можете получить MJPEG поток с битрейтом до 50% от исходного потока со скоростью 0.1 кадр в секунду. Используйте это только при необходимости.

Получение JPEG скриншотов из архива по времени GMT

Скриншоты автоматически сохраняются в архив. Они могут быть получены с помощью HTTP API.

Экономично по ресурсам получать JPEG скриншоты, указывая приблизительное GMT время в URL. Flussonic найдет ближайший сохраненный скриншот и вернет URL с точным временем.

```
curl -v 'http://192.168.2.3:8080/ort/2018/05/02/06/59/38.jpg'
```

```
...
```

```
< HTTP/1.1 302 Found
```

```
< Location: /ort/2018/05/02/07/00/40.jpg
```

По полученному времени находится реальный скриншот.

Подробно этот способ описан в DVR API, Получение JPEG скриншотов по времени GMT.

Получение JPEG скриншотов из архива по времени UTC

Внимание! Этот способ затратный по ресурсам, мы не рекомендуем его использовать. Лучше получать скриншоты по времени GMT, см. Получение JPEG скриншотов по времени GMT.

Сначала надо выбрать временной промежуток, для которого надо прочитать архив. Например, сейчас 21 апреля 2017, 13:10 GMT, что соответствует 1492780200 UTC. Если вы хотите получить скриншоты за последний час, надо запросить следующий URL:

```
curl 'http://flussonic:8080/clock/recording_status.json?from=1492776600&to=1492780200&request=brief_thumbnails'
```

По умолчанию Flussonic не включает в ответ список таймстемпов. Для их получения к запросу нужно добавить request=brief_thumbnails.

Ответ может быть таким:

```
[{"stream":"clock","ranges":[{"duration":3642,"from":1492776599}], "brief_thumbnails": [1492776599, 1492776605, 1492776617, 1492776629, 1492776641, 1492776653, 1492776665, 1492776677, 1492776689, 1492776701, 1492776713, 1492776725, ....]]
```

Вы получите длинный список таймстемпов, которые необходимо сконвертировать в пути к скриншотам. Например, 1492776605 будет преобразован в <http://flussonic:8080/clock/2017/04/21/12/10/05.jpg>.

Таким образом, сначала вы получаете список таймстемпов, а затем получаете сами скриншоты по сформированному URL.

Генерация JPEG по запросу

Иногда хранить все JPEG скриншоты на диске очень накладно, и вы можете попросить Flussonic Media Server генерироваться JPEG по требованию. В этом случае не нужно включать генерацию скриншотов в настройках потока.

Запросите URL для конкретного времени:

```
http://flussonic:8080/clock/2017/04/21/12/10/05-preview.jpg
```

и Flussonic Media Server прочитает один сегмент, возьмёт первый ключевой кадр и сгенерирует JPEG.

Этот способ может привести к непредсказуемой загрузке процессора и, поэтому, не рекомендуется.

Непредсказуемая нагрузка здесь означает, что её действительно трудно предсказать. С включенным процессом генерации JPEG у вас ровная умеренная нагрузка процессора и не более. С генерацией JPEG по запросу у вас может быть низкая нагрузка процессора, но в час наибольшей нагрузки может произойти её резкий рост и работа сервера станет нестабильной.

Мозаика

Несколько потоков можно "склеить" в одну мозаику и показывать как один поток. Создается мозаика с помощью транскодирования.

Сборка мозаики из потоков

Через веб-интерфейс Watcher можно включить только клиентскую мозаику из камер. Подробнее об этом написано в документации Watcher.

Чтобы создать серверную мозаику:

Установите пакет flussonic-transcoder:

Замечание. Пакет flussonic-transcoder необходим только в случае, если вы планируете использовать CPU для выполнения транскодирования. Если вы используете Nvidia NVENC, то он не нужен.

```
apt-get -y install flussonic-transcoder
```

Добавьте поток с источником url mosaic://...:

```
stream cam1 {
    url rtsp://...;
}
stream cam2 {
    url rtsp://...;
}
stream cam3 {
    url rtsp://...;
}
stream cam4 {
    url rtsp://...;
}
stream mosaic0 {
    url mosaic://cam1,cam2,cam3,cam4?fps=20&preset=ultrafast&bitrate=1024k&size=340x240&mosaic_size=16;
}
```

После mosaic:// идет через запятую список камер, которые будут использоваться в мозаике.

В опциях можно указывать настройки, которые будут использоваться в энкодере.

Опция fps=20 жестко указывает скорость видео. Для камер можно указывать fps=video, чтобы привязать кадры мозаики к первой камере.

Опция size=320x240 настроит размер каждой камеры в мозаике. Если от камеры поток с картинкой больше, то она будет уменьшена до этого размера.

Опция mosaic_size указывает, на сколько камер будет рассчитана мозаика. Это может быть удобно для того, чтобы фиксировать размер мозаики.

VOD — вещание файлов

Содержание:

- Контейнеры и кодеки
- Создание VOD-локации
- Управление файлами через веб-интерфейс
- Как вставить плеер на сайт
- Как проиграть файл по разным протоколам
- Как проиграть файл из административного интерфейса
- Мультикаталоговая раздача
- Настройки VOD
- Мультиязыковой стриминг
- Экспорт трека с субтитрами в виде SRT
- Адаптивный стриминг (мультибитрейт)
- Подготовка файла к вещанию (транскодирование)
- Рестриминг VOD

Контейнеры и кодеки

Flussonic Media Server умеет раздавать видео из файлов в контейнерах MP4 с видео кодеками H.264, H.265 (HEVC), VP6 и аудио кодеками AAC, MP3, AC3, PCMA, PCMU.

Контейнер	Видео	Аудио
MP4 (.mp4, .f4v, .mov, .m4v, .mp4a, .3gp, .3g2)	H.264, H.265	MP3, AAC (все профили)

Создание VOD-локации

Для того, чтобы транслировать файлы, например, из каталога /movies, необходимо выбрать произвольное, неиспользуемое имя для префикса и вписать следующую строку в конфигурационный файл /etc/flussonic/flussonic.conf:

```
file vod {  
    path /movies;  
}
```

Здесь vod — префикс месторасположения файлов, или VOD-локация.

Настроить вещание файлов можно и через веб-интерфейс. Создайте VOD-локацию в списке на странице Media:

Теперь файлы будут отдаваться по префиксу vod по протоколам HLS, DASH, HDS, RTMP, RTSP.

Для одного файлового префикса можно указывать несколько источников, включая облачные

хранилища. Подробнее про настройку нескольких путей для одного префикса.

Управление файлами через веб-интерфейс

В Flussonic Media Server встроен полноценный менеджер файлов, позволяющий загружать и просматривать файлы как на локальном диске, так и в облачном хранилище.

После добавления файлового префикса в конфигурационном файле или через веб-интерфейс надо перейти во вкладку VOD:

Создайте поддиректории:

После того как вы выбрали нужную директорию, залейте в нее файлы:

Кликнув на файл, можно просмотреть его из этого интерфейса.

При наведении мышки на его имя появляется возможность удалить файл.

Как вставить плеер на сайт

В Flussonic Media Server есть специальная страница — `embed.html`, которая предназначена для вставки видео на сайт или просмотра видео через браузер. Она доступна по ссылке:

`http://HOSTNAME/vod/bunny.mp4/embed.html`

Страница автоматически определяет браузер и выбирает поддерживаемый протокол. Для большинства устройств на сегодня — HLS.

Подробнее в статье «Вставка видео на сайт (`embed.html`)».

Как проиграть файл по разным протоколам

Для того, чтобы проиграть файл лежащий на диске по пути `/movies/elementary/s01e02.mp4` надо указать следующие источники для плееров:

HDS (StrobeMediaPlayback или другие OSMF-плееры)

`http://HOSTNAME:8080/vod/elementary/s01e02.mp4/manifest.f4m`

HLS (iOS, Android, STB)

`http://HOSTNAME:8080/vod/elementary/s01e02.mp4/index.m3u8`

RTMP (Jwplayer, Flowplayer, Uppod)

`application: rtmp://HOSTNAME/vod, path: elementary/s01e02.mp4`

MSS

`http://HOSTNAME:8080/vod/elementary/s01e02.mp4.isml/manifest`

Как проиграть файл из админки

Необходимо выбрать файл в списке файлов конкретного VOD:

После можно дополнительно выбрать нужный протокол и нажать `click to play`.

Мультикаталоговая раздача

Иногда при большом трафике стриминга принимается решение монтировать жесткие диски в

системе не в RAID, а в режиме JBOD. То есть каждый жесткий диск виден в системе отдельно и монтируется в отдельный каталог.

Можно указать Flussonic Media Server одну файловую зону на несколько каталогов. Это даст возможность перемещать файл между устройствами без изменения URL-адреса:

```
file vod {  
    path /mount/disk1;  
    path /mount/disk2  
}
```

Настройки VOD

Эти настройки можно указывать внутри директивы file, которая хранит настройки вещания файлов. Мы называем их опции.

file

file vod { path /storage;} Конфигурации локации для проигрывания файлов.

cache

cache /ssd misses=5 2d 40G; Все запросы к файлам будут кешироваться в папке /ssd сроком не более чем 2 дня и с лимитом в 40G. Кеширование включится, если к одному файлу было более 5 некешированных запросов.

domain

domain host.ru; Указание доменов, на которых можно проигрывать это видео. Не работает для клиентов, которые не передают Referer.

domains

domains host1.ru *.host1.ru; Указание доменов, на которых можно проигрывать это видео. Не работает для клиентов, которые не передают Referer.

path

path /storage; path s3://key:secret@s3.amazonaws.com/bucket/; Путь для поиска файлов. Можно указать несколько путей.

read_queue

read_queue 100; Количество одновременных обращений к диску для этого префикса.

download

download; Эта опция разрешает скачивание файла и обращение к нему с Range запросами.

max_readers

max_readers 10; Опция настраивает максимальное количество одновременных дисковых запросов ко всему префиксу.

thumbnails

thumbnails offset=10; Опция включает генерацию скриншотов к файлам с опциональным

указанием секунды видео.

auto_mbr

auto_mbr; Включает автоматическое создание мультибитрейтного HLS плейлиста из нескольких файлов с разным битрейтом.

Мультиязыковой стриминг

Протоколы HLS и HDS дают возможность переключать языки. Flussonic Media Server включит эту опцию автоматически, если вы просто добавите дополнительные языковые дорожки в mp4 файл.

Для включения субтитров, надо также просто добавить субтитры в формате tx3g в виде дорожек в MP4 файл.

Экспорт трека с субтитрами в виде SRT

Flussonic Media Server может отдать субтитровый трек в mp4 в виде SRT, необходимом для некоторых flash-плееров:

<http://192.168.2.3:8080/vod/video.mp4/track-t1.srt>

Адаптивный стриминг (мультибитрейт)

Для того, чтобы обеспечить комфортный просмотр пользователям, подключенным на разных скоростях к интернету, можно воспользоваться адаптивным стримингом. Flussonic поддерживает два способа:

Использование нескольких файлов с одинаковым содержимым, но с разными качествами.

Flussonic автоматически создает один мультибитрейтный плейлист для воспроизведения этих файлов как одного ресурса. Как это настроить

Использование одного файла, содержащего дорожки разного качества.

Для этого надо создать мультибитрейтный MP4 файл и запросить для него манифест для проигрывания. Дальнейшее Flussonic Media Server сделает сам. В разделе про подготовку файлов подробнее описано, как делать мультибитрейтные файлы.

Подготовка файла к вещанию (транскодирование)

Ниже приведен принципиальный пример, как сделать мультибитрейтный файл с помощью ffmpeg:

```
ffmpeg -i bunny.mp4 \
-mar 0:0 -c:v copy \
-mar 0:0 -c:v libx264 -b:v 150k \
-mar 0:0 -c:v libx264 -b:v 100k \
-mar 0:1 -c:v libx264 -b:v 50k \
-mar 0:1 -c:a copy \
-mar 0:1 -c:a copy \
```

-y out.mp4

Более подробно процесс описан в разделе «Подготовка к вещанию».

Рестриминг VOD

Библиотека VOD имеет большой объём, и её копирование между серверами дорого по ресурсам, но Flussonic может выполнять рестриминг видеофайлов на соседние серверы Flussonic. Это сэкономит не только время, но и место, требуемое для хранения VOD контента. Сэкономленные ресурсы можно использовать для включения кэширования VOD контента, что увеличит производительность VOD-рестримера.

Пример конфигурации VOD-каталога:

Основной сервер VOD

```
file vod {  
    path priv;  
    download;  
}
```

Рестример VOD

```
file vod {  
    path http://FLUSSONIC-SOURCE/vod;  
    cache /mount/cache 500G misses=2;  
}
```

Кэш

Для ускорения раздачи VOD можно использовать кэш.

Для оригинальных файлов из облака или с HTTP сервера используется cache.

Для файлов на SSD диске приходится использовать схему с промежуточным SSD кэшированием, т.е. кэшировать сегменты с помощью segment_cache.

Файловый кэш на SSD

Если файлы берутся из облака или с простого HTTP сервера (например, другого Flussonic Media Server), то нужно настроить файловый кэш. То есть зону, куда файл будет скачиваться целиком.

Этот механизм позволит выстроить полноценный файловый CDN из нескольких Flussonic Media Server, поскольку даже скачивание файла с вторичного сервера приведет к кэшированию файла на нём.

И, конечно, Flussonic Media Server не будет скачивать одни и те же данные с источника дважды.

Важно! Для кэширования файлов не используйте разделы SSD, смонтированные с опцией noatime.

Для настройки файлового кэша пропишите в конфигурации:

```
file vod {  
    path http://big-central-origin.mycdn.tv/vod;  
    cache /mount/ssd 400G;  
    download;  
}
```

При такой конфигурации можно проигрывать файлы, скачивая их с указанного сервера. Они будут сохраняться в кэше, причем сохраняться будет только та часть данных, которая была запрошена.

Кэширование по количеству обращений

Для файлов можно указать условие для помещения файла в кэш — этим условием будет количество запросов файла клиентами.

Опция misses=3 говорит серверу, что файл нужно поместить в кэш, если к нему было более трех обращений:

```
file vod {  
    path http://big-central-origin.mycdn.tv/vod;  
    cache /mount/ssd 400G misses=3;  
    download;  
}
```

Сегментный кэш для SSD

На сегодняшний день один из самых главных способов радикально ускорить работу сервера по раздаче контента с дисков — использование SSD накопителей.

Так как твердотельные накопители стоят существенно дороже, чем обычные диски, то

приходится использовать схему с промежуточным SSD кэшированием.

Flussonic Media Server умеет самостоятельно кэшировать на диске запрашиваемые отрезки видео по протоколам HLS и HDS, что позволяет сильно ускорить раздачу.

Для настройки сегментного кэша пропишите в конфигурации:

```
file vod {  
    paths /mount/hdd1 /mount/hdd2 /mount/hdd3;  
    segment_cache /mount/ssd1 20G 48h misses=2;  
}
```

При такой конфигурации Flussonic Media Server будет сам поддерживать заполнение кэша на уровне 20 гигабайт, стирая файлы старше двух суток и кэшируя только те файлы, к которым было больше двух обращений.

Важно! До версии 4.6.14 эта директива называлась `cache`.

Важно! Мы не рекомендуем использовать `segment_cache` для обычных дисков!

Облачное хранение

Flussonic Media Server умеет транслировать видеофайлы, которые находятся на облачных хранилищах (Amazon S3 или OpenStack Swift) или на HTTP серверах.

Вещание с HTTP сервера

```
file http {  
    path http://storage/prefix;  
}
```

Можно передать параметры в query string. Это может быть нужно, если сервер проверяет какой-нибудь параметр в query string. Например:

```
file http {  
    path http://storage/prefix?key=12345;  
}
```

Например, при обращении к файлу vod/bunny.mp4, Flussonic Media Server превратит запрос в http://storage/prefix/bunny.mp4?key=12345.

Вещание с Amazon S3

ACCESS_KEY и SECRET_KEY — это ключи которые можно взять из профиля на Amazon AWS.

```
file public {  
    path http://s3.amazonaws.com/publicbucket;  
}  
file private {  
    path s3://ACCESS_KEY:SECRET_KEY@s3.amazonaws.com/privatebucket;  
}
```

Вещание из Swift storage

```
file swift {  
    path swift://user=USER&password=PASSWORD&region=1@swift-proxy/bucket;  
}
```

Транскодирование файлов

Для того, чтобы обеспечить комфортный просмотр видео пользователям, подключенным на разных скоростях к интернету, можно воспользоваться адаптивным стримингом. Для этого надо сделать мультибитрейтный MP4 файл и запросить для него манифест. Дальнейшее Flussonic Media Server сделает сам.

Ниже мы рассмотрим, как настроить компьютер и создать мультибитрейтный файл.

Установка программ

Вам нужно иметь установленный ffmpeg и кодеки. Процесс установки отличается для разных операционных систем.

Инструкция для Windows

Скачайте ffmpeg: <http://ffmpeg.zeranoe.com/builds/>

Распакуйте, например, в C:\ffmpeg.

Добавьте в PATH соответствующие два каталога (C:\ffmpeg и C:\ffmpeg\bin). Чтобы добавить в PATH в Windows 8.1: нажмите «Win+Pause» > «Дополнительные параметры системы» > «Переменные среды» > таблица «Системные переменные» > строка с названием переменной «PATH» > в самое начало значения переменной добавьте "C:\ffmpeg;C:\ffmpeg\bin;" (без кавычек).

Скачайте и установите K-Lite Mega Codec Pack: http://www.codecguide.com/download_k-lite_codec_pack_mega.htm. После запуска инсталлятора будет предложено несколько вариантов установки, нужно выбрать самый полный («Lots of stuff»).

Мы рекомендуем поставить уже собранный ffmpeg отсюда: <http://johnvansickle.com/ffmpeg> Или любой другой собранный ffmpeg с официального сайта: <https://www.ffmpeg.org/download.html>

С большой вероятностью ffmpeg, идущий в вашем дистрибутиве либо не будет уметь кодировать H264, либо будет слишком старым, чтобы подошли наши инструкции (вообще любые инструкции в интернете, которые основываются на возможностях свежих версий ffmpeg), или произойдет еще какая-нибудь другая неприятность.

Конструирование команды для ffmpeg

Допустим, у нас есть видеофайл "h.m4v" с двумя звуковыми дорожками (английская, русская) и двумя субтитрами (английские, русские).

Вначале нужно посмотреть, из чего он состоит, какие внутри есть потоки. Для этого в консоли печатаем:

```
ffmpeg -i h.m4v
```

На экран выведется огромное количество текста, из которого важно вот это:

Stream #0:0(eng): Video: h264 (Constrained Baseline) (avc1 / 0x31637661), yuv420p, 640x360 [SAR 1331:1000 DAR 2662:1125], 1800 kb/s, 23.98 fps, 23.98 tbr, 25k tbn, 180k tbc

Metadata:

creation_time : 2013-01-14 14:46:26

handler_name :

Stream #0:1(rus): Audio: aac (mp4a / 0x6134706D), 48000 Hz, stereo, s16, 127 kb/s

Metadata:

creation_time : 2013-01-14 14:47:59

handler_name :

Stream #0:2(eng): Audio: aac (mp4a / 0x6134706D), 48000 Hz, stereo, s16, 127 kb/s

Metadata:

creation_time : 2013-01-14 14:48:19

handler_name :

Stream #0:3(rus): Subtitle: mov_text (tx3g / 0x67337874)

Metadata:

creation_time : 2013-01-14 14:48:38

handler_name :

Stream #0:4(eng): Subtitle: mov_text (tx3g / 0x67337874)

Metadata:

creation_time : 2013-01-14 14:48:38

handler_name :

Секции Stream, а в них - номер потока (0:0,0:1,0:2,0:3,0:4), тип потока (video, audio, subtitles) и язык (в данном случае eng и rus).

В файле, который мы стремимся получить, будут всё те же потоки, но увеличенное количество видеопотоков. Представим, что нам нужно получить видеофайл с возможностью выбора из 3 различных качеств видео.

3 потока видео + 2 потока аудио + 2 потока субтитров = 7 потоков всего.

Теперь приступаем к конструированию команды для ffmpeg.

Первая строка такая:

```
ffmpeg -i "/home/olegchir/temp/h.m4v" \
```

Обратите внимание на символ \. Это - перевод строки в Linux. В Windows вместо него используется символ ^, т.е. строка выглядит так:

```
ffmpeg -i "/home/olegchir/temp/h.m4v" ^
```

В целом эта строка означает, что мы будем конвертировать файл, лежащий по пути, заданному после ключа -i.

Далее, нужно указать, как мы будем получать потоки в выходном файле. Нужно три раза взять поток 0:0 и превратить его в 3 видеопотока в разных качествах. Поэтому пишем: -map 0:0 -map 0:0 -map 0:0 - "три раза взять поток 0:0" Далее взять каждый из оставшихся стримов (0:1, 0:2, 0:3, 0:4) по одному разу и просто скопировать. Поэтому пишем: -map 0:1 -map 0:2 -map 0:3 -map 0:4.

Вместе первые две строки выглядят так:

```
ffmpeg -i "/home/olegchir/temp/h.m4v" \
```


-map 0:0 -map 0:0 -map 0:0 -map 0:1 -map 0:2 -map 0:3 -map 0:4 \

Далее нужно объяснить, как все эти дорожки кодируются.

Важное замечание. Если в информации о файле все потоки считались равноправными (видео стоит в том же ряду, что и аудио), то дальше у каждого типа будут свои номера, начиная с 0. То есть, первое видео обозначается как v:0, а второе аудио - a:1.

-c:v:0 libx264 -b:v:0 1800k -metadata:s:v:0 language=eng \ - взять первую видеодорожку, кодировать как x264 с битрейтом как на исходном файле, указать что видео английское.

-c:v:1 libx264 -b:v:1 150k -metadata:s:v:1 language=eng \ - взять первую видеодорожку, кодировать как x264 с битрейтом 150k, указать что видео английское.

-c:v:2 libx264 -b:v:2 100k -metadata:s:v:2 language=eng \ - то же, что предыдущее, но с битрейтом 100k.

Необходимо перекодировать не только дополнительные дорожки, но и исходную, чтобы они были синхронизированы, с одинаковой гор-структурой, что очень важно для адаптивного стриминга.

Когда на второй позиции стоит copy, никакого кодирования не происходит, копируется как есть.

Эти команды скопируют всё аудио:

-c:a:0 copy -metadata:s:a:0 language=rus \

-c:a:1 copy -metadata:s:a:1 language=eng \

-c:s:0 copy -metadata:s:s:0 language=rus \

-c:s:1 copy -metadata:s:s:1 language=eng \

Команда принимает вид:

ffmpeg -i "/home/olegchir/temp/h.m4v" \

-map 0:0 -map 0:0 -map 0:0 -map 0:1 -map 0:2 -map 0:3 -map 0:4 \

-c:v:0 libx264 -b:v:0 1800k -metadata:s:v:0 language=eng \

-c:v:1 libx264 -b:v:1 150k -metadata:s:v:1 language=eng \

-c:v:2 libx264 -b:v:2 100k -metadata:s:v:2 language=eng \

-c:a:0 copy -metadata:s:a:0 language=rus \

-c:a:1 copy -metadata:s:a:1 language=eng \

-c:s:0 copy -metadata:s:s:0 language=rus \

-c:s:1 copy -metadata:s:s:1 language=eng \

Теперь указываем параметры синхронизации и файл, куда будут писаться перекодированный видеофайл:

-async 1 -vsync 1 \

"/home/olegchir/temp/h2.m4v"

Теперь соберем все вместе. Команда будет выглядеть так:

ffmpeg -i "/home/olegchir/temp/h.m4v" \

-map 0:0 -map 0:0 -map 0:0 -map 0:1 -map 0:2 -map 0:3 -map 0:4 \

-c:v:0 libx264 -b:v:0 1800k -metadata:s:v:0 language=eng \

-c:v:1 libx264 -b:v:1 150k -metadata:s:v:1 language=eng \

-c:v:2 libx264 -b:v:2 100k -metadata:s:v:2 language=eng \

-c:a:0 copy -metadata:s:a:0 language=rus \

```
-c:a:1 copy -metadata:s:a:1 language=eng \  
-c:s:0 copy -metadata:s:s:0 language=rus \  
-c:s:1 copy -metadata:s:s:1 language=eng \  
-async 1 -vsync 1 \  
"/home/olegchir/temp/h2.m4v"
```

Кодирование участка видео

Иногда нужно перекодировать не всё видео, а только какой-то его участок. Для этого используется следующий параметр: `-ss 00:00:00 -t 00:05:00`. Здесь первая цифра показывает, с какой секунды должен начинаться кодируемый фрагмент, а вторая цифра - его продолжительность.

Можно использовать этот параметр в сочетании со многими другими. Например:

```
ffmpeg -i "/home/olegchir/temp/h.m4v" \  
-ss 00:00:00 -t 00:05:00 \  
-map 0:0 -map 0:0 -map 0:0 -map 0:1 -map 0:2 -map 0:3 -map 0:4 \  
-c:v:0 libx264 -b:v:0 1800k -metadata:s:v:0 language=eng \  
-c:v:1 libx264 -b:v:1 150k -metadata:s:v:1 language=eng \  
-c:v:2 libx264 -b:v:2 100k -metadata:s:v:2 language=eng \  
-c:a:0 copy -metadata:s:a:0 language=rus \  
-c:a:1 copy -metadata:s:a:1 language=eng \  
-c:s:0 copy -metadata:s:s:0 language=rus \  
-c:s:1 copy -metadata:s:s:1 language=eng \  
-async 1 -vsync 1 \  
"/home/olegchir/temp/h2.m4v"
```

Это кодирование из предыдущей части, но сделанное только для первых 5 секунд фильма.

Смена разрешения для треков с пониженным битрейтом

Иногда нужно не только изменить битрейт, но и уменьшить разрешение видео. Для этого используется следующий параметр: `-filter:v:3 scale=320:240`. Этот параметр нужно добавить к свойству трека так же, как в предыдущих примерах добавлялся битрейт или субтитры.

Разберемся, что здесь написано.

"`-filter`" означает, что дальше будет следовать описание фильтра, "`:v:3`" — это номер видеотрека, который будет присвоен треку в новом разрешении,

"`scale`" — название фильтра (ffmpeg поддерживает разные фильтры, конкретно этот отвечает за смену разрешения),

"`320:240`" — новое разрешение. Важно отметить, что если мы знаем только необходимую ширину, то высоту можно заменить на `-1`, т.е. "`320:-1`". Это позволит автоматически сохранить правильное соотношение сторон.

Теперь посмотрим, как это выглядит на практике. Для этого возьмем код из предыдущих примеров, и добавим четвертый видеотрек ("`-c:v:3`") с шириной 320 ("`scale=320:-1`"). Количество "`-map 0:0`" теперь 4 штуки, что соответствует четырем трекам.

```
ffmpeg -i "/home/olegchir/temp/h.m4v" \  

```

```
-ss 00:00:00 -t 00:05:00 \  
-map 0:0 -map 0:0 -map 0:0 -map 0:0 -map 0:1 -map 0:2 -map 0:3 -map 0:4 \  
-c:v:0 libx264 -b:v:0 1800k -metadata:s:v:0 language=eng \  
-c:v:1 libx264 -b:v:1 150k -metadata:s:v:1 language=eng \  
-c:v:2 libx264 -b:v:2 100k -metadata:s:v:2 language=eng \  
-c:v:3 libx264 -b:v:3 100k -metadata:s:v:3 language=eng -filter:v:3 scale=320:-1 \  
-c:a:0 copy -metadata:s:a:0 language=rus \  
-c:a:1 copy -metadata:s:a:1 language=eng \  
-c:s:0 copy -metadata:s:s:0 language=rus \  
-c:s:1 copy -metadata:s:s:1 language=eng \  
-async 1 -vsync 1 \  
"/home/olegchir/temp/h2.m4v"
```

Плагин Dynafile

Flussonic Media Server умеет транскодировать VOD файлы на лету с помощью плагина dynafile. Этот плагин запускает транскoder при запросе файла. Плагин dynafile поможет вам наложить на видео уникальный для каждого зрителя водяной знак (watermark). Это полезно, например, чтобы определить через кого произошла утечка контента.

Мы рекомендуем использовать плагин с осторожностью, потому что транскодирование "на лету" — дорогая по ресурсам операция с непредсказуемой нагрузкой на процессор. Правильнее было бы заранее подготовить (транскодировать) контент перед раздачей.

Настройка плагина

Плагин включается через конфигурационный файл. Откройте `/etc/flussonic/flussonic.conf` и добавьте:

```
plugin dynafile {  
    cache /tmp/dynacache 2G;  
    transcoder vb=2048k size=1280x720 vb=600k size=360x-1 hw=nvenc;  
}
```

В этом примере мы будем кодировать в два качества и использовать аппаратное ускорение. Обязательно нужно указать директорию для кэширования.

Подробнее про опции транскodера

Важно! Плагин dynafile включает транскодирование сразу для всех VOD локаций, нельзя включить транскодирование только для одной директории.

Для просмотра видео через плагин нужно добавить `/dynafile/` к URL файла:

`http://FLUSSONIC-IP/vod/bunny.mp4/embed.html` — просмотр видео как есть, без транскodера.

`http://FLUSSONIC-IP/dynafile/vod/bunny.mp4/embed.html` — просмотр видео, прошедшего через транскoder.

Наложение водяного знака

Flussonic Media Server может накладывать уникальный watermark ("водяной знак") для каждого зрителя. Водяной знак может содержать ID клиента и IP адрес, такой водяной знак нужно создавать во внешней системе, Flussonic сам такие знаки не создает.

На сервере Flussonic настройте авторизационный бэкэнд, чтобы возвращал PNG изображение в формате base64. Ниже код lua-бэкэнда, который делает запрос за логотипом к внешней системе и передаст его транскoderу:

```
prepare = function(key)  
  
    name = string.gsub(req.name, "vod/", "")  
    qs = {ip = req.ip, token = req.token, name = name, resolution = key}  
    url = "http://example.com/getWatermark?".http.qs_encode(qs)  
    response = http.get(url, 10000)  
  
    if not (response.code == 200) then
```

```

    return false,{code = response.code}
end

png = base64.decode(response.body)
-- path = "/tmp/"..crypto.sha1(key..name..request.token)..".png"
return png
end

extra = {}

string.gsub(req.sizes, "([^\,]+)", function(w) extra[w] = prepare(w) end)

flussonic.log(table.toString(req))
return true,{extra = extra}

Сохраните этот файл как /etc/flussonic/dynafile.lua и настройте Flussonic Media Server:

file vod {
    path /storage/vod;
    cache /tmp/cache 10G;
    auth /etc/flussonic/dynafile.lua;
}
plugin dynafile {
    cache /tmp/dynacache 2G;
    transcoder vb=2048k size=1280x720 vb=600k size=360x-1 hw=nvenc;
}

```

Мультибитрейтный плейлист из файлов

Создание мультибитрейтного содержимого из нескольких файлов

Пусть у вас есть копии фильма в нескольких файлах с разным качеством, и вы не хотите создавать из них один мультибитрейтный файл. Теперь можно проигрывать все эти файлы, используя один HLS плейлист. Клиентский плеер сможет выбирать битрейт точно так же, как в ситуации с одним мультибитрейтным файлом.

Flussonic Media Server может отдавать несколько файлов с разным битрейтом как один мультибитрейтный ресурс. HLS плейлист в этом случае содержит информацию об этих файлах как о разном качестве одного файла.

Вам нужно подготовить файлы и включить автоматическое создание мультибитрейтного ресурса для VOD локации.

Подготовка файлов

Поместите файлы в одну директорию и назовите их так, чтобы имя файла начиналось как имя директории, в которой они находятся. То есть имена файлов должны соответствовать маске DIR_NAME*.mp4, где * — любые допустимые символы. Например:

Имя директории: DIR_NAME, имена файлов: DIR_NAME-1.mp4, DIR_NAMEabc.mp4 и т.д.

См. шаг 2 ниже.

Включение создания мультибитрейтного ресурса

Предположим, что вы уже создали VOD локацию для обращения к файлам, которые будут использоваться для создания мультибитрейтного плейлиста.

В созданную VOD локацию добавьте опцию auto_mbr.

Через файл:

```
file vod {  
  path /storage/movies;  
  auto_mbr;  
}
```

Через UI:

Откройте Files (VOD) > зайдите в VOD локацию > на вкладке Output отметьте Enable MBR from multiple files.

Поместите в директорию файлы, например, такие:

/storage/movies/bunny/bunny.480x360.mp4

/storage/movies/bunny/bunny.720x480.mp4

/storage/movies/bunny/bunny.1080x720.mp4

Размеры видео Flussonic определяет сам, поэтому необязательно указывать их в названии

файла. После слова bunny в имени файла может идти произвольный набор допустимых символов.

Теперь можно запрашивать плейлист по такому URL:

`http://FLUSSONIC-IP:8080/vod/bunny/index.m3u8`

Из примера видно, что список воспроизведения запрашивается на директорию, а не на один файл.

При запросе плейлиста на директорию `/vod/bunny/index.m3u8` Flussonic составляет HLS плейлист из файлов `/vod/bunny/bunny*.mp4`. Плеер "думает", что это один файл.

Замечание. Клиент сможет прочитать содержимое только тех директорий, для которых в конфиге указана опция `auto_mbr`. Иначе Flussonic вернет ошибку 404.

Кластер

Кластер — группа из нескольких серверов, соединенных вместе, чтобы выполнять работу, которая не может быть выполнена одним сервером.

Flussonic Media Server поддерживает несколько режимов для объединения серверов в кластер. Учтите, что нет такого понятия как «просто кластер». Вы всегда должны понимать что именно вы хотите получить, устанавливая несколько серверов: это может быть уменьшение простоя в случае отказа, увеличение общей пропускной способности сервиса или объединение хранилищ между серверами.

Далее описано, как Flussonic Media Server может помочь вам.

Сценарии

Flussonic Media Server может быть настроен для:

- Захвата потоков на одном или нескольких серверах и автоматический рестрим их на другом сервере (немедленный failover).

- Моментальный доступ через рестример к потокам опубликованным на источнике.

- Удаленный доступ к DVR, записанному на сервере источнике.

- Захват и транскодирование потоков группой серверов с автоматическим резервированием.

- Автоматическая и управляемая балансировка клиентов между серверами в группе пиров.

Терминология

Здесь мы объясняем терминологию, которая поможет вам не запутаться:

Кластер

Группа из нескольких серверов с установленным Flussonic Media Server, чтобы работать вместе в одном сервисе.

Источник

Flussonic Media Server, который уже захватывает (или может начать захватывать потоки «по запросу») и может использоваться как источник для сервера рестримера.

Рестример

Flussonic Media Server, который может получать (или уже получает) потоки с одного или нескольких источников.

Рестриминг

Конфигурация группы источников и рестримеров позволяющая автоматически получить на рестримере live-потоки и DVR, записанный на источнике.

Пир

Flussonic Media Server который находится рядом с другим Flussonic Media Server. Только один из них может захватить некоторые потоки, в случае, если двойной захват очень дорог. Например, когда поток — это IP-камера, которая подключена через плохое соединение.

Кластерный захват

Конфигурация группы Flussonic Media Server, гарантирующая что каждый поток будет захвачен лишь единожды. Если один из пиров откажет, другие пиры начнут захватывать его потоки.

Рестриминг

Прочтите статью о кластерном рестриминге.

DVR в кластере

Узнайте про доступ к DVR в кластере из отдельной статьи.

Кластерный захват и транскодирование

Узнайте, как настроить отказоустойчивое транскодирование и надежный захват с помощью кластерного захвата.

Перенаправление на пиры

Flussonic может направлять клиентов на соответствующий пир используя механизм кластерного пиринга.

Ретрансляция потоков

Flussonic Media Server (рестример) может подключиться к другому Flussonic Media Server (источнику), получить список запущенных и доступных по запросу потоков, а затем рестримить их. Вы также получаете прозрачный доступ к архиву на источнике.

Flussonic Media Server позволяет указать несколько серверов-источников и построить отказоустойчивую конфигурацию.

Отличия от HTTP прокси

Многие CDN предлагают решение проблемы доставки видео, используя кластер из обычных HTTP-прокси серверов, которые кешируют сегменты HLS потока и доставляют их до пользователей.

В отличие от простого HTTP прокси, Flussonic Media Server, установленный на все сервера в сети, предоставляет следующие возможности:

- вы получаете не только HLS, но и DASH, HDS, RTMP, RTSP, HTTP MPEG-TS, и UDP MPEG-TS;
- единая авторизация пользователей по всем доступным протоколам;
- централизованная агрегация сессий и сбор статистики.

Главное отличие между обычным HTTP прокси и рестримингом на Flussonic Media Server в том, что вы доставляете видео между серверами лишь единожды, а на выходе получаете весь функционал Flussonic Media Server на рестриминг сервере.

Это нереализуемо на HTTP прокси, потому что он не работает с видео на низком уровне.

Конфигурация

Чтобы включить рестриминг в кластере Flussonic Media Server, вам нужно использовать следующие директивы:

- `source` — чтобы указать сервер, с которого вы хотите рестримить видео.
- `cluster_key` — чтобы Flussonic Media Server мог забрать видео через кластерную авторизацию.

Директива `source` имеет следующий синтаксис и опции:

`cluster_key samekeyforall;`

```
source origin1.tv {  
}
```

```
source origin2.tv {  
    only cbc football;  
}
```

```
source origin3.tv {  
    cluster_key anotherkey;
```

```
except comedy;  
}
```

Вы должны установить одинаковые `cluster_key` на источнике и рестримере. Кластерный ключ очень важен и его нужно хранить в секрете, потому что он может быть использован для настройки удаленного сервера. Он не передается в открытом виде, только хэш.

Директива `source` включает автоматический захват потоков с сервера источника. В ней потоки можно разделить на несколько списков:

- белый список — эти потоки будут статическими (`static`) на рестримере;
- серый список — эти потоки будут доступны на рестримере по запросу (`ondemand`);
- черный список — эти потоки не будут видны на рестримере.

По умолчанию все работающие потоки с источника попадают в белый список, а все потоки по запросу с источника будут в сером списке рестримера.

Опция `except` помещает потоки в черный список. Эта опция имеет более высокий приоритет, чем `only`.

Опция `only` разделяет доступные потоки (кроме тех, что в черном списке) на белый и серый списки: `only` – в белый список, остальные не будут статическими и будут ждать запроса для запуска.

Если на сервере-рестримере будут локальные или публикуемые потоки с тем же именем, что и на сервере источнике, тогда поток с источника будет игнорироваться и будет использоваться только поток, настроенный локально.

Дополнительные настройки

Вы можете применить настройки сразу для всех потоков, запущенных через `source`:

```
source origin1 origin2 {  
    segments 10;  
    auth http://backend/auth.php;  
    dvr /storage 2d 97%;  
}
```

Такая конфигурация автоматически применится для всех потоков, запущенных на рестримере.

Если у вас на основном сервере используется опция `backup`, то вам следует загрузить этот файл на рестример и указать путь:

```
source origin {  
    backup vod/bunny.mp4;  
}
```

Несколько источников

Можно указать несколько источников на рестримере. Если у нескольких источников будут одинаковые имена потоков, такой поток будет настроен с несколькими URL.

Это значит, что если первый источник упадет или потеряет поток, рестример переключит поток на другой источник.

Когда настроено несколько источников с помощью кластерного захвата, вы получаете

действительно высокодоступный кластер.

Протокол M4F

Flussonic Media Server по умолчанию использует для рестриминга свой внутренний протокол M4F.

Этот протокол гарантирует следующие возможности:

- потоки между источником и рестримером хорошо синхронизированы;
- одинаковые таймстампы кадров;
- одинаковое тело;
- нет короткого счетчика таймстапов, как в MPEGTS или RTMP: все таймстампы в UTC;
- сохраняет такую же структуру сегментов, делая byte-to-byte копию с источника для всех протоколов на рестримере;
- сохраняет одинаковыми номера сегментов на источнике и рестримере;
- имеет такую же байт-структуру, как формат хранения архива на диске;
- умеет слать пуш-уведомления клиенту о появлении новых данных;
- предоставляет рестримеру информацию об архиве источника.

Таким образом, при передаче по протоколу M4F точно передаются время и данные. Протокол поддерживает все те кодеки, что и Flussonic.

Наш протокол M4F имеет ряд преимуществ по сравнению с HLS или RTMP:

- RTMP имеет только миллисекундную точность таймстампов, и он ломает таймстампы;
- RTMP имеет лишь 24 (или 32 бита) таймер миллисекунд, MPEG-TS дает 33 бита для таймера, основанного на 90Khz. Это значит, что сложно синхронизировать время между источником и рестримером;
- RTMP и MPEG-TS не имеют способов синхронизации времени потока и реального времени;
- RTSP имеет механизм синхронизации времени потока и реального времени, но имеет проблемы с доставкой В-кадров и некоторых кодеков;
- У M4F достаточно места, чтобы сохранить реальное время в 90khz, давая абсолютное время каждого кадра с высокой точностью.

Кластерный захват потоков

Кластерный захват потоков необходим для решения следующей проблемы: допустим у вас есть некоторое количество серверов (до 20 штук), объединённых в группу, и есть множество потоков, которые надо принять, причём каждый поток только на каком-то одном сервере из группы.

Если один из серверов выходит из строя, надо потоки захватывать на другом сервере автоматически.

Механизм кластерного захвата работает следующим образом: в конфигурационном файле описываются все сервера, участвующие в захвате и прописывается директива `cluster_key`:

```
cluster_key MYSECRET;
```

```
peer s01.myhosting.com;  
peer s02.myhosting.com;  
peer s03.myhosting.com;
```

После чего описываются нужные потоки и им указывается опция `cluster_ingest`:

```
stream cam01 {  
    url ...;  
    cluster_ingest;  
}
```

```
stream cam02 {  
    url ...;  
    cluster_ingest;  
}
```

```
stream cam03 {  
    url ...;  
    cluster_ingest capture_at=s01.myhosting.com;  
}
```

В качестве параметра опции можно указать явную привязку к одному серверу. Это не жесткая привязка, потому что если этот сервер выключится, поток всё равно поднимется на другом.

При достаточно большом количестве потоков они будут равномерно распределены между серверами.

Если сервер отключится, потоки автоматически поднимутся на других серверах. Если он включится, потоки автоматически отключатся на других и вернуться на него.

Если обратиться за потоком к любому серверу из кластера, на котором этого потока нет, то он отдаст код на редирект к другому серверу.

Т.е. фактически можно обратиться к любому серверу из кластера и перенаправят на текущий активный сервер.

В данный момент этой возможностью можно пользоваться для захвата с камер или в ситуации,

когда надо пробросить по узкому каналу много телеканалов и распределить их между ретрансляторами в датацентре.

К этому механизму планируется доделать следующие возможности:

- автоматическая репликация архива с резервного сервера на основной с последующим стиранием с реплики после восстановления

- возможно альтернативную конфигурацию для транскодера в случае аварийного приёма канала с соседнего сервера

Таймауты

Вы можете играть с таймаутами в конфигурации, но вам нужно быть очень осторожными. Установка слишком малых тайм-аутов сделает систему нерабочей.

Помните очень важный факт: в сети невозможно отличить потерю связи и очень долгую задержку.

```
peer s01.myhosting.com {  
    fetch_timeout 1;  
    stale_timeout 3;  
}
```

Это покажет Flussonic Media Server, чтобы получать потоки от пиров раз в 1 секунду. Это **ОЧЕНЬ** часто и так нельзя использовать в работе. Но вы можете играть с ним. `fetch_timeout` несет за это ответственность.

`stale_timeout 3;` — сообщит Flussonic Media Server, что потоки от этого пира были мертвы после 3 секунд отсутствия ответа от этого пира.

Поэтому, если этот пир перегружен и не может ответить через 3 секунды, он считается мертвым, и механизм кластеризации запустит свои потоки на локальном хосте.

Балансировщик нагрузки

Flussonic Media Server (IPTV плагин) может балансировать пользователей между несколькими серверами Flussonic Media Server, используя механизм кластерного пиринга.

Он переадресует пользователя с сервера-балансировщика на другие сервера.

Важно! Балансировщик не может сам стримить потоки, только переадресовывать.

Важно! Балансировщик устанавливается **ИСКЛЮЧИТЕЛЬНО** на отдельный сервер и требует дополнительной лицензии.

Настройка

Включите балансировщик и укажите несколько других Flussonic Media Server пиров:

```
# Global settings
http 8080;
cluster_key somekey;

peer peer1.example.com;
peer peer2.example.com;
peer peer3.example.com;

plugin iptv {
    database sqlite:///opt/flussonic/priv/iptv.db;
    loadbalance bitrate;
}
```

На всех серверах должен быть установлен одинаковый cluster_key.

Пример конфигурации пира:

```
# Global settings
http 8080;
cluster_key somekey;

source origin.example.com {
}
```

У балансировщика есть два режима: bitrate и usage:

В режиме bitrate Flussonic Media Server перенаправляет клиентов на сервер, где наименьший исходящий битрейт.

В режиме usage тоже, но с учетом указанной максимальной пропускной способности ($\%usage = \text{исходящий битрейт} / \text{максимальный} * 100$).

Если пиры имеют одинаковый набор потоков. Flussonic Media Server перенаправит клиента на

наименее загруженный сервер.

Если пиры имеют разный набор потоков, то Flussonic Media Server направит клиентов на соответствующий сервер с учетом наличия потока и загрузки интерфейсов.

Как пользоваться балансировщиком

Просто запросите потоки, так как вы это обычно делаете:

`http://loadbalancer/STREAM/index.m3u8` - for HLS

`http://loadbalancer/STREAM/mpegts` - for HTTP MPEG-TS

`rtmp://loadbalancer/static/STREAM` - for RTMP

Балансировщик знает все ссылки, по которым Flussonic Media Server предоставляет потоки, и сделает переадресацию на пира.

Системные требования

В большинстве случаев будет достаточно виртуальной машины с 1vCPU и 1GB RAM.

Балансировщик не занимается передачей видео, только перенаправлением запросов, поэтому таких ресурсов достаточно даже для нескольких тысяч онлайн клиентов.

Пиринг

Flussonic Media Server может соединиться с другим Flussonic Media Server, получить список работающих и доступных по запросу потоков и перенаправлять клиентов на нужный пир используя свои кластерные возможности.

Настройка

Укажите пир в настройках Flussonic Media Server:

```
# Global settings
http 8080;
cluster_key somekey;

peer peer.example.com;
```

Параметр `cluster_key` должен быть одинаковым на всех серверах в кластере.

Можно задать несколько пиров:

```
# Global settings
http 8080;
cluster_key somekey;

peer peer1.example.com;
peer peer2.example.com;
peer peer3.example.com;
```

Всё пиры могут иметь различные потоки, Flussonic Media Server будет направлять клиентов на нужный пир.

Внутренние и внешние адреса пиров

Когда сервера Flussonic находятся в локальной сети и в настройках вы используете внутренние адреса (имена хостов), которые необходимы для общения пиров между собой внутри локальной сети, эти внутренние адреса могут быть видны на клиентах при проигрывании потока.

Чтобы показывать вовне внешние адреса пиров, но при этом разрешить им использовать внутренние адреса для взаимодействия внутри кластера, добавьте в настройки каждого пира опцию `public`, в которой укажите внешний адрес этого пира:

```
peer streamer1.local {
    public streamer1.example.com;
}
```

При перенаправлении на `streamer1.example.com` будет использоваться тот же протокол, HTTP или HTTPS, который указан в настройках.

Перенаправление

Flussonic Media Server будет перенаправлять клиентов на другой сервер, когда они запрашивают поток.

Между пиром (peer) и источником (source) существует очень важное различие, потому что source спроектирован для копирования видео по выделенному каналу, от источника (origin) на вещающий сервер (edge).

Пиринг спроектирован для случаев, когда клиент может получать видео от любого сервера в группе, а сервера (пиры) в группе общаются друг другом, сообщая, какие потоки они имеют.

Когда клиент подключается по HLS, HTTP MPEG-TS, RTSP, RTMP или открывает embed.html на любой сервер в группе, он может быть перенаправлен на другой сервер, где этот поток действительно находится.

Балансировка загрузки

IPTV плагин может выполнять балансировку нагрузки и перенаправлять запросы на другой сервер.

Плагин может оценить загрузку сервера и других серверов в группе и перенаправить клиента на другой сервер.

Организация CDN

Если одного сервера для раздачи видео перестает хватать, необходимо организовывать сеть доставки контента (CDN).

Flussonic Media Server обладает рядом возможностей, позволяющих упростить эту задачу. В данной статье будем рассматривать небольшую сеть на 3-10 серверов, вещающих прямые эфиры.

Терминология

CDN — Content Delivery Network — сеть доставки контента. Это множество серверов с специализированным ПО, которые ускоряют доставку и отдачу контента конечному пользователю. Сервера расположены по всему миру таким образом, чтобы время ответа посетителям сайта было минимальным. Под «контентом» обычно подразумевают видео и статические элементы веб-сайтов (не требующие выполнения кода на сервере или запросов в базу данных, такие как css/js).

Региональное распределение

Будем рассматривать ситуацию, когда видео захватывается со спутника в России/Европе и потом передается в Европу/Америку для ретрансляции.

Передавать видео придется на большое расстояние по публичному интернету, а следовательно гарантировать качество канала не получится.

Схема организации:

В регионе захвата будет минимум два резервирующих друг друга сервера.

В регионе вещания сервера будут захватывать видео с одного из двух источников.

Каждый канал будет передаваться между регионами только один раз, чтобы не создавать лишний трафик.

Какие-то редко используемые каналы будут передаваться только по запросу от пользователя.

В регионе захвата будет вестись запись что бы видео не потерялось при провалах канала.

В регионе вещания будет так же вестись запись для раздачи архива.

На такой схеме продемонстрируем возможности Flussonic Media Server.

Захват

Можно сделать разные конфигурации захвата потоков в сеть доставки и их конфигурации зависят от того: можно ли брать из источника видео несколько раз или нет.

В самом простом варианте, если у вас видео приходит мультикастом по UDP, можно просто на разных серверах захвата (далее grabber1.cdn.tv и grabber2.cdn.tv) настроить захват одного и того же видео:

http 80;

```
cluster_key mysecretkey;
```

```
stream ort {  
    url udp://239.0.0.1:1234;  
    dvr /storage 3d;  
}
```

Здесь и далее мы подразумеваем, что у серверов хостнеймы правильные и они все резолвятся.

Также важный момент с единым ключом кластера на всех серверах. Здесь мы выбрали mysecretkey, но его можно поменять.

В таком режиме сервера захвата работают абсолютно независимо, архив пишется на обоих серверах и оба сервера постоянно доступны. Но такая схема требует захвата из источника несколько раз, а это не всегда бывает удобно или возможно. Например, если пакет каналов, получаемый по HTTP, укладывается в 500-800 мегабит, то двойной захват может потребовать серьезного расширения входного канала выше гигабита.

Если вы не хотите брать видео из источника несколько раз, то можно настроить кластерный захват.

На серверах захвата добавляется одинаковый конфиг со стримами:

```
http 80;  
cluster_key mysecretkey;  
  
stream ort {  
    url tshttp://origin/ort/mpegts;  
    cluster_ingest capture_at=grabber1.cdn.tv;  
    dvr /storage 3d;  
}
```

С таким конфигом на обоих серверах захвата всё видео будет захватываться на одном сервере, второй будет работать в режиме горячего резерва. Опция capture_at указывает серверам, что grabber1 является более приоритетным для захвата. Если её не указать, то стримы равномерно распределятся между серверами, что тоже может быть неплохой идеей.

Если grabber1.cdn.tv упадет, то grabber2.cdn.tv отреагирует на это и автоматически добавит стримы себе.

При такой настройке второй сервер простаивает, архив на нём не пишется и появится на нем только при падении первого сервера.

Если хочется полностью срезервировать и архив, то нужна другая конфигурация.

Если хочется сохранить одну точку захвата видео, но резервировать архив, тогда надо на втором сервере постоянно забирать потоки и писать их. Для этого придется сделать разный конфиг на разных серверах.

На grabber1.cdn.tv будет такой конфиг:

```
http 80;  
cluster_key mysecretkey;  
  
stream ort {  
    url tshttp://origin/ort/mpegts;  
    dvr /storage 3d;  
}
```

Видео захватывается из источника и пишется на диск.

На grabber2.cdn.tv будет немного другой конфиг:

```
http 80;
```

```
cluster_key mysecretkey;
```

```
stream ort {
```

```
  url hls://grabber1.cdn.tv/ort/mono.m3u8;
```

```
  url tshttp://origin/ort/mpegts;
```

```
  dvr /storage 3d;
```

```
}
```

grabber2 будет стараться забрать видео с первого сервера, но в случае его падения пойдет к источнику напрямую.

Транзит от захвата к стримингу

С точки зрения серверов, находящихся в регионе раздачи, сервера захвата являются источником, который как правило нельзя захватывать больше одного раза, так что можно воспользоваться рецептами касательно раздачи.

Однако нет необходимости прописывать все каналы руками и следить за ними, можно воспользоваться механизмом Flussonic для этого.

На сервере streamer1.cdn.tv, который занимается приёмом видео с захвата достаточно прописать в конфигурационный файл:

```
http 80;
```

```
cluster_key mysecretkey;
```

```
source grabber1.cdn.tv grabber2.cdn.tv {
```

```
  dvr /storage 7d replicate;
```

```
}
```

При такой конфигурации Flussonic Media Server будет забирать каналы с одного или с второго сервера, писать их локально в архив и при необходимости докачивать те данные, которые есть удаленно, но отсутствуют локально.

Если какие-то каналы не требуются для постоянной работы, можно пометить их как каналы по запросу:

```
http 80;
```

```
cluster_key mysecretkey;
```

```
source grabber1.cdn.tv grabber2.cdn.tv {
```

```
  except ort 2x2;
```

```
  dvr /storage 7d replicate;
```

```
}
```

Раздача

При раздаче большого количества видео надо решать вопрос с распределением нагрузки.

Оптимально, когда распределением занимается Middleware, это самая надежная схема с точки

зрения клиентов (не все поддерживают редиректы), но можно воспользоваться и другими вариантами.

Сами стримеры имеет смысл организовывать также, как и транзит. Но только забирать с локальных серверов:

```
http 80;
```

```
cluster_key mysecretkey;
```

```
source streamer1.cdn.tv streamer2.cdn.tv {  
    cache /cache 2d;  
}
```

В этом случае мы включили не DVR, а сегментный кэш. Flussonic Media Server будет складывать сегменты в кэш и при необходимости отдавать их оттуда. Кэш, понятно, размещать на шпиндельных дисках не имеет никакого смысла, это только SSD.

Прямой эфир при этом всё равно обслуживается из памяти и без проблем забивает 10 гигабит, а вот кэш с одного SATA SSD упрется в 6 гигабит SATA шины. Можно это решить сборкой RAID 0 из нескольких SSD.

Важный момент здесь заключается в том, что сегменты, захваченные на граббере дойдут без изменения и с теми же именами до последнего в цепочке стримера, причем сохранятся в таком же виде и для лайва, и для архива. Это поведение существенно отличается от других видеостриминговых серверов.

Запись видеопотоков (Digital Video Recording)

Flussonic Media Server имеет богатые возможности для работы с видеоархивом:

Запись и просмотр видео в реальном времени с камер наблюдения без ограничения на объём архива

Запись и отложенный просмотр телевизионных каналов без ограничения на глубину архива

Поддержка требуемой глубины архива (например не больше 1 недели или не более 100 гигабайт)

Отдача архива по протоколам HLS, HDS, MPEG-TS, RTSP, RTMP, DASH

Отдача архива в режиме таймшифт, т.е. отдача потокового видео со сдвигом на час назад (или на другое время)

Экспорт архива в виде MP4 файлов

Хранение архива в системе хранения данных и облачных системах типа Amazon S3

Запись архива на программный массив дисков с высокой надежностью и доступностью архива.

Преимущества Flussonic для записи видео в архив

DVR в Flussonic Media Server имеет преимущества:

Неограниченный объем архива: можно хранить месяцы и годы записей

Доступ к архиву как к бесконечной ленте. Удобная навигация

Тамбнейлы являются частью архива (быстрый предпросмотр кадров из записи без необходимости перематывать саму запись)

Встроенное зеркалирование и реплицирование архива (при наличии нескольких серверов Flussonic Media Server)

Хорошая скорость записи и воспроизведения, уменьшенное количество обращений к диску при записи архива.

Возможность создания кэша архива на SSD.

Как работать с DVR архивами в Flussonic Media Server

Настройка DVR.

Проигрывание архива с помощью веб-интерфейса или специальных URL.

Доступ к архиву по различным протоколам.

Таймшифт в другой часовой пояс.

Запись передач (Catchup TV).

Экспорт архивных записей в MP4 файл.

Экспорт ключевых кадров в MP4 файл для создания time-lapse видео.

DVR API

Авторизация доступа к архиву.

Скриншоты из видео и сохранение их в архиве.

Кластеризация DVR. Хранения архива в распределенной среде видеодоставки.

Автоматическая репликация архива видеопотока между серверами.

Запись архива в облачные хранилища (S3 или Swift).

Запись архива в программно управляемый RAID.

Настройка DVR (Digital Video Recording)

Flussonic Media Server позволяет записывать потоки в архив и затем работать с архивом. Эта функциональность называется Digital Video Recording (DVR).

Настройки архива свои для каждого потока.

Замечание. Все настройки доступны в конфигурационном файле. Более быстрый способ - через веб-интерфейс (UI), там есть основные настройки.

Запись потока включается именно в настройках архива. Чтобы начать запись, достаточно указать место хранения архива. Но, возможно, вы захотите настроить и другие параметры - например, ограничение на размер архива, включить репликацию или кэширование.

Содержание:

- Где можно хранить архивы
- Настройка DVR в веб-интерфейсе
- Настройка DVR в конфигурационном файле
- Репликация архива
- Кэширование архива на SSD
- Запись архива в WebDAV
- Копирование потоков частями в другое место
- Путь к архиву на диске
- Запрет индексирования

Где можно хранить архив

Flussonic Media Server позволяет вести запись архива:

- Локально на сервер Flussonic Media Server. См. далее на этой странице.
- В массив дисков, которым программно управляет Flussonic. См. Flussonic RAID
- В системе WebDAV.
- В облачном HTTP хранилище (OpenStack Swift или Amazon S3). См. Хранение архива в облаке.

Настройка DVR в UI

После того, как вы добавили поток, можно настроить его запись и хранение.

Чтобы настроить DVR для потока:

- Откройте администраторский веб-интерфейс к Flussonic Media Server и перейдите в Media.
- В списке потоков под Streams перейдите в поток, щелкнув его имя, и перейдите на вкладку DVR.

В настройках DVR для потока укажите:

Path (обязательное поле) – локальная директория, куда будет сохраняться поток. Например, /storage.

На практике, удобно указать одну и ту же директорию для множества потоков. Flussonic создаст для каждого потока свою вложенную директорию, так что архивы будут храниться по-отдельности внутри указанной директории.
См. Путь к архиву на диске.

Важно. Если вы отредактируете настройки и измените путь к DVR, Flussonic не удалит ранее использованный каталог, так что старые записи останутся там. Вы можете удалить их вручную.

Можно по мере записи копировать поток частями в другое место при помощи опции copy. Например:

```
dvr /storage copy=/opt/movies;
```

Saved duration (опционально) – сколько последних часов (hour) или дней (days) хранить. Если указать 30 days, то запись старше 30 дней будет удаляться.

Чтобы изменить единицу измерения, щелкните по ней. Доступные единицы - hours и days.
Allowed absolute disk usage (опционально) - максимальный объём потребляемого места на диске, по умолчанию в Гигабайтах. Например, 10 Gigabytes.

Чтобы изменить единицу измерения, щелкните по ней. Доступные единицы - Kilobytes, Megabytes и Gigabytes.

Дополнительно, можно настроить репликацию архива и кэширование на SSD.

Увидеть, что поток записывается, можно по индикатору в UI - напротив имени потока будет зелёный индикатор:

Внимание! При переименовании потока его видеоархив удаляется.

Настройка DVR через файл

Чтобы начать запись архива, в файл /etc/flussonic/flussonic.conf добавьте параметр dvr для каждого потока, который необходимо записывать.

Пример:

```
stream channel1 {  
  url tshttp://vlc:9090/  
  dvr /storage;  
}
```

```
stream channel2 {  
  url tshttp://vlc:9090/  
  dvr swift://user=office:max&password=secretpass&@prod1.local:8080/movies;  
}
```

При таких настройках, поток channel1 сохраняется в /storage/channel1/, а поток channel2 в облачное хранилище Swift.

Важно. Если вы отредактируете настройки и измените путь к DVR, Flussonic не удалит с диска

ранее использованный каталог, так что старые записи останутся там. Вы можете удалить их вручную.

Размер архива

Размер архива можно ограничить при помощи опций:

2d

Глубина архива в днях (d) или часах (h). Примеры: 2d, 10h. Меньше двух часов указывать не рекомендуется.

90%

Максимальное допустимое заполнение диска в процентах. По умолчанию 97%. Не указывайте 100%, потому что полностью заполненный диск не будет, скорее всего, нормально работать.

Внимание. Мы рекомендуем указывать одинаковое для всех потоков заполнение диска в процентах, например, для каждого 90%. Иначе возможны конфликты и потеря данных.

10G

Максимальное допустимое заполнение диска в килобайтах (K), мегабайтах (M) или гигабайтах (G). Пример: 10G. Можно рассчитать, что один мегабитный поток в день создает 10 гигабайт.

Запись по расписанию

schedule=8:00-16:00

Опция schedule задает расписание для записи архива, в виде интервалов времени.

Указывается UTC время в часах и опционально минутах, интервал может перехлестываться через полночь: 22-1:30. Расписание может содержать несколько интервалов, разделенных запятой: 8:00-16:00,22-1:30.

Пример настройки DVR

```
stream channel0 {  
  url tshttp://vlc:9090/;  
  dvr /storage 6h 90% 10G schedule=8:00-16:00 replicate;  
  cache /mount/ssd1 3d 50G;  
}
```

Репликация архива

Если у вас несколько серверов Flussonic Media Servers, можно включить репликацию DVR с помощью опции Replication.

Важно! Включать репликацию нужно на сервере-рестримере. При этом должны быть указаны сервера-источники в настройках (Media > Origin servers).

По этой теме:

Ретрансляция потоков

Репликация

Чтобы включить репликацию в конфигурационном файле, добавьте опцию replicate:

```
stream channel0 {  
  url tshttp://vlc:9090/;  
  dvr /storage replicate;
```

```
}
```

Чтобы включить репликацию через веб-интерфейс:

Перейдите в Media.

В списке потоков под Streams щелкните имя потока и перейдите на вкладку DVR.

Щелкните enabled под Replication и сохраните настройки (Save).

Кэширование архива на SSD

Кэширование архива на SSD ускоряет работу с ним - когда пользователи перематывают запись назад и вперёд, видео загружается очень быстро. Flussonic можно настроить, чтобы видео кэшировалось на SSD.

Важно! Для кэша DVR не используйте разделы SSD, смонтированные с опцией noatime.

Чтобы включить кэширование на SSD через веб-интерфейс:

В Media, кликните имя потока под Streams, перейдите на вкладку DVR.

Под SSD disk cache укажите:

Path - путь к директории на SSD диске где будет храниться кэшированная запись.

Также можно ограничить размер кэша (в гигабайтах) и время хранения записи в кэше (в днях).

Чтобы включить кэширование на SSD в файле, добавьте параметр cache:

```
stream channel0 {  
  url tshttp://vlc:9090/  
  dvr /storage;  
  cache /mount/ssd1 3d 50G;  
}
```

Запись архива в WebDAV

Вы можете хранить записи DVR в файловой системе WebDAV.

Чтобы настроить хранилище в WebDAV, добавьте в настройки потока строку с адресом WebDAV:

```
stream channel1 {  
  url fake://fake;  
  dvr http://localhost:1234/webdav/storage;  
}
```

Flussonic поддерживает экономичный способ копирования видео на WebDAV по частям с помощью опции copy:

```
stream channel1 {  
  url fake://fake;  
  dvr /storage copy=http://localhost:1234/webdav/storage;  
}
```

Копирование потоков частями в другое место

Flussonic пишет потоки в указанное хранилище посегментно. Можно сделать так, чтобы данные накапливались во временной директории и копировались более крупными порциями в требуемое хранилище - на локальном диске или в облаке.

Как работает копирование

Копирование записанных данных происходит во время записи. Архив записывается в локальную директорию, и затем один раз в час запись копируется в указанное хранилище.

Чтобы копировать архив в облако:

В конфигурационном файле укажите параметр `copy`:

```
stream chan0 {  
    url fake://fake;  
    dvr /storage copy=s3://AWS_ACCESS_ID:AWS_SECRET_KEY@s3.amazonaws.com/mybucket 10G;  
}
```

Поток будет записываться в `/storage` и копироваться частями каждый час в S3.

Чтобы копировать архив в локальное хранилище:

```
stream chan0 {  
    url fake://fake;  
    dvr /storage copy=/opt/movies 10G;  
}
```

Поток будет записываться в `/storage` и копироваться частями каждый час в `/opt/movies`.

Путь к архиву на диске

Для каждого потока Flussonic создает директорию внутри той директории, которая указана в настройках архива как путь на диске. Имя новой директории совпадает с именем потока.

Например, укажем, что поток `my_stream` нужно записывать в `/storage`:

```
stream my_stream {  
    url tshttp://vlc:9090/;  
    dvr /storage;  
}
```

Тогда архив потока будет храниться в `/storage/my_stream`.

Запрет индексирования

Эта настройка актуальна только для сервисов с UGC (User-generated content):

Если на сервере множество потоков, публикуемых пользователями, и нужно удалять старые данные.

Если требуется постоянно хранить архивы всех таких потоков.

Чтобы быстро определить, какие архивы устарели и какие данные в архиве устарели, Flussonic индексирует потоки, публикуемые пользователями, т.е. создает индексный файл с информацией о потоках и глубине хранения архивов. Очистка архивов происходит при запуске Flussonic и может занимать много времени если на сервере очень много потоков. Чтобы

ускорить запуск Flussonic, можно отключить индексирование. Тогда удаление устаревших данных будет происходить только при обращении к архиву.

Если же вам требуется постоянно хранить архивы со всей записью, индексирование не имеет смысла.

Чтобы запретить индексирование DVR, используйте параметр `no_index`:

```
live mylive {  
    url fake://fake;  
    dvr /storage no_index;  
}
```

Если DVR в облаке (S3, Swift), то при конфигурации с запретом индексирования разные серверы Flussonic смогут вести запись в одно хранилище (при миграции потока с одного сервера на другой).

Таймшифт в другой часовой пояс

Многие телеканалы вещаются с расчётом только на один часовой пояс и если мы говорим про Россию, то зачастую это только московский часовой пояс.

Если хочется этот же канал отдавать пользователям в Германии или в США, то возникает неудобство: на часах у людей ещё раннее утро, а в телевизоре уже вечерние передачи.

Flussonic Media Server может отложить проигрывание потока на несколько часов, чтобы у людей в другом часовом поясе передача «Доброе утро» шла добрым утром, а не глубокой ночью.

Есть несколько технических способов организовать это в Flussonic Media Server исходя из частоты обращения к различным каналам в различных часовых поясах. Разница между способами заключается в том, сколько раз будет читаться архив для отложенного показа канала. Можно запустить отложенный поток и тогда архив будет читаться один раз вне зависимости от количества желающих посмотреть, а можно отдать пользователям персональные URL-адреса и тогда архив будет читаться на каждого пользователя.

Если каналов пишется порядка 250 и хочется сделать вещание для 3-х локаций, то суммарно получается 250 каналов на запись и 750 на чтение. Некоторые каналы имеет смысл сделать постоянно запущенными, а некоторые только по запросу пользователей.

Отложенный поток

Пускай у нас есть настроенный канал:

```
stream ort {  
    url udp://239.1.2.3:1234;  
    dvr /storage 1d;  
}
```

Важно, что у него должен быть настроен архив. Теперь можно сделать второй поток:

```
stream ort-1h {  
    url timeshift://ort/3600;  
}
```

Этот поток будет вычитывать из архива и показывать то, что было один час назад (3600 секунд).

Таких потоков можно создавать столько, сколько нужно.

Персональный доступ к архиву

Если есть настроенный поток:

```
stream ort {  
    url udp://239.1.2.3:1234;  
    dvr /storage 1d;  
}
```

то к нему можно выдать URL http://flussonic/ort/timeshift_rel/3600 для проигрывания по HTTP MPEG-TS, http://flussonic/ort/timeshift_rel-3600.m3u8 для проигрывания по HLS и для мультязыковых каналов можно отдать http://flussonic/ort/timeshift_rel_mono-3600.m3u8 для приставок.

В этом случае каждый клиент будет отдельно читать архив. Такой метод стоит использовать для редко используемых сочетаний канала и часового пояса.

Пропуск «дырок» в архиве

В случае если в архиве есть незаписанные участки (например источник был недоступен несколько минут), то при проигрывании таймшифта по HLS Flussonic Media Server будет отдавать пустой плейлист при достижении незаписанного участка.

Если же допустимо нарушить временной сдвиг (таймшифт) и перепрыгнуть через эту «дырку», то следует запрашивать плейлист с параметром `?ignore_gaps=true`:

http://your-flussonic-server-domain/your-stream/timeshift_abs-123123123.m3u8?ignore_gaps=true

Запись передач (Catchup TV)

Мы реализовали и за несколько лет отладили превосходный механизм записи видео в архив в Flussonic Media Server. В этой статье расскажем, как сделать так, что бы воспользоваться всеми возможностями архива в MiddleWare.

Концепция архива

В большинстве MiddleWare запись передач устроена по принципу старого видеомэгнитофона. В нужное время по расписанию включается запись, потом через некоторое время запись прекращается.

Такой подход страдает массой проблем и прежде всего погрешностью в расписании. В начале такой записи идет хвост предыдущей передачи, а финал нужной передачи обрезается. Эту проблему пытаются решать расширением границ записываемой передачи, в итоге на диске образуются нахлестывающиеся записи.

Мы эту задачу в Flussonic Media Server решили по-другому. Flussonic Media Server пишет всё видео в архив в собственном формате и предоставляет доступ к нему, как к бесконечной ленте. Каждый кадр имеет свой адрес — это его реальное время прихода в архив. Когда необходимо посмотреть архив, сообщите с какого времени и по какое надо отдать видео.

Плееры и протоколы к такому решению в целом не привыкли, поэтому Flussonic Media Server умеет подстраиваться под разные варианты использования и предлагать архив в разном виде. Например, можно обратиться к Flussonic Media Server по URL-адресу <http://flussonic-ip/ort/index-1429829884-3600.m3u8> и получить часовой HLS плейлист с момента 1429829884, который будет выглядеть как файл. Таким образом плеер даже не будет догадываться, что это бесконечный архив, а будет просто показывать границы.

Также доступны варианты обращения, когда можно начать играть архив с какого-то момента в режиме потока.

Реализация в MiddleWare

Для того, что бы предоставить человеку доступ к уже записанной передаче, MiddleWare должна сформировать URL к архиву и отдать этот адрес плееру для проигрывания. URL будет иметь вид <http://flussonic-ip/ort/index-1429829884-3600.m3u8>.

Время для URL надо брать из EPG, расписания передач, которое есть в каждой MiddleWare. Важно обратить внимание на то, что Flussonic Media Server требует указания времени в UTC, т.е. по Гринвичу.

Плеер получит URL, решит что это файл и покажет стандартные контролы для проигрывания файла. Передачу можно будет спокойно перематывать, ставить на паузу и продолжать играть дальше.

Очень важный момент с постановкой на паузу: дело в том, что поставить на паузу поток — это очень сложная операция, которая доступна не во всех протоколах. Притормозить проигрывание файла гораздо проще.

Просмотр текущей передачи

С передачей, которая ещё не закончилась, всё немного сложнее. Некоторые плееры, как то iOS, Android или StrobeMediaPlayback умеют работать с т.н. Event-плейлистами. Это такой способ предоставления контента, при котором плеер знает, что сервер сейчас показывает какое-то локальное во времени событие. При этом плеер предоставляет возможность как отмотать к началу, так и вернуться к прямому эфиру.

Для этого надо сформировать URL вида: <http://flussonic-ip/ort/index-1429829884-now.m3u8>.

Будьте аккуратны, если вы запросите URL за сутки, то Flussonic Media Server отдаст огромного размера плейлист. Мы видели, как сбойный клиент забил себе 100 мегабит канала повторяющимся несколько раз в секунду запросом к одному и тому же event плейлисту. При этом Flussonic Media Server отдавал гигантского размера плейлист в несколько сот килобайт без особой нагрузки за счёт хорошо отлаженной реализации архива.

Однако для многих приставок такой URL не подойдет, потому что приставка будет показывать только прямой эфир без возможности отмотки назад. Для таких приставок надо писать JS код, который отлавливает перемотку и переводит клиента на другой URL:

http://flussonic-ip/ort/timeshift_abs-1429829884.m3u8, где 1429829884 — это то время, с которого надо играть передачу.

С timeshift_abs HLS URL-адресами есть большая сложность, связанная с природой HLS протокола. Дело в том, что Flussonic Media Server может лишь вероятностно связывать отдельные HTTP запросы в одну сессию. Flussonic Media Server считает, что сессия та же, если у двух запросов совпадает IP адрес клиента, имя канала, протокол запроса и токен. В случае с несколькими, идущими подряд timeshift abs запросами, Flussonic Media Server решит, что это одна и та же сессии. В итоге может получиться искажение просмотра. Чтобы избежать этого, следует передавать в timeshift_abs запрос новый токен.

Вариант попроще — запросить HTTP MPEGTS http://flussonic-ip/ort/timeshift_abs-1429829884.ts. Однако HTTP MPEGTS вариант лишает доступа к мультибитрейту.

Мультиязык

Традиционно MPEG-TS протокол предлагает стандартизованный механизм выбора языковой дорожки и субтитров при одном видеобитрейте.

Протокол HLS, основанный на MPEG-TS отказывается от упаковки нескольких аудиодорожек в один поток и предлагает раздавать альтернативные аудиодорожки в отдельных сегментах, которые потом надо микшировать на плеере. Так работает плеер в iOS, но так не работает большинство плееров в STB.

Для того, что бы пользователь увидел несколько разных языков при передаче видео по HLS на такие STB, как Mag, Dune, Eltex, надо использовать другой URL: <http://flussonic-ip/ort/video.m3u8>, <http://flussonic-ip/ort/video-1429829884-3600.m3u8>, http://flussonic-ip/ort/timeshift_abs_video-1429829884.m3u8.

При запросе этих URL Flussonic Media Server по-другому упаковывает дорожки и позволяет плеерам, не поддерживающим полностью стандарт HLS, добраться до разных вариантов аудио.

Статус записи

Более продвинутая MiddleWare может уточнять у Flussonic Media Server, записана передача или нет. Для этого надо обратиться по HTTP API:

http://flussonic-ip/ort/recording_status.json?from=1429960179&to=1429963716

Поля from и to означают границы передачи.

В ответ придет JSON:

```
[{  
  "stream": "ort",  
  "ranges": [{ "from": 1429960179, "duration": 3542 }],  
  "motion_log": []  
}]
```

Поле ranges содержит массив из объектов, означающих зоны непрерывной записи. Если есть дырки в записи, в массиве будет более одного объекта. При отсутствии записи за указанное время массив будет пустой.

Проигрывание архива

Доступ к DVR архиву

Архивные записи можно посмотреть с помощью административного веб-интерфейса. Кроме того, к архивным записям можно получить доступ с помощью специальных URL.

Доступ к архиву по URL может осуществляться в режиме потока и в режиме файла. Файл отличается от потока тем, что он конечен. То есть, при просмотре файла плеер будет показывать перемотку, а доступный для просмотра участок видео будет ограничен началом и концом. При просмотре потока перемотка не отображается, у потока нет конца, и он может продолжаться длительное время.

Это отличие видно и по URL. Например, файловый URL может заканчиваться на "index-1345345345354-3600.m3u8" (определены границы: начало 1345345345354 и через 3600 секунд окончание), а потоковый URL может заканчиваться на "timeshift_abs-1345345345354.ts" (определено только начало).

URL зависит от протокола, который вы используете для передачи видео, записанного в архиве. О протоколах читайте в разделе Доступ к DVR по разным протоколам.

Electronic Program Guide

DVR можно использовать вместе с EPG. Современный подход к предоставлению архива телепередач — записывать весь эфир и потом давать просматривать прошедшие (или отматывать назад текущие) передачи, используя расписание телепередач — EPG, Electronic Program Guide, он же электронный телегид.

Подробнее об этом можно прочитать в статье про работу с IPTV middleware.

Информация о записанных передачах и об их времени хранится в Middleware, а Flussonic Media Server предоставляет доступ к своему архиву как к бесконечной ленте (с удобной навигацией).

Есть два режима:

- просмотр уже записанной передачи;
- просмотр передачи, которая сейчас идет.

Если передача уже прошла и закончилась, то middleware на основании EPG формирует ссылку для просмотра из архива. Пользователь получает возможность посмотреть записанную передачу, как обычный файл. Например, если передача началась в 18:15 по Москве (14:15 UTC) 27 августа и длилась час, то middleware должен при выборе передачи в списке прошедших сформировать URL вида: `http://flussonic:8080/ort/index-1409148900-3600.m3u8`

Если передача сейчас всё ещё идет, то middleware может сформировать специальный URL к архиву, позволяющий отматывать назад прямой эфир на начало передачи. Данная функциональность, к сожалению, поддерживается далеко не на всех устройствах и приставках, но тем не менее она существует. URL для такой незакончившейся передачи будет выглядеть так: `http://flussonic:8080/ort/index-1409148900-now.m3u8`

Просмотр архива из административного веб-интерфейса

Вы можете просмотреть содержимое архива видеорегистратора через веб-интерфейс Flussonic Media Server. Для этого нажмите на название видеопотока на вкладке Media, на открывшейся

странице откройте вкладку DVR.

Появится таймлайн записи и содержимое видеопотока в реальном времени в плеере.

Таймлайн

На таймлайне есть несколько зон, отмеченных разными цветами. Красный цвет означает, что записи на этот период нет, зеленый означает, что запись доступна, а голубой цвет означает текущее время.

Навигация

Вы можете кликнуть по любому месту на таймлайне, чтобы начать воспроизведение видеопотока в плеере с этого места. Также вы можете использовать кнопки для навигации по таймлайну:

"-" и "+" для того, чтобы "приблизить" представленный на таймлайне промежуток времени для более точного выбора времени или его "отдалить"

"<" и ">" для того, чтобы передвинуть показанный на таймлайне промежуток времени на более раннее или позднее время.

Экспорт

Вы можете выделить часть архива для экспорта в виде видеофайла с расширением MP4. Для этого переместите указатели границ серого цвета. Рядом с кнопкой MP4 можно видеть время начала и конца выбранного временного промежутка. Сохраните файл, нажав MP4.

Экспорт в mp4

Скачивание с сервера

Фрагмент архива можно выгрузить на компьютер клиента в виде файла, обратившись по адресам:

в виде MP4 <http://flussonic:8080/channel/archive-1350274200-4200.mp4>

MPEG-TS файла <http://flussonic:8080/channel/archive-1350274200-4200.ts>

Где 1350274200 это время начала сегмента в Unix-времени. А 4200 - длительность сегмента в секундах.

С помощью опции `precise=true` можно повысить точность экспорта до секунды:

<http://flussonic:8080/channel/archive-1350274200-60.mp4?precise=true>

Сохранение на сервере

Фрагмент архива можно сохранить на диске сервера в виде MP4, обратившись по адресу, при этом потребуется админская авторизация:

```
curl -u USER:PASSWORD http://flussonic:8080/channel/save-mp4-1350274200-4200?file=recording.mp4
```

Файл будет сохранен в том же каталоге, куда пишется запись канала.

При необходимости можно сохранить файл в другую директорию, указав полный путь к файлу:

```
curl -u USER:PASSWORD http://flussonic:8080/channel/save-mp4-1350274200-4200?file=/path/to/file/recording.mp4
```

Доступ к DVR по различным протоколам

Основные способы доступа к архиву

Доступ к записанному архиву осуществляется на основании времени в UTC. Подобный подход реализован для удобства из-за постоянной смены летнего/зимнего времени и т.п.

На этой странице:

Проигрывание по HLS

Мультиязыковой HLS и моноязыковой HLS

Проигрывание по HDS

Проигрывание по HTTP MPEG-TS

Проигрывание по DASH

DASH манифест для проигрывания неоконченных трансляций из архива

Проигрывание по MSS

Проигрывание по RTMP

Проигрывание по RTSP

Просмотр со сдвигом по времени (таймшифт)

Проигрывание по HLS

HLS может проигрываться на компьютере и приставке, но в особенности HLS нужен для проигрывания видео на мобильных устройствах (iOS, Android).

Для доступа к видео по HLS нужно использовать URL-адрес: <http://flussonic:8080/channel/archive-1350274200-4200.m3u8>

Этот URL означает, что Flussonic Media Server должен отдать 4200 секунд с секунды 1350274200 в виде файла, к которому осуществляется доступ по HLS. Число 1350274200 — это UNIX time, указанное по времени UTC.

Если оригинальный поток имеет более одной аудио или более одной видео дорожки, то по указанному URL отдается variant playlist по стандарту Apple, который позволяет устройствам под управлением iOS (iPhone, iPad) выбирать язык и битрейт. Сегменты в этом случае отдаются с уменьшенным количеством дорожек. Если вы используете какое-то другое устройство (не Apple), то вам пригодятся специальные URL.

Этот режим особенно удобен для демонстрации передач по расписанию: в веб-интерфейсе middleware можно указать такой адрес на основании расписания и не надо тратить место на запись отдельных передач.

В OSMF плеере эта функциональность выглядит так:

Мультиязыковой HLS и моноязыковой HLS

Для приставок, которые не полностью поддерживают HLS, можно указывать специальные video-URL. Разница в том, что по такому адресу отдаются сегменты видео со всеми аудиодорожками. Это нарушение стандарта HLS, но большинство приставок может работать только в таком режиме.

Вот список поддерживаемых URL-адресов (для разных методов доступа):

с фиксированным временем начала и конца: <http://flussonic:8080/channel/video-1350274200-4200.m3u8>

с абсолютным таймшифтом: http://flussonic:8080/channel/video-timeshift_abs-1350274200.m3u8

с возможностью отмотки назад: <http://flussonic:8080/channel/video-1350274200-now.m3u8>

Перемотка и таймшифт описаны в соответствующей части статьи.

Проигрывание по HDS

Необходимо использовать URL: <http://flussonic:8080/ORT/archive-1350274200-4200.f4m>

Ранее использовался следующий URL:

<http://flussonic:8080/channel/archive/1350274200/4200/manifest.f4m>, который некоторое время будет поддерживаться параллельно, но в последующих релизах поддержка URL прежнего вида будет удалена.

Этот URL означает, что Flussonic Media Server должен отдать 4200 секунд с секунды 1350274200 в виде файла, к которому осуществляется доступ по HDS. Число 1350274200 — это UNIX time, указанное по времени UTC.

HDS нужен для использования во flash-плеерах. На устройствах под управлением Apple iOS (iPhone, iPad) проиграть его не получится.

Проигрывание по HTTP MPEG-TS

Фрагмент архива можно получить не на полной скорости, а в потоковом режиме, то есть за время, совпадающее с протяженностью этого фрагмента. По адресу http://flussonic:8080/channel/timeshift_abs-1508403742.ts архив отдается по протоколу MPEG-TS в потоковом режиме.

Проигрывание по DASH

Фрагмент архива можно проиграть по протоколу DASH в виде файла, используя URL следующего вида:

<http://flussonic:8080/channel/archive-1350274200-4200.mpd>

Этот URL означает, что Flussonic Media Server должен отдать диапазон в 4200 секунд с момента времени 1350274200 в Unix timestamp.

DASH манифест для проигрывания неоконченных трансляций из архива

Замечание. Эта информация нужна, если вам требуется использовать статический манифест при проигрывании неоконченных трансляций из архива по DASH.

<http://flussonic:8080/channel/archive-1350274200-4200.mpd>

1350274200 — время начала отрывка для проигрывания из архива.

4200 — сколько секунд проиграть.

Речь идет о потоках, которые идут в прямой трансляции и пишутся в архив. При запросе архива указанная в URL верхняя граница диапазона может уходить в будущее, где трансляции ещё нет.

Flussonic позволяет выбирать, какой DASH манифест (плейлист) будет отсылаться клиенту. Бывает динамический (обновляемый) или статический DASH манифест.

По умолчанию Flussonic обновляет файл манифеста по мере трансляции потока, т.е. манифест в это время динамический. Когда реальное время достигает верхней границы диапазона, то обновлять уже нечего, вся информация о диапазоне получена, и Flussonic поменяет манифест на статический. В некоторых ситуациях лучше использовать сразу статический манифест.

Чтобы указать вид манифеста, используйте параметр `dynamic`:

`dynamic=false`. Будет формироваться статический манифест. В этом случае архив будет проигрываться как файл. Манифест будет содержать информацию о запрошенном диапазоне из архива и не будет обновляться.

`http://flussonic:8080/channel/archive-1350274200-4200.mpd?dynamic=false`

`dynamic=auto`. Это поведение по умолчанию, и этот параметр можно не указывать. Сначала будет формироваться динамический манифест (пока идет прямая трансляция), который поменяется с динамического на статический в момент, когда трансляция в реальном времени достигнет конца запрошенного диапазона.

`http://flussonic:8080/channel/archive-1350274200-4200.mpd?dynamic=auto`

Проигрывание по MSS

Список поддерживаемых URL:

Фрагмент архива можно проиграть по протоколу MSS в виде файла, используя URL следующего вида:

`http://flussonic:8080/channel/archive=1568988189-120).ism/manifest`

с абсолютным таймшифтом:

`http://flussonic:8080/channel(timeshift_abs=1568988189).ism/manifest`

таймшифт с относительным временем:

`http://flussonic:8080/channel(timeshift_rel=600).ism/manifest`

Плейлист с большим «скользящим» окном, позволяющий перематывать и ставить на паузу MSS потока:

`http://flussonic:8080/channel(rewind=7200).ism/manifest`

Проигрывание по RTMP

Flussonic Media Server может проигрывать архив по протоколу RTMP. Для этого надо указать следующие аргументы:

```
var flashvars = {
    streamer:'rtmp://flussonic/rtmp',
    file: 'ort?from=1398267588&to=1398268588',
    'rtmp.tunneling':false,
    autostart: true
};
swfobject.embedSWF('/flu/jwplayer.swf',element,'640','480','10.3','false', flashvars,
```

```
{allowfullscreen:'true',allowscriptaccess:'always'},  
{id:'jwplayer',name:'jwplayer'}  
);
```

Указывается имя потока, но добавляются query string с обязательным параметром from и опциональным параметром to.

Также можно указать параметр speed=2, speed=4 или speed=8 и тогда Flussonic Media Server будет проигрывать архив без звука в ускоренном режиме.

Проигрывание по RTSP

Flussonic Media Server может проигрывать архив по протоколу RTSP. Для этого надо запросить следующий URL: `rtsp://flussonic:ort?from=1398267588&to=1398268588`

Указывается имя потока, но добавляется query string с обязательным параметром from и опциональным параметром to.

Также можно указать параметр speed=2, speed=4 или speed=8 и тогда Flussonic Media Server будет проигрывать архив без звука в ускоренном режиме.

Просмотр со сдвигом по времени (таймшифт)

Таймшифт с относительным временем

К архиву можно обратиться как к обычному источнику, но со смещением во времени. Для каждого протокола существует свой URL:

MPEG-TS: `http://flussonic:8080/channel/timeshift_rel/3600`

HLS: `http://flussonic:8080/channel/rel-timeshift_rel-3600.m3u8`

mono HLS: `http://flussonic:8080/channel/mono-timeshift_rel-3600.m3u8`

Важно отметить, что лучше использовать отдельный тип источника timeshift, описанный ниже.

Таймшифт с абсолютным временем

По URL `http://flussonic:8080/channel/timeshift_abs-1350274200.ts` можно получить MPEG-TS поток с указанной секунды. Например, это может быть удобно для старых приставок и для демонстрации записанных передач по расписанию.

Трансляция события с отмоткой назад

Эта возможность работает с HDS, HLS, DASH, позволяет получить прямой эфир события с возможностью отмотать назад до указанного момента в секундах.

Для HDS доступ осуществляется по адресу URL: `http://flussonic:8080/ORT/archive-1350274200-now.f4m`

(ранее использовался URL: `http://flussonic:8080/channel/archive/1350274200/now/manifest.f4m`).

Для HLS существуют два URL:

Первый: `http://flussonic:8080/channel/archive/1350274200/now/index.m3u8` работает только с продуктами Apple.

Второй: `http://flussonic:8080/channel/index-1350274200-now.m3u8` будет работать везде.

Для DASH: `http://flussonic:8080/channel/archive-From-now.mpd`

Здесь 1350274200 — это UNIX time в секундах по времени UTC.

В OSMF плеере эта функциональность выглядит так:

Запуск таймшифт-потока с постоянной задержкой

Чтобы запустить дополнительный поток, который отстает от реального времени на постоянное время, следует воспользоваться дополнительным типом источников:

```
stream channel {  
  url tshttp://vlc:9090;  
  dvr /storage 10080;  
}  
stream channel-1hour {  
  url timeshift://channel/3600;  
}
```

В системе появится новый поток, который отстает от channel ровно на один час. Если в записи будут перерывы, отставание не поменяется.

Повторные запросы к timeshift

Часто возникает вопрос: при повторных обращениях к одному и тому же HLS URL с использованием timeshift_abs, в плеере проигрывается разное видео. Почему так, разве абсолютный URL не должен указывать на вполне конкретное время — то, которое передали ему параметром?

Дело в том, что, когда запрашивается HLS URL для какого-то потока, Flussonic Media Server открывает для него новую сессию. Если timeshift URL используется много раз, любой новый запрос продолжает использовать эту же самую сессию. Иначе говоря, все запросы к видео выполняются относительно текущей сессии. Так что, если используется одно и то же значение времени в параметре timeshift_abs, на самом деле это не совершенно «абсолютное» время, ведь оно всё еще рассчитывается относительно текущей сессии. Поэтому каждый раз, когда запрашивается одно и то же время, оно в конечном счёте пересчитывается в разное время, и вы видите разное видео. Это нормальное поведение, а не баг и не ошибка пользователя. Это единственный хороший способ реализовать HLS timeshift.

Это поведение можно обойти вручную, чтобы каждый все-таки получать одно и то же видео. Для этого у каждого нового запроса нужно менять token. Изменение token заставляет Flussonic Media Server создавать новую сессию.

Например: http://flussonic:8080/mystream/timeshift_abs-1430227800.m3u8?token=123
http://flussonic:8080/mystream/timeshift_abs-1430227800.m3u8?token=124
http://flussonic:8080/mystream/timeshift_abs-1430227800.m3u8?token=125 И так далее.

Timelapse

Экспорт ключевых кадров в MP4 файл

Flussonic Media Server позволяет выгрузить только ключевые кадры в виде MP4 файла. Это может быть полезно для создания timelapse-видео.

Такой файл можно выгрузить на компьютер клиента, обратившись по адресам:

<http://FLUSSONIC-IP:8080/channel/archive-1350274200-4200.mp4?timelapse> — запрос файла из ключевых кадров на скорости 25 fps.

<http://FLUSSONIC-IP:8080/channel/archive-1350274200-4200.mp4?timelapse=20> — запрос файла с коррекцией fps, длительность файла будет 20 секунд.

DVR API

Обзор API для работы с DVR

Flussonic предоставляет HTTP API для работы с DVR, который позволяет получать данные о потоке и настраивать запись архива. Часть команд доступна только администратору, а другая часть также и пользователям (с защитой с помощью токенов).

Администратор может изменить настройки архива или сохранить его в виде файла на диск. Пользователи могут запрашивать информацию о потоке и т.п.

Ниже список команд HTTP API.

Действия, доступные только администратору

Настроить DVR для потока

Остановить запись в архив

Начать запись в архив

Заблокировать запись в интервале времени интервал с целью защиты от удаления

Разблокировать запись в интервале времени, чтобы ее можно было удалить

Сохранить архив в MP4 на диск на сервере

Информация, которую могут запросить пользователи

Общее время записи

Информация о записанных периодах

Запрос JPEG скриншотов

Генерация JPEG скриншотов по запросу

Получение видео-скриншотов

Экспорт архива в MP4, MPEG-TS файлы

Настройка DVR для потока

Для настройки DVR необходимо указать логин и пароль администратора и передать текстовую строку с настройками, которая будет записана в конфигурационный файл Flussonic:

```
curl -u flussonic:pass --  
data '{"streams":{"ort":{"dvr":{"dvr_limit":7200,"root":"/storage"}}}}' http://192.168.2.3:8080/flussonic/api/  
modify_config
```

Создать и настроить DVR архив можно также и через SQL API.

Чтобы изменить настройки, используйте такую же команду, как и при их создании:

```
curl -u flussonic:pass --  
data '{"streams":{"ort":{"dvr":{"dvr_limit":172800,"dvr_offline":true,"root":"/storage"}}}}' http://192.168.2.  
3:8080/flussonic/api/modify_config
```

Остановка записи в архив

```
curl -u flussonic:pass --data "" http://192.168.2.3:8080/flussonic/api/dvr_disable/ort
```

Эта команда остановит DVR на транслируемом потоке ort. Но если поток будет перезапущен, то запись архива возобновится.

Начало записи в архив

```
curl -u flussonic:pass --data "" http://192.168.2.3:8080/flussonic/api/dvr_enable/ort
```

По этой команде начнется запись в архив работающего потока ort. Если поток был настроен как dvr_offline или ранее была выполнена команда dvr_disable, этот вызов начнет запись такого потока в архив. Но при перезапуске потока его статус снова станет такой, как указано в настройке DVR.

DVR Lock

Вы можете заблокировать запись, чтобы защитить ее от автоматического удаления из архива. Это может быть полезно для организации услуги nPVR (Network Personal Video Recorder) или просто для сохранения важных данных.

```
curl -u flussonic:letmein! --  
data '{"stream":"ort","from":1483971680,"duration":1000}' http://192.168.2.3:8080/flussonic/api/dvr/lock
```

Где:

1483971680 — время начала в Unix timestamp;

1000 — продолжительность в секундах.

Замечание. Если для архива настроена опция сору для копирования записи, например, на Amazon S3, то /api/dvr/lock не работает.

Вы можете получить список защищенных записей из API по ссылке:

```
curl http://192.168.2.3:8080/ort/recording_status.json?from=1483970680&to=now&request=ranges,locks  
[{"stream":"ort","ranges":[{"duration":3687,"from":1483970675}, {"duration":56758,"from":1483974376}, {"  
duration":332,"from":1484031143}], "locks":[{"duration":1004,"from":1483971680}]]
```

DVR Unlock

Если запись разблокировать, то она будет автоматически удалена в соответствии с настройками очистки архива.

```
curl -u flussonic:letmein! --  
data '{"stream":"ort","from":1483971680,"duration":1000}' http://192.168.2.3:8080/flussonic/api/dvr/unlock
```

Где:

1483971680 — время начала в Unix timestamp;

1000 — продолжительность в секундах.

Сохранение архива в MP4

Администратор может экспортировать части архива в MP4 файлы и сохранять их на локальном диске на сервере, не передавая по сети.

```
curl -u flussonic:letmein! --data " http://192.168.2.3:8080/ort/save-mp4-1350274200-4200?file=/storage/recording1.mp4"
```

Где:

1483971680 — время начала в Unix timestamp;

1000 — продолжительность в секундах.

Указанная часть архива будет сохранена в файл /storage/recording1.mp4

Будьте осторожны при указании имени файла, чтобы не затереть существующие файлы.

Можно сохранять и метаданные в MP4 файл:

```
curl -u flussonic:letmein! --data 'some opaque value' http://192.168.2.3:8080/ort/save-mp4-1350274200-4200?file=/storage/recording1.mp4&meta=true
```

Метаданные будут записаны в атом udta.meta.ilst.data.

Общее время записи

Flussonic может дать ответ на вопрос, сколько всего было интервалов с наличием записи, где начало и окончание записи в каждом из них:

```
$ curl http://192.168.2.3:8080/ort/recording_status.json
{"ort":{"from":1525186456,"to":1526910900}}
```

Время Flussonic отдает в UTC timestamp.

Отчет о периодах записи

Flussonic можно спросить о том, какие были периоды с наличием записи, начиная с указанного времени:

```
curl -v 'http://192.168.2.3:8080/ort/recording_status.json?from=1525186456'
```

...

< HTTP/1.1 200 OK

...

< Server: Flussonic

< X-Route-Time: 83

< X-Run-Time: 3391

< Content-Type: application/json

```
[{"stream":"ort","warning":"too_big_range","ranges":[{"duration":28800,"from":1525183200},{"duration":7200,"from":1525219200},{"duration":3600,"from":1525233600},{"duration":10800,"from":1525240800},{"duration":43200,"from":1525255200},{"duration":7200,"from":1525309200},{"duration":39600,"from":1525323600},{"duration":7200,"from":1525377600},{"duration":3600,"from":1525392000},{"duration":3600,"from":1525402800}]
```

Обычно, когда вы запрашиваете данные за несколько дней, вам не интересны точные данные о периодах, потому что их всё равно будет сложно использовать. Поэтому Flussonic выдал сокращенный ответ и предупреждение "warning":"too_big_range" о том, что интервал с записью

очень длинный - более 4 дней. В сокращенном ответе Flussonic отдает неточные данные об интервалах, получение которых не требует больших затрат.

Если вам необходимы подробные данные за большой период времени, добавьте опцию `force_detailed`.

```
curl -v 'http://192.168.2.3:8080/ort/recording_status.json?from=1525186456&request=ranges,force_detailed'
...
< HTTP/1.1 200 OK
...
< Server: Flussonic
< X-Route-Time: 83
< X-Run-Time: 174210
< Content-Type: application/json

[{"stream":"ort","ranges":[{"duration":24179,"from":1525186456}, {"duration":4788,"from":1525221374}, {"duration":0,"from":1525236971}, {"duration":131,"from":1525244207}, {"duration":2018,"from":1525244378}, {"duration":656,"from":1525246742}, {"duration":0,"from":1525249845}, {"duration":1169,"from":1525249848}, {"duration":27216,"from":1525257917}, {"duration":10177,"from":1525285203}, {"duration":0,"from":1525296286}, {"duration":151,"from":1525311266}, {"duration":314,"from":1525311872}, {"duration":2359,"from":1525312194}, {"duration":7,"from":1525325359}, {"duration":0,"from":1525328402}, {"duration":533,"from":1525328405}, {"duration":
...

```

Предупреждения о большом интервале больше нет, но запрос выполнялся в 50 раз дольше.

Почему могут быть не нужны точные данные за долгий период:

При отрисовке timeline на экране шириной 1920 пикселей, если хотим поместить одну неделю записи на всю ширину, то 1 пиксель будет соответствовать 5 минутам. На час записи придется 10-11 пикселей. Поэтому совсем необязательно делать дорогостоящий запрос к серверу, хватит и точности до часа-получаса.

Запросим более короткий период времени (меньше, чем 4 дня):

```
curl -
v 'http://192.168.2.3:8080/ort/recording_status.json?from=1525186456&to=1525272856&request=ranges'
...
< HTTP/1.1 200 OK
...
< Server: Flussonic
< X-Route-Time: 83
< X-Run-Time: 20723
< Content-Type: application/json

[{"stream":"ort","ranges":[{"duration":24179,"from":1525186456}, {"duration":4788,"from":1525221374}, {"duration":0,"from":1525236971}, {"duration":131,"from":1525244207}, {"duration":2018,"from":1525244378}, {"duration":656,"from":1525246742}, {"duration":0,"from":1525249845}, {"duration":1169,"from":1525249848}, {"duration":14941,"from":1525257917}]]}

```

Такой запрос выполняется дольше, чем при кратком ответе, но быстрее, чем при запросе точных данных.

Значения параметра request

Перечислим значения, которые принимает request:

ranges — список непрерывно записанных периодов в DVR архиве. Этот список может меняться, если в текущий момент Flussonic производит репликацию архива с сервера-источника.

brief_thumbnails — список скриншотов, сохраненных в архиве, в виде UTC времени каждого. Мы не рекомендуем получать JPEG-скриншоты по UTC, потому что мы сделали доступ по приблизительному GMT - об этом ниже на этой странице.

force_detailed — принудительно формирует ответ с точными данными о периодах, даже если запрошен долгий период времени более 4 дней.

hour_bitmap — формирует ответ в виде строки, содержащей карту записи по часам. Это самый компактный способ представления данных из архива.

Не работает вместе с ranges.

locks — запрашивает список заблокированных записей в архиве.

motion_log — запрашивает данные о событиях, полученных от датчика движения.

Значения можно перечислить через запятую, например, request=ranges,hour_bitmap.

Данные от датчиков движения

При приеме событий о движении с камеры Flussonic сохраняет время начала и окончания событий движения.

Чтобы запросить, когда движение началось и сколько продолжалось, добавьте к recording_status.json параметр request=motion_log. В параметрах from и to задайте границы интересующего вас периода времени, в UTC.

```
curl http://192.168.2.3:8080/cam_office/recording_status.json?from=1528144663&to=1528148263&request=motion_log
```

```
[{"stream":"cam10","motion_log":[{"data":{"duration":42,"start":1528145203},"subtype":"range","type":"motion"}]}
```

Получение JPEG скриншотов по времени GMT

Когда вы включили скриншоты для DVR или по URL, Flussonic начинает создавать и писать скриншоты на диск, а вы можете получать эти скриншоты по специальным URL. Эти URL содержат GMT время того момента, в который был получен скриншот.

Flussonic позволяет находить JPEG-скриншоты по приблизительному времени. Например, вы можете знать из датчика движения, что в районе определенного времени записаны события или вы можете знать необходимое время посмотрев его в плеере.

Узнавать точное время скриншотов не экономично, т.к. нужно получать список скриншотов с UTC временем и конвертировать в GMT. Flussonic приходит на помощь - он умеет находить ближайшее GMT время к указанному вами, когда был снят скриншот.

В DVR HTTP API, Flussonic исправляет время на точное, если вы указали не очень точное в своем запросе. Он возвратит часть URL для получения реального скриншота.

Например, запросим скриншот по такому времени: 2018/05/02/06/59/38. Именно в этот момент скриншота нет, но есть рядом. Flussonic возвращает Location с правильным временем 2018/05/02/07/00/40, которое мы используем далее для доступа к JPEG файлу:

```
curl -v 'http://192.168.2.3:8080/ort/2018/05/02/06/59/38.jpg'
```

...

```
< HTTP/1.1 302 Found
< Location: /ort/2018/05/02/07/00/40.jpg

curl -v 'http://192.168.2.3:8080/ort/2018/05/02/07/00/40.jpg'
...
< HTTP/1.1 200 OK
...
< Content-Length: 5738
< Content-Type: image/jpeg
< Last-Modified: Wed, 02-May-2018 07:00:40 GMT
< X-Thumbnail-Utc: 1525244440
...далее идёт jpeg...
```

Генерация JPEG скриншотов по запросу

Flussonic может создавать JPEG-скриншоты на лету. При этом экономятся ресурсы диска. Нагрузка на процессор будет ощутимая, поэтому защищайте сервер с помощью авторизации доступа. А лучше используйте видео-скриншоты, если нет необходимости именно в JPEG.

```
curl -v 'http://192.168.2.3:8080/ort/2018/05/02/07/00/40-preview.jpg'
...
< HTTP/1.1 200 OK
...
< Content-Length: 5738
< Content-Type: image/jpeg
< Last-Modified: Wed, 02-May-2018 07:00:40 GMT
< X-Thumbnail-Utc: 1525244440
...here goes jpeg...
```

Получение MP4 видео-скриншотов

Мы рекомендуем использовать видео-скриншоты вместо JPEG-скриншотов. Видео-скриншоты архива получают почти так же, как JPEG скриншоты. Flussonic исправляет URL, если по запрошенному URL нет подходящего кадра для создания скриншота.

```
curl -v 'http://192.168.2.3:8080/ort/2018/05/02/06/59/38-preview.mp4'
...
< HTTP/1.1 302 Found
< Location: /ort/2018/05/02/07/00/40-preview.mp4

curl -v 'http://192.168.2.3:8080/ort/2018/05/02/07/00/40-preview.mp4'
...
< HTTP/1.1 200 OK
...
< Content-Length: 8756
< Content-Type: vide/mp4
< Last-Modified: Wed, 02-May-2018 07:00:40 GMT
< X-Thumbnail-Utc: 1525244440
...here goes jpeg...
```

Flussonic вернет MP4, состоящий из одного кадра.

Экспорт части архива в MP4

Пример запроса к Flussonic на получение части архива длиной в 1 час в MP4 файле по сети:

```
curl -v http://192.168.2.3:8080/ort/archive-1525186456-3600.mp4
```

```
...
```

```
< HTTP/1.1 200 OK
```

```
< Content-Type: video/mp4
```

```
< X-Session: a666873ba918d02f81f9b39e336906d85bdbac20
```

```
< Content-Disposition: attachment; filename=ort_1525186456_3600.mp4
```

```
..
```

```
< X-Prepare-Percent: 70
```

```
< X-Prepare-Percent: 80
```

```
< Content-Length: 81773187
```

```
< X-Prepare-Time: 27487
```

```
...here goes big mp4...
```

Кластеризация DVR

Задача хранения архива в распределенной среде видеодоставки создает несколько проблем:

необходимо обеспечить сохранность и доступность архива. RAID-системы не решают проблему, потому что при отключении сервера все данные становятся недоступны;

для популярного и востребованного контента необходимо обеспечить доставку архива ближе к пользователям и снижение нагрузки на сервера источника;

в условиях геораспределенной доставки видео надо уметь восстанавливать целостность архива на вторичных серверах после потери с сервером источником.

Самый простой вариант зеркалирования DVR — включить DVR на источнике и на вторичных серверах.

Кластеризация DVR в Flussonic включается очень просто: достаточно забирать через директиву source:

```
source origin1 {  
    dvr /storage 2d;  
}
```

При этом локальный сервер даже с выключенным DVR начнет отвечать на DVR запросы (не по всем протоколам), забирая недостающее видео с источника.

Важной особенностью Flussonic является возможность использования отдельного сегментного кэша на SSD для снятия нагрузки. Обычно на последние сутки видео приходится до 90% всех просмотров, поэтому при вещании масштабных событий можно использовать SSD для снятия нагрузки с HDD:

```
stream ort {  
    url udp://239.0.0.1:5000;  
    dvr /storage 80d;  
    cache /cache 2d 800G;  
}
```

При использовании кластеризации DVR можно не указывать сторэдж на вторичных серверах, а указать только сегментный кэш:

```
source origin1 {  
    cache /cache 2d 400G;  
}
```

При такой конфигурации вторичный сервер будет использовать сегментный кэш для хранения DVR, но фактически управлять всем архивом будет сервер источник. В случае потери связи с источником вторичный сервер уже не сможет отвечать на запросы к архиву.

Репликация

Догоняющая репликация DVR

Flussonic Media Server имеет уникальную возможность автоматической репликации архива видеопотока между серверами. После установления соединения между источником и вторичным сервером, вторичный сервер сам заберет недостающее видео с источника.

При этом вторичный сервер умеет ограничивать суммарную скорость репликации что бы не прерывать качество вещания прямого эфира. Эта возможность может быть использована в нескольких случаях:

- копирование архива на другие сервера для надежности с автовосстановлением после сбоев;
- вещание со сдвигом времени в другом часовом поясе с надежным автоматическим восстановлением недостающего видео.

Чтобы включить репликацию, необходимо использовать ключевое слово `replicate` в конфигурации `dvr`:

```
source origin1 {  
    dvr /storage 20d replicate;  
}
```

При наличии ключевого слова `replicate` запись будет включена постоянно, поэтому использовать опцию `dvr_offline` вместо `dvr` не рекомендуется.

Репликацию можно включить и для отдельного потока:

```
stream ort {  
    url m4f://flussonic1.myhosting.com/ort;  
    dvr /storage 7d replicate ;  
}
```

В этом случае для получения потока мы рекомендуем использовать внутренний протокол Flussonic `m4f`. О преимуществах протокола `m4f` можно прочитать по ссылке.

DVR в облаке

Хранение архива в облаке

Flussonic Media Server может писать видеоархив в HTTP хранилище, например, Amazon S3 или OpenStack Storage (Swift).

Flussonic пишет поток посегментно сразу в облачное хранилище, что может стоить дорого. Чтобы записывать более крупными частями, используйте параметр `soru` (см. ниже).

Для записи на Amazon S3 необходимо сконфигурировать поток следующим образом:

```
stream chan0 {
  dvr s3://AWS_ACCESS_ID:AWS_SECRET_KEY@s3.amazonaws.com/mybucket 10G;
}
```

Для записи на OpenStack Storage (Swift) сконфигурируйте поток следующим образом:

```
stream chan0 {
  dvr swift://user=video:streamer&password=SECRETPASS&@swift-storage.local/movies 10G;
}
```

Для записи на Akamai storage сконфигурируйте поток следующим образом:

```
stream chan0 {
  dvr akamai://keyName:keyValue@akamaihd.net/cpCode/dvr 10G;
}
```

Копирование архива в облако

Параметр `soru` позволяет значительно снизить количество обращений к диску при записи в облачное хранилище.

При использовании копирования Flussonic сначала записывает поток на локальный диск (в указанную директорию). Затем, каждый час, он копирует записанные данные в хранилище.

Указывать параметр `soru` нужно так:

```
stream chan0 {
  dvr /storage copy=s3://AWS_ACCESS_ID:AWS_SECRET_KEY@s3.amazonaws.com/mybucket 10G;
}
```

Запись в сетевое хранилище при миграции потока

Группа серверов Flussonic может работать с одним сетевым хранилищем, при этом запись ведется в один каталог. При переносе потока с одного сервера на другой новый сервер будет подхватывать запись, сделанную старым.

Проще всего перенести поток с помощью SQL API, это делается одной строкой:

```
mysql> update streams set server='srv03.cdn.local' where name='bunny2';
Query OK, 1 row affected (0.02 sec)
```

Flussonic полностью перенесет конфигурацию потока на новый сервер, а архив продолжит

работу автоматически. Подробнее про SQL API.

Важно! Несколько серверов не должны писать один и тот же поток одновременно.

Читайте также

[Запись видеопотоков](#)

[Настройки архива](#)

Flussonic RAID для DVR

Flussonic RAID для DVR — это программный RAID-массив, который обеспечивает высокую надежность и удобство при записи видеоданных на десятки дисков. Flussonic RAID имеет существенные преимущества перед похожими решениями:

Нет необходимости покупать дорогой аппаратный RAID-контроллер, например, на 60 дисков. Все диски используются в режиме JBOD (Just a Bunch of Disks). Каждый диск нужно отформатировать отдельно и смонтировать в систему в определенный каталог. После этого вы настраиваете Flussonic, и он начинает следить за их состоянием и самостоятельно распределять данные по ним.

Надежность: При сбое в работе какого-либо диска запись продолжится на другие диски в массиве. Может пострадать только часть данных, записанная на отказавший диск.

Непрерывная работа: Можно добавлять диски или удалять их из RAID прямо во время записи архива, и не требуется перезапуск Flussonic — изменения применятся сразу.

Автоматическая "бесшовная" миграция данных между дисками в RAID, что делает возможным, например, удалить все данные с какого-либо диска без прерывания работы с этим DVR архивом (записи или проигрывания).

Автоматическое распределение записываемых данных между дисками в RAID. Flussonic сам решает, на какой диск оптимальнее будет записать данные. Поток данных может быть больше, чем можно успеть записать на один диск за приемлемое время — поэтому Flussonic равномерно распределяет запись между дисками. А чтобы снизить затраты электроэнергии, можно ограничить число дисков, на которые одновременно может производиться запись.

Защита от записи в случае, если произошла ошибка с монтированием дисков. Такая защита предотвратит запись всех данных в корневой раздел.

Настройка программного массива дисков для Flussonic Media Server

Существующие архивы перенести в RAID нельзя, можно только начать запись заново. Чтобы начать работу с RAID, нужно сконфигурировать сервер, добавив настройки дискового массива (это по сути глобальные настройки DVR) и в потоках указать этот массив для записи архива.

Порядок настройки различается для Flussonic Watcher и для Flussonic Media Server.

Массив дисков для записи DVR создается на уровне операционной системы, когда вы монтируете диски, а затем весь массив управляется программно Flussonic-сервером.

Настройки массива — это глобальные настройки DVR. Flussonic позволяет задать эти настройки в конфигурационном файле `/etc/flussonic/flussonic.conf`.

Сначала зададим глобальные настройки массива, например:

```
dvr my_raid {  
    root /dvr/raid;  
    raid 0;  
    metadata idx;  
  
    disk d1;  
    disk d2 keep;
```



```
disk d3 migrate;  
}
```

Затем в настройках потока укажем, что архив следует записывать в массив `my_raid` с глубиной семь дней:

```
stream channel5 {  
    url tshttp://10.0.4.5:9000/streamname;  
    dvr @my_raid 7d;  
}
```

Поток примет глобальные настройки, указанные в настройках массива `my_raid`. Некоторые из них можно переопределить в настройках потока.

Пример с полным набором настроек

Дисковый массив имеет три вида настроек:

Глобальные настройки DVR массива

Настройки, которые можно переопределить в настройках отдельного потока

Параметры, управляющие процессом записи на диски.

```
dvr my_raid {  
    root /dvr/raid;    # базовая директория  
    raid 0;            # включить RAID  
    active 2;          # вести запись на 2 диска  
    metadata idx;      # где хранить кэш с метаданными  
    no_index;          # не индексировать потоки  
    limits 90% 3G 1d;  # ограничения на архивы потоков  
    copy /opt/storage; # делать копию в /opt/storage  
    replicate port=2345; # включить репликацию (указывать на рестримере)  
    schedule 3-6,5-8,23-5;# когда производить запись  
  
    disk d1;           # диск в обычном режиме записи (чтение и запись)  
    disk d2 keep;       # не удалять данные  
    disk d3 readonly;   # только чтение  
    disk d4 abandon;    # не писать данные, можно удалять старое  
    disk d5 migrate;    # скопировать на другие диски и освободить диск  
    disk d6 rescue;     # скопировать на другие диски, но не удалять с диска  
}
```

Глобальные настройки DVR, которые действуют только для DVR на дисковом массиве:

`root` — базовый каталог, в котором смонтированы диски и находятся индексы.

`raid` — включает возможность работы с массивом (допустимое значение — 0). Если вы включили эту опцию, проверяется наличие активных дисков. Flussonic проверит `major device` корневого пути и файлов в каталогах и не разрешит запись, если это не смонтированная папка.

`active` — количество дисков, на которые будет идти запись данных. При большом количестве дисков вести запись сразу на все неэкономно по затратам электроэнергии, поэтому можно ограничиться несколькими дисками. Если не указывать эту опцию, то запись будет производиться на все диски, имеющие достаточно свободного места.

`metadata` — директория под `root` для хранения кэшированных метаданных. Эту директорию

создавать вручную не нужно, она будет создана при применении настроек. Мы рекомендуем использовать SSD для быстрого доступа к архиву.

disk — путь до примонтированного диска. Указанные в опции disk пути должны быть реальными точками монтирования. Например:

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/pve-vm--15--disk--1	7.9G	5.7G	1.8G	77%	/
/dev/loop0	196G	4.0G	182G	3%	/dvr/_raid_/d1
/dev/loop1	196G	4.0G	182G	3%	/dvr/_raid_/d2

Настройки, которые применимы к отдельным потокам (при указании в глобальных настройках DVR они будут по умолчанию применяться ко всем потокам. Их можно переопределить в настройках отдельного потока):

no_index — не добавлять поток в индекс DVR, отключая тем самым механизмы удаления данных для удаленных потоков.

limits — ограничения на размер и глубину архива. Например, 90% 3G 1d.

replicate — репликация DVR. Можно указать порт в необязательном параметре port=1234.

copy — копирование данных по частям в другое место.

schedule — расписание записи в архив.

Опции для управления записью на диски:

keep — Flussonic пишет данные, но никогда не удаляет. Используйте эту опцию, когда нужно хранить весь архив.

readonly — Flussonic не пишет данные на этот диск, возможно только чтение. Используйте для:

жесткого диска с большим износом, писать на который ничего нельзя (в том числе удалять файлы и обновлять индекс), и нельзя копировать. Диск будет использоваться в режиме только чтение, пока данные на нем не устареют.

работоспособного диска, когда-то записанного в режиме keep, если нужно хранить данные.

жесткого диска с большим износом, писать на который ничего нельзя (в том числе удалять файлы и обновлять индекс), и нельзя копировать. Диск будет использоваться в режиме только чтение, пока данные на нем не устареют.

работоспособного диска, когда-то записанного в режиме keep, если нужно хранить данные.

abandon — Flussonic не будет больше записывать новые данные на этот диск. Когда данные устареют, он удалит их. Но обновляет индекс. Используйте эту опцию, когда нужно заменить диск, например, на более емкий.

migrate — используйте, если срочно нужно извлечь диск, но данные нельзя терять. Flussonic на лету сохранит данные в массиве и удалит их с диска. При этом запись архива не прекращается. Вы узнаете, что данные смигрировались, если больше нет обращений к этому диску. При удалении диска из массива не нужно перезагружать Flussonic.

rescue — экстренно копирует данные со старого диска, который может скоро стать неработоспособным. Данные, в отличие от migrate, не удаляются. Архив с диска продолжает читаться, но новые данные не пишутся.

Авторизация

В Flussonic Media Server реализован механизм идентификации пользователей и отслеживания подключений с помощью авторизационных бэкендов. По протоколам HLS и HDS используются HTTP механизмы отслеживания сессий, а по протоколам RTMP, RTSP и MPEG-TS обрабатываются постоянные TCP сессии. Также отслеживается экспорт архива в формате MPEG-TS и MP4.

Процесс работы Flussonic Media Server с бэкендом описан в разделе «Авторизация через бэкенд».

Кроме того, Flussonic Media Server имеет встроенный механизм базовой защиты от вставки плеера на других сайтах. Более подробно про эту защиту вы можете прочесть в разделе Domain lock.

Flussonic Media Server также может проверять пароль при публикации потока. Более подробно про это вы можете прочесть в разделе Авторизация при публикации потока.

Авторизация через бэкенд

Flussonic Media Server поддерживает несколько авторизационных бэкендов.

Включение бэкенда

Бэкенды включаются путем добавления в конфигурационный файл директивы auth:

```
auth http://host;
```

Где host:

Оставлен пустым (по умолчанию)

Если у директивы auth не указана опция, то Flussonic Media Server разрешает все обращения.

Сетевой адрес HTTP

Если в качестве бэкенда указан сетевой адрес HTTP, то Flussonic Media Server будет делать HTTP запросы по этому адресу, передавая параметры сессии бэкенду.

Путь на диске

Если в качестве бэкенда указан путь на диске, то он интерпретируется как путь к Lua скрипту, который будет выступать в роли бэкенда. Подробнее про скриптовое расширение Flussonic Media Server вы можете прочесть в статье, посвященной Lua скриптам.

Процедура авторизации через бэкенд

Схематически работа с бэкендом выглядит так:

Более детальное описание процедуры авторизации:

Вы размещаете Flash-плеер или HTML тег video на веб-сайте или middleware, указывая в пути ключ авторизации (token), который был создан веб-сайтом, одним из этих способов:

В виде query string для HLS, HDS, HTTP MPEG-TS и других доступов по HTTP:

http://192.168.2.3:8080/stream1/manifest.f4m?token=60334b207baa

http://192.168.2.3:8080/stream1/index.m3u8?token=60334b207baa

В виде адреса для RTMP: rtmp application rtmp://192.168.2.3/static stream name:
stream1?token=60334b207baa

В виде адреса для RTSP: rtsp://192.168.2.3/stream1?token=60334b207baa

Если веб-сайт или middleware не указывает ключ token в пути к видео, то Flussonic Media Server генерирует token автоматически.

Если в конфигурационном файле есть глобальная опция no_auto_token, то Flussonic Media Server не будет генерировать token и будет сразу возвращать статус 403, запрещая доступ к контенту.

В виде query string для HLS, HDS, HTTP MPEG-TS и других доступов по HTTP:

http://192.168.2.3:8080/stream1/manifest.f4m?token=60334b207baa

http://192.168.2.3:8080/stream1/index.m3u8?token=60334b207baa

В виде адреса для RTMP: rtmp application rtmp://192.168.2.3/static stream name:
stream1?token=60334b207baa

В виде адреса для RTSP: rtsp://192.168.2.3/stream1?token=60334b207baa

Получив запрос к потоку с ключом token, сервер Flussonic Media Server проверяет, открыта ли сессия (транслируется ли уже поток с сервера на этот клиент). Идентификатором сессии служит хеш-сумма, создаваемая для имени потока, IP-адреса клиента и token следующим образом:

hash(name + ip + token)

Если пользователь меняет свой IP адрес или переключается на другой поток, создается новая сессия.

Если сессия не открыта, то Flussonic Media Server делает запрос к авторизационному бэкенду со следующими параметрами:

token.

Token, переданный с веб-сайта или сгенерированный автоматически

name.

Имя потока или файла

ip.

IP пользователя

referer.

HTTP Referer или RTMP pageUrl

total_clients.

Общее количество открытых сессий на сервере

stream_clients.

Количество открытых сессий на этом потоке

request_type.

Значение new_session если создается новая сессия или update_session, если перепроверяется старая

type.

Запрашиваемый протокол: hds, hls, rtmp, rtsp, mpegts или mp4

token.

Token, переданный с веб-сайта или сгенерированный автоматически

name.

Имя потока или файла

ip.

IP пользователя

referer.

HTTP Referer или RTMP pageUrl

total_clients.

Общее количество открытых сессий на сервере

stream_clients.

Количество открытых сессий на этом потоке

request_type.

Значение new_session если создается новая сессия или update_session, если перепроверяется старая

type.

Запрашиваемый протокол: hds, hls, rtmp, rtsp, mpegts или mp4

Если бэкенд возвращает статус HTTP 200, сессия открывается или продолжается.

Если бэкенд возвращает статус 403 или 401, сессия закрывается.

Если бэкенд возвращает статус HTTP 301 или 302, запрос перенаправляется на адрес из HTTP заголовка Location.

Все остальные статусы и таймауты интерпретируются как отсутствие данных, и запрос повторяется.

Сессия открыта

Если бэкенд разрешил открытие сессии, то, по умолчанию, Flussonic Media Server будет перепроверять сессию раз в 3 минуты, чтобы определять, что сессия всё ещё активна.

Чтобы поменять это время, отправьте HTTP заголовок X-AuthDuration. X-AuthDuration указывается в секундах.

Через 3 минуты (или другой промежуток времени, если он был изменен с помощью X-AuthDuration) запрос к сессии приведет к повторному обращению к бэкенду.

Если бэкенд недоступен или возвращает статус 500, то Flussonic Media Server сохранит предыдущий статус, полученный от бэкенда, и попыбует ещё раз обратиться к нему.

Важно! Если вы поменяли в файле конфигурации настройку auth (добавили её, например), то это новое значение не применится к сессиям, которые уже открыты.

Сессия закрыта

Если бэкенд запретил открытие сессии, то информация о ней кешируется на сервере. В случае если пользователь пытается ещё раз с тем же токеном открыть поток, Flussonic Media Server будет отказывать, не делая повторных обращений к бэкенду.

Просмотр видео в веб-интерфейсе

Администратор может просматривать любое видео в веб-интерфейсе Flussonic без авторизации. То есть обращения к бэкенду авторизации не производится.

Технически это реализовано так: при просмотре из веб-интерфейса генерируется специальный токен "ADM-xxx", который перехватывается Flussonic Media Server. Такой токен воспринимается как разрешение воспроизводить видео без авторизации.

Можно запретить администратору просматривать видео, защищенное при помощи механизма бэкенд авторизации.

Простейший пример скрипта авторизации (PHP)

Будем хранить авторизацию в файле auth.txt, заполненном такими данными:

```
user1:token1
user2:token2
user3:token3
```

Следующий скрипт на PHP будет проверять, содержится ли токен в этом файле, и если да - разрешать открытие сессии:

```
<?php

$get = print_r($_GET, true);
$token = $_GET["token"];
if(!$token || !strlen($token)) {
    header('HTTP/1.0 403 Forbidden');
    error_log("No token provided", 4);
    die();
}

$tokens = array();
$contents = explode("\n", file_get_contents("auth.txt"));
foreach($contents as $line) {
    if(strlen($line) > 3) {
        $parts = explode(":", $line);
        $tokens[$parts[1]] = $parts[0];
    }
}

if($tokens[$token]) {
    header("HTTP/1.0 200 OK");
    header("X-UserId: ".$tokens[$token]."\r\n");
    header("X-Max-Sessions: 1\r\n"); // Turn this on to protect from multiscreen
} else {
    header('HTTP/1.0 403 Forbidden');
}
?>
```

Сбор статистики с помощью X-UserId

Бэкенд при открытии сессии может отправить Flussonic Media Server HTTP заголовок X-UserId (к примеру, X-UserId: 100), который после закрытия сессии будет записан во внутреннюю базу данных вместе с данными о сессии. Вы можете запрашивать данные о сессии по протоколу MySQL с указанием X-UserId для сборки статистики.

Если бэкенд отправляет заголовок X-Unique: true вместе с X-UserId, то происходит отключение всех остальных открытых сессий, которые имеют такой же X-UserId. Важно отметить, что отключенные сессии на некоторое время остаются в памяти сервера и клиенты с теми же сочетаниями IP-адреса, имени потока и token не смогут получить доступ к контенту.

При использовании опции X-Unique следует генерировать различные token-ы при каждом обращении пользователя к странице.

Логирование запросов к бэкенду

Подробное описание того, как сделать логирование запросов с помощью PHP, находится в отдельной статье.

Таймаут авторизационного бекенда

В случае если авторизационный бекенд не успевает ответить за 3 секунды, то происходит следующая ситуация:

Состояние сессии Что происходит

Не открыта Не открывается

Разрешена Остается открытой

Запрещена Остается запрещенной

Конфигуратор бэкендов

Flussonic позволяет создавать авторизационные бэкенды в основном файле конфигурации.

Можно указать белые и черные списки для IP адресов, токенов, User-Agents, стран, несколько параллельных HTTP-бэкендов. Не требуется писать свои lua-скрипты.

Настройка авторизации

Добавьте в файл /etc/flussonic/flussonic.conf:

```
auth_backend main {
    allow ip 127.0.0.1;
    allow ip 192.168.0.1;
    allow ip 172.16/24;
    deny ip 8.8.8.8;
    allow country RU US;
    deny country GB;
    allow token test_token1;
    deny ua "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.2.10)";
    backend http://stalker-1.iptv.net/auth.php;
    backend http://stalker-2.iptv.net/auth.php;
}
```

allow — объявляет "белый" список, т.е. немедленно разрешает просмотр без дальнейших проверок.

deny — объявляет черный список, запрещает просмотр.

Flussonic проверяет правила в следующем порядке:

```
allow token
deny token
allow ip
deny ip
allow country (страна)
deny country (страна)
allow ua (useragent)
deny ua (useragent)
```

Опрашивает несколько параллельных бэкендов

Запрещает доступ, если не указано allow default.

Приоритет правил важен. Правила с более высоким приоритетом применяются немедленно, и тогда правила с более низким уже не учитываются. Например, если вы разрешили IP адрес, но клиентское приложение или устройство приходит с запрещенным токеном, доступ будет запрещен, т.к. у токена приоритет выше.

Чтобы использовать этот авторизационный бэкенд для потока, укажите auth://main:

```
stream ort {
    url udp://239.255.0.1:1234;
```



```
auth auth://main;
}
```

Правила срабатывают после перезагрузки конфигурации.

Опция allow default

Опция allow default разрешает доступ к видео всем клиентам (получателям), кроме перечисленных в опции deny. Если эта опция не указана, то доступ к потоку запрещается для всех клиентов, а не только для тех, которым вы запретили явно через deny.

Эта опция разрешает просмотр потока в случае, когда авторизационный бэкенд не отвечает по причине неработоспособности скрипта или ошибки на сервере. Таким образом, опция allow default оставляет возможность доступа к контенту в случае неработоспособного бэкенда.

Рассмотрим, как Flussonic поступает с разными ответами от бэкенда и как влияет опция allow default на решение о предоставлении доступа к видеопотоку.

Опция allow default в случае одного бэкенда

Если авторизационный бэкенд запрещает доступ (отвечает с кодом ошибки 4xx, например, 403 Forbidden), Flussonic не разрешит доступ к содержимому, даже если в конфигурации указано allow default.

Однако если бэкенд не отвечает (не работает по причине ошибки) или произошла ошибка сервера, на котором работает бэкенд-скрипт (с кодом ошибки 5xx, например, 500 Internal Server Error), Flussonic разрешает доступ к содержимому. Т.е. при помощи опции allow default Flussonic оставляет возможность доступа к контенту в случае неработоспособного бэкенда.

Опция allow default в случае нескольких бэкендов

Если у вас несколько параллельных бэкендов, правила примерно те же.

Если хотя бы один из бэкендов разрешает доступ, а остальные бэкенды запрещают его или не отвечают, то доступ будет разрешен.

Если хотя бы один из бэкендов запрещает доступ, а остальные не отвечают, т.е. ни один не разрешает, доступ будет запрещен.

Однако если все бэкенды не работают (не отвечают), Flussonic разрешает доступ к содержимому (при наличии в настройках потока опции allow default).

В таблице показана логика авторизации при использовании нескольких бэкендов для потока:

Backend 1	Backend 2	Backend 3	Результат	allow	allow	allow
ban	ban	Ban	ban	allow	ban	Allow
not responding	not responding	Allow	not responding	allow	allow	not responding
responding	Allow	not responding	ban	not responding	Ban	

Примеры

Мультиавторизация и доступ из локальной сети

```
auth_backend multi_local {
```

```
allow ip 192.168.0/24;
backend http://127.0.0.1/tv/auth; # iptv plugin
backend http://<HOSTNAME>/stalker_portal/server/api/chk_flussonic_tmp_link.php;
}
```

Заблокировать несколько пользователей, остальным разрешить

```
auth_backend blacklist {
    deny ip 1.1.1.1;
    deny ip 2.2.2.2;
    deny ip 10.10/16;
    allow default;
}
```

Использовать HTTP бэкенд и разрешить просмотр видео клиентам с указанными токенами

```
auth_backend myauth {
    allow token friend_token1;
    allow token friend_token2;
    backend http://<HOSTNAME>/stalker_portal/server/api/chk_flussonic_tmp_link.php;
}
```

Разрешить только свои приставки по User-Agent, остальных блокировать

```
auth_backend agents {
    allow ua MAG;
    allow ua TVIP;
}
```

Сервис статистики

Мы предоставляем облачный сервис для наших клиентов по сбору статистики и предоставлению доступа к ней в личном кабинете пользователя на нашем сайте.

Включить сбор статистики можно в личном кабинете, нажав на кнопку на карточке лицензии: После этого Flussonic начнет сохранять в наш облачный сервис сессии просмотра видео и вы незамедлительно увидите отчеты в личном кабинете.

Замечание. Сбор статистики и базовые отчеты доступны пользователям с активными подписками или разовыми лицензиями с действующим доступом к обновлениям.

Чем полезна статистика

С помощью нашего сервиса статистики вы сможете узнать:

- с каких устройств смотрели ваше видео
- сколько трафика ушло
- из каких стран смотрят
- какие каналы (или иные потоки) интереснее вашим пользователям

Эти данные помогут вам узнать точнее, что интересно, а что не интересно вашим пользователям: вы перестанете оплачивать ненужные каналы и добавите интересные.

Вы сможете с помощью нашего сервиса делать отчеты для своих клиентов о том, сколько трафика вы для них раздали и сколько просмотров было. Эти данные стоит сопоставлять с другими способами оценки статистики, чтобы получить более точную картину по расходам.

Анализ устройств, с которых смотрят видео, даст вам информацию для выбора качества контента: на телефоне достаточно среднего качества, а для телевизора лучше приготовить FullHD.

Географическое распределение ваших пользователей подскажет вам, куда стоит перенести серверы, или оставить всё как есть, расширяя имеющиеся инсталляции.

Какие данные мы храним

Каждый Flussonic с включенным сбором статистики передает на наши сервера:

- уникальный идентификатор сервера, генерирующийся произвольно на первичном запуске сервера
- время создания сессии
- время закрытия сессии
- имя канала или имя файла
- IP адрес пользователя
- уникальный идентификатор сессии
- авторизационный токен, использовавшийся при создании сессии
- протокол сессии

количество переданных в сессии байт

идентификатор пользователя, переданный с авторизационного бекенда

referer для этой сессии

User Agent этой сессии

Геопозиционирование и разбор User Agent происходит у нас в сервисе при записи сессий.

Трафик от Flussonic к нашему сервису чрезвычайно мал, и можно его не учитывать.

Что мы показываем в статистике

Мы показываем:

общую статистику, т.е. сколько всего было записано сессий за выбранный период

количество уникальных сессий (см. ниже пояснение)

суммарный трафик

суммарное время просмотра, т.е. общая длительность сессий

Отчеты можно посмотреть на разных вкладках меню (Dashboard, Servers и т.д.)

В верхней части страницы есть фильтры, которые применимы ко всем отчетам:

From - To — выбор диапазона дат

filter by stream — фильтр списка по имени канала (возможен поиск по подстроке - т.е. по части имени потока)

Дополнительные фильтры

Щелкнув по ссылке more filters, можно задать дополнительные фильтры:

Select servers - объем трафика через сервер

minimum duration — минимальная длительность одной сессии в секундах

filter by ip — IP адрес клиента

filter by user_id — идентификатор клиента (UserId, полученный от авторизационного бекенда)

useragent - программы, с помощью которых пользователь получил доступ к видео

maximum duration — максимальная длительность одной сессии в секундах.

Dashboard

На дашборде мы показываем:

популярные страны просмотра (сортированы по трафику)

популярные каналы (сортированы по количеству сессий)

популярные протоколы (сортированы по трафику)

Channels

На вкладке Channels мы показываем каналы, сортированные по трафику:

Sessions

В вкладке Sessions мы показываем сортированные по времени создания сессии, с распределением по дням (на вкладках).

Unique sessions

Содержание вкладки Unique sessions описано в разделе Уникальные сессии далее на этой странице.

Useragents

В вкладке Useragents мы показываем распределение типов устройств по количеству сессий и трафику, а также список User Agents.

Servers

Во вкладке Servers показаны сводные данные о всех серверах с Flussonic:

- сколько данных было скачано (Traffic)

- сколько было пользовательских сессий (Connections)

- какова общая длительность сессий (Duration).

Servers online

Во вкладке Servers online показаны сводные данные о серверах Flussonic, которые работают в текущий момент. Здесь Last updated показывает, когда снималась статистика.

Users

Вкладка Users показывает данные, сгруппированные по UserId. Показ статистики возможен для тех пользователей, которым авторизационный бекенд отдает UserId.

- Traffic - общий трафик пользователя

- Count - количество сессий

- View time - суммарная длительность пользовательских сессий.

Worldmap

Вкладка Worldmap показывает статистику по странам:

- статистика по количеству просмотров в стране

- статистика по объему трафика, скачанного в стране

Для переключения на статистику по просмотрам нажмите сверху слева в отчете кнопку Switch to views.

А для переключения на статистику по объему трафика Switch to traffic.

Durations

Вкладка Durations показывает срез по длительностям сессий. По горизонтали показана длительность сессий, а по вертикали количество сессий и суммарный трафик.

Уникальные сессии

Мы в реальном времени можем сделать агрегацию сессий, которые выглядят похожими, т.е. если в течение дня есть сессии, у которых совпадает IP адрес, имя, протокол, user_id, Referer, User-Agent, то мы их склеиваем вместе и показываем их во вкладке Unique sessions, а их суммарное количество наверху в блоке Summary. Также в разделе Unique sessions показываем количество таких склеенных сессий.

Эти данные помогут увидеть, что клиент часто переподключается или что есть проблемы с учетом сессий.

Domain lock

Защита от вставки плеера на других сайтах работает с клиентами, передающими Referer в той или иной форме, т.е. преимущественно с флеш-плеером. К примеру, iOS устройства не передают Referer при обращении к видео серверу.

Включите эту защиту в конфигурационном файле следующим образом:

```
live user15 {  
    domains myhost.com *.myhost.com;  
}
```

Важно отметить, что это достаточно простая защита, спасающая лишь от простых схем встраивания.

Авторизация в Flussonic через Middleware

Middleware

Очень важная задача, которую надо решить при запуске OTT IPTV сервиса — ограничение доступа к стриминговым серверам. По нашей статистике многие люди вообще не обращают на это внимание и, как следствие, переплачивают за трафик — их потоки попросту воруют.

Видео можно раздавать всем, но хитро зашифрованное и не всем отдавать ключи, это называется DRM. Другой способ защиты — ограничивать раздачу самого видео, это уже авторизация.

В Flussonic Media Server реализована очень гибкая схема авторизации, требующая определенных действий со стороны Middleware.

Схема работы такая:

- клиентская приставка обращается за адресом потока;
- Middleware отдает адрес с уникальным токеном;
- Flussonic Media Server использует этот токен для идентификации сессии;
- при открытии сессии Flussonic Media Server проверяет этот токен у Middleware.

Такая трехсвязная схема нужна для того, что бы не встраивать авторизацию в Flussonic Media Server. В свою очередь Flussonic Media Server не на каждый запрос клиента ходит к Middleware, а только раз в определенное время.

Вопрос правильного выбора токена открытый и мы можем предложить пару вариантов по его генерации.

Можно генерировать токены, упаковывая в них всю необходимую для авторизации информацию. Например, токен можно сгенерировать таким образом:

```
token=sha1(secret_key + ip + stream_name)
```

Такой токен можно проверить, зная лишь secret_key. При этом, если злоумышленник попытается воспользоваться этим токеном, у него ничего не получится, потому что IP будет другим.

Однако такой токен можно сохранить и пользоваться им бесконечно. Если пользователь один раз оплатил подписку на сервис, то потом с таким токеном ему можно уже не платить.

В токен можно вставить время:

```
time = utc()  
token=sha1(secret_key + ip + stream_name + time)+":"+time
```

Теперь Middleware при проверке может проверить время жизни токена и, если ему больше суток, может его отключать. На практике почти никто (кроме публичных телевизоров и поклонников 24 Le Mans) не в состоянии смотреть эфир больше суток подряд.

Можно совместить авторизацию с учетом просмотра и под каждый просмотр создавать пользователю новый уникальный токен, помещая его в базу:

```
token=uuid()
```

Потом при повторных обращениях Flussonic Media Server к Middleware можно обновлять статистику по этой сессии, сохраняя информацию о том, кто сколько просмотрел.

Защита доступа к потокам (Авторизация по токену)

В этой статье приведен пример того, как можно реализовать систему авторизации без написания собственного бэкенда.

Схема работы авторизации:

Ваш сайт генерирует токен по несложной формуле и хеширует его с помощью секретного ключа.

Клиент открывает поток с полученным токеном.

Flussonic генерирует токен по той же формуле, используя тот же ключ (и дополнительно используя имя потока и IP адрес клиента).

Хеши совпали — доступ разрешен. Не совпали — запрещен.

Настройка Flussonic для использования авторизации по токену

В поставке Flussonic есть вся необходимая логика для проверки генерируемых токенов. Достаточно просто указать опцию `securetoken` и пароль для авторизации:

```
stream example-stream {  
    url fake://fake;  
    auth securetoken://SECRETKEY;  
}
```

Можно включить авторизацию (директиву `auth`) как для одного потока, так и глобально.

Код для сайта

Чтобы сгенерировать токен, Flussonic должен знать следующее:

IP-адрес клиента

Имя потока

Секретный ключ

Текущее время

Код на сайте должен собрать в одну строку эти данные:

```
string = streamname + ip + starttime + endtime + secretkey + salt
```

И получить токен по формуле:

```
sha1(string) + salt + endtime + starttime.
```

Где:

`ip` — IP адрес клиента.

`name` — название потока.

`starttime` — обычно это текущее время UTC (в Unix Timestamp).

endtime — время окончания жизни токена, обычно это текущее время + несколько часов. По прошествии этого времени токен перестанет работать и его надо будет запрашивать заново.

secretkey — это ключ, указанный в файле /etc/flussonic/flussonic.conf.

salt — строка из случайных символов, необходима, чтобы для одинаковых входных данных генерировались разные токены.

PHP пример

```
<?php

$flussonic = 'http://flussonic-ip'; // flussonic address
$key = 'SECRETKEY'; // key from flussonic.conf file. KEEP IT IN SECRET
$lifetime = 3600 * 3; // 3 hours after link will be invalid

$stream = $_GET['stream']; // this script get streamname from a query string (script.php?stream=bbc)

$ipaddr = $_SERVER['REMOTE_ADDR'];
$desync = 300; // allowed time desync between flussonic and hosting servers in seconds
$starttime = time() - $desync;
$endtime = $starttime + $lifetime;
$salt = bin2hex(openssl_random_pseudo_bytes(16));

$hashsrt = $stream.$ipaddr.$starttime.$endtime.$key.$salt;
$hash = sha1($hashsrt);

$token = $hash.'-'. $salt.'-'. $endtime.'-'. $starttime;
$link = $flussonic.'/'. $stream.'/embed.html?token='.$token;
$embed = '<iframe allowfullscreen style="width:640px; height:480px;" src="'. $link.'"></iframe>';

echo $embed;
```

Rails пример

config/routes.rb:

```
Rails.application.routes.draw do
  ...
  get '/securetoken/:id', to: 'securetoken#index'
end
```

app/controllers/securetoken_controller.rb:

```
class SecuretokenController < ApplicationController

  def index

    flussonic = 'http://flussonic-ip'
    secret = 'SECRETKEY'

    streamname = params[:id]
```

```

lifetime = 3600 * 3
starttime = Time.now.to_i - 300
endtime = Time.now.to_i + lifetime
salt = rand(8**8).to_s(8)

hash = Digest::SHA1.hexdigest(streamname + request.remote_ip + starttime.to_s + endtime.to_s + secret + salt)
token = hash + '-' + salt + '-' + endtime.to_s + '-' + starttime.to_s
@url = flussonic + '/' + streamname + '/' + 'embed.html?token=' + token
end
end

app/views/securetoken/index.html.erb:
<iframe allowfullscreen style="width:640px; height:480px;" src="<%= @url %>"></iframe>

```

Ограничение количества сессий на пользователя (защита от кражи)

Ограничение сессий

Для того чтобы любой ваш пользователь получивший доступ к потокам не устроил полное зеркалирование на свой сервер (например с целью дальнейшей перепродажи), во Flussonic Media Server есть возможность ограничить количество одновременно просматриваемых потоков.

Таким образом, даже получив доступ ко всем потокам, пользователь сможет просматривать одновременно лишь N и попытки зеркалировать все потоки ни к чему не приведут.

Ограничение производится на каждого пользователя, который имеет свой UserId и задаётся с помощью авторизации.

Детали

Для того чтобы ограничить количество сессий до 2, нужно в авторизационном бэкенде отдавать следующие заголовки:

X-UserId: some-user-id

X-Max-Sessions: 2

А через lua-бэкенд поля user_id и max_sessions соответственно.

Если после такой авторизации пользователь попытается просматривать одновременно три потока, то просмотр одного из них будет прерван.

Бан

После того как сессия была забанена, любые попытки переоткрыть её в течении

X-AuthDuration

будут отклонены Flussonic Media Server.

Таким образом, если указать X-AuthDuration: 3600, и открыть лишний поток, то после блокировки этот поток будет нельзя открыть используя старый токен в течении часа.

После того как сессия будет заблокирована на очередной запрос HLS-плейлиста клиент получит ответ 403 Forbidden. В случае же с RTSP, RTMP, HTTP MPEG-TS сокет будет просто молча закрыт.

Каждая заблокированная сессия сопровождается записью в лог вроде:

```
14:58:51.598 <0.391.0> [stream-  
name] session_limiter:174 Ban session_id: <<"604551981e3e787b897afbaf35bb9f4d168d70b9">> for user_  
id: <<"8471796306">> and token: <<"5cfb82ecaf56ebfe7ac32a9020c86ef1d231d49e">> due to exceded  
session limit
```

Мягкое ограничивание

Некоторые Middleware не могут генерировать новый токен на каждый запрос HLS потока. При переключении между потоками это может создать проблемы, так как сессии на старых потоках будут помечены как лишние, и будут заблокированы.

Специально для этих случаев в Flussonic Media Server начиная с 4.5.23 версии есть механизм мягкого ограничения сессий.

Иногда блокировка не происходит в первую проверку, необходимо время чтобы понять что все сессии действительно используются. Тогда она происходит во вторую или третью. Таким образом, после появления лишних сессий они обычно блокируются через 30-90 секунд.

Для того чтобы включить этот режим, необходимо указать дополнительный ключ `soft_limitation=true` к опции `auth`, например:

```
stream foobar {  
  auth http://localhost:8081/my_auth_script.php soft_limitation=true;  
}
```

X-Unique: true

Заголовок

X-Unique

считается устаревшим, вместо него предполагается использовать

X-Max-Sessions

описанный выше.

X-UserId: some-id

X-Unique: true

Эквивалентно:

X-UserId: some-id

X-Max-Sessions: 1

Кроме того, если указаны и X-Max-Sessions и X-Unique, то приоритет отдаётся X-Max-Sessions. Таким образом:

X-UserId: some-id

X-Max-Sessions: 5

X-Unique: true

Эквивалентно:

X-UserId: some-id

X-Max-Sessions: 5

Замечания по версиям

В версии 4.5.5 и выше Flussonic Media Server может разрешать N сессий, а не только одну. (X-Unique: true)

В версии 4.5.13 и выше период перепроверки сессии через бэкенд (X-AuthDuration) по умолчанию равен 180 секунд (3 минуты), вместо 30 секунд.

В версии 4.5.15 и выше возвращаемый auth_time из lua-бэкенда трактуется как секунды (до этого -- миллисекунды), по-анalogии с X-AuthDuration http-бэкенда.

Как настроить два авторизационных бекенда

Если вы уже используете IPTV Middleware, такое как IPTVPortal, Stalker или подобное, то вы можете столкнуться с ситуацией, когда надо добавить определенные исключения в авторизацию.

Например, передать партнеру ссылку на стрим с паролем, но при этом не хочется модифицировать код самой Middleware. Для этого понадобится написать небольшой блок в конфигурационном файле.

Авторизационный бекенд с исключениями

Предположим, что авторизационный бекенд вашей Middleware — `http://iptv.myservice.com/auth.php`

Вы хотите чтобы ваш партнер смог забрать стримы по паролю PASS, а также хотите разрешить доступ из локальной сети 192.168.1.0/24.

Добавьте в файл `/etc/flussonic/flussonic.conf`:

```
auth_backend main {
    allow ip 192.168.1/24;
    allow token PASS;
    backend http://iptv.myservice.com/auth.php;
}
```

Здесь мы сначала проверяем IP-адрес и токен клиента. Если они не подходят, то идем к оригинальному бэккенду.

Чтобы использовать этот (main) авторизационный бэккенд для потока, укажите `auth://main`:

```
stream cnn {
    url udp://239.255.0.1:1234;
    auth auth://main;
}
```

Аналогичным образом можно организовать два разных авторизационных бекенда.

Два авторизационных бекенда

```
auth_backend parallel {
    backend http://DOMAIN.iptvportal.ru/auth/flussonic/arescrypt/;
    backend http://stalker/stalker_portal/server/api/chk_flussonic_tmp_link.php
}
```

```
stream cnn {
    url udp://239.255.0.1:1234;
    auth auth://parallel;
}
```

В этом случае реализуется параллельная авторизация на нескольких HTTP-бэкендах.
Более подробно о конструкторе бэкендов.

Как ограничить доступ по IP адресам

Иногда возникает необходимость ограничить доступ к каналам определенным IP адресам. Или наоборот, разрешать доступ к каналу только определенному списку IP. Эту задачу можно решить с помощью бэкенда, собранного с помощью конструктора бэкендов.

С помощью конфигулятора можно настроить очень гибкие схемы авторизации. На этой странице приведем примеры как заблокировать определенные IP-адреса, или наоборот, пустить мимо системы авторизации. Это может быть полезно для систем мониторинга.

Все правила, описанные далее, могут быть применены к потоку или глобально.

```
stream ort {  
    url udp://239.255.0.1:5500;  
    auth auth://blacklist;  
}
```

Где blacklist - имя одного из настроенных бэкендов. Конечно же, вы можете создать более одного авторизационного правила.

Заблокировать

Вся настройка происходит через /etc/flussonic/flussonic.conf.

```
auth_backend blacklist {  
    deny ip 1.1.1.1;  
    deny ip 2.2.2.2;  
    deny ip 10.10/16;  
    allow default;  
}
```

Это правило запрещает просмотр с двух определенных адресов (1.1.1.1, 2.2.2.2) и целой подсети (10.10.0.0/16).

Строка allow default; означает, что по умолчанию нужно всем разрешать просмотр, кроме адресов, перечисленных в deny. Подробнее об опции

Разрешить

```
auth_backend whitelist {  
    allow ip 192.168.0/24;  
    allow ip 10.10/16;  
    allow ip 8.8.8.8;  
}
```

Это правило разрешает просмотр только из указанных сетей и конкретного IP-адреса. Остальные соединения будут блокироваться.

```
auth_backend multi {  
    allow ip 192.168.0/24;  
    backend http://<HOSTNAME>/stalker_portal/server/api/chk_flussonic_tmp_link.php;  
}
```


Это правило разрешает просмотр из локальной сети, а остальные обращения через IPTV Middleware.

Авторизация доступа к архиву

Подробнее об организации архива можно прочесть в статье по ссылке.

Авторизация доступа к архиву потока осуществляется точно так же как и к самому потоку.

Исключением является экспорт архива в файл с сохранением на диске сервера, в этом случае требуется административный доступ к серверу.

При использовании авторизации бэкенду авторизации передаются параметры `type` и `media_request`.

Параметр `type` описывает запрашиваемый протокол: `hds`, `hls`, `rtmp`, `rtsp`, `mpegts` или `mp4`.

Параметр `media_request` уточняет, что именно именно запрошено.

Возможные соответствия между URL и значением параметра `media_request` описаны в таблице.

В первой колонке указано только окончание URL, т.е. значение `/STREAMNAME/index.m3u8` на самом деле обозначает URL типа `http://flussonic-ip/STREAMNAME/index.m3u8`

URL	media_request
-----	---------------

/STREAMNAME/index.m3u8	hls_live-hls_mbr_playlist
------------------------	---------------------------

/STREAMNAME/tracks-1,2/index.m3u8	hls_live-hls_track_playlist
-----------------------------------	-----------------------------

/STREAMNAME/mono.m3u8	hls_live-hls_playlist
-----------------------	-----------------------

/STREAMNAME/manifest.f4m	hds_live-hds_manifest
--------------------------	-----------------------

/STREAMNAME/bootstrap	hds_live-bootstrap
-----------------------	--------------------

/STREAMNAME/mpegts	mpegts_handler-request
--------------------	------------------------

/STREAMNAME/index-1362504585-3600.m3u8	dvr_session-hls_mbr_playlist-1362504585-3600
--	--

/STREAMNAME/tracks-1,2/index-1362504585-3600.m3u8	dvr_session-hls_track_playlist-1362504585-3600
---	--

/STREAMNAME/mono-1362504585-3600.m3u8	dvr_session-hls_playlist-1362504585-3600
---------------------------------------	--

/STREAMNAME/archive-1362504585-3600.mp4	dvr_handler-mp4-1362504585-3600
---	---------------------------------

/STREAMNAME/archive-1362504585-3600.ts	dvr_stream_handler-ts_file-1362504585-3600
--	--

/STREAMNAME/archive/1362504585/3600/manifest.f4m	dvr_session-hds_manifest-1362504585-3600
--	--

/STREAMNAME/archive/1362504585/3600/bootstrap	dvr_session-bootstrap-1362504585-3600
---	---------------------------------------

/STREAMNAME/archive/1362504585/3600/mpegts" dvr_stream_handler-ts_stream-1362504585-3600

/STREAMNAME/timeshift_abs-1362504585.ts dvr_handler-timeshift_abs-1362504585

/STREAMNAME/timeshift_rel/3600 dvr_handler-timeshift_rel-3600

/STREAMNAME/timeshift_abs-1362504585.m3u8 hls_timeshift_playlist-hls_timeshift_abs-1362504585

/STREAMNAME/tracks-1/timeshift_abs-1362504585.m3u8 hls_timeshift_playlist-hls_track_timeshift_abs-1362504585

/STREAMNAME/timeshift_abs_mono-1362504585.m3u8 hls_timeshift_playlist-hls_track_timeshift_abs-1362504585

/STREAMNAME/timeshift_rel-360.m3u8 hls_timeshift_playlist-hls_timeshift_rel-360

/STREAMNAME/tracks-1/timeshift_rel-360.m3u8 hls_timeshift_playlist-hls_track_timeshift_rel-360

/STREAMNAME/timeshift_rel_mono-360.m3u8 hls_timeshift_playlist-hls_track_timeshift_rel-360

Псевдонимы для имен потоков

В Flussonic есть функция «Алиасер» — создание псевдонимов для имен потоков и файлов. Она позволяет спрятать имена потоков от конечных пользователей. На каждый запрос от пользователя имя будет переписано на внутреннее имя, которое вы указали.

Итак, у вас есть поток «clock» и обычно вы даете пользователям ссылку `http://192.168.2.3/clock/index.m3u8`.

Теперь вы можете оставить имя «clock», но давать людям ссылку `http://192.168.2.3/crf7930803e4e334e104/index.m3u8`.

Это может пригодиться в ситуациях, если вам нужно:

- спрятать имена потоков или файлов от пользователей, создать временные имена;
- управлять именами потоков, что позволит делиться потоками, отзываться embed.

Чтобы включить «Алиасер», вам нужно написать бэкэнд с необходимой логикой и включить его в настройках Flussonic:

```
aliaser /opt/flussonic/priv/rewrite.lua;
```

Можно указать только один lua бэкэнд. Он должен вернуть false или имя потока. Объект req с полем name передается бэкэнду.

Если включен «Алиасер», то Flussonic будет передавать авторизационному бэкэнду новый параметр:

`user_name` - оригинальное имя запрашиваемого потока.

Пример перезаписи бэкэнда

Мы покажем вам пример сценария перезаписи, который поможет вам понять механизм работы «Алиасера».

Представим себе, что мы хотим скрыть исходное имя потока clock от пользователя и дать ему временную ссылку.

Создайте файл `/etc/flussonic/alias.lua` с кодом:

```
a = {}
```

```
a["alias"] = "clock"
```

```
if a[req.name] ~= nil then
    return a[req.name]
else
    return req.name
end
```

и затем включите его в `/etc/flussonic/flussonic.conf`:

```
# Global settings:
```

...

```
aliaser /etc/flussonic/alias.lua;
```

Теперь мы можем получить доступ к потоку clock через два имени:

```
http://flussonic/clock/index.m3u8
```

```
http://flussonic/alias/index.m3u8
```

Вы можете добавить дополнительные псевдонимы для своих потоков. Например:

```
a = {}
```

```
a["alias"] = "clock"
```

```
a["alias2"] = "clock"
```

```
a["alias3"] = "clock"
```

```
a["bbc-news"] = "bbc"
```

```
a["bbc-entertainment"] = "bbc"
```

```
a["BBC"] = "bbc"
```

```
if a[req.name] ~= nil then
```

```
    return a[req.name]
```

```
else
```

```
    return req.name
```

```
end
```

Пример перезаписи бэкэнда: криптографическая версия

Мы покажем вам пример сценария перезаписи, который поможет вам понять механизм работы «Алиасера».

Представим себе, что мы хотим скрыть исходное имя потока clock от пользователя и дать ему временную ссылку.

Так как мы не хотим связываться с базами данных, то будем просто шифровать имя потока с помощью известного ключа и предоставлять его пользователю.

Например, наш ключ будет 000102030405060707090A0B0C0D0E0F (16 байт), IV будет заполнен нулями, поэтому шестнадцатеричное значение зашифрованной строки clock с помощью aes ctr encryption будет: a5cd5454ec

Чтобы расшифровать его в нашем сценарии lua, нам нужно будет написать следующий бэкэнд:

```
decrypted = crypto.aes_ctr_decrypt(crypto.from_hex("000102030405060708090A0B0C0D0E0F"), crypto.from_hex(req.name))
```

```
return decrypted
```

Новое имя потока не будет уникальным, оно будет таким же.

Добавим к нашему имени потока некоторую случайную последовательность 4 байта. Теперь зашифрованные «1234» + «clock» будут: f7930803e4e334e104

Теперь на нашем веб-сайте нам нужно добавить 4 случайных байта перед нашим потоком и затем зашифровать. Бэкэнд будет выглядеть довольно просто:

```
decrypted = crypto.aes_ctr_decrypt(crypto.from_hex("000102030405060708090A0B0C0D0E0F"), crypto.from_hex(req.name))
```

```
m_hex(req.name))  
return string.sub(decrypted,5)
```

HTTP API к Flussonic Media Server

Аутентификация и авторизация

Flussonic Media Server предоставляет возможность получать информацию и управлять определенной функциональностью по HTTP.

Запросы на получение информации можно защитить с помощью директивы `view_auth user password`; в конфигурационном файле.

Запросы на модификацию состояния Flussonic Media Server можно защитить с помощью директивы `edit_auth user password`; в конфигурационном файле `/etc/flussonic/flussonic.conf`.

При включенной авторизации для доступа к HTTP API необходимо указывать логин и пароль в формате HTTP Basic Auth.

Информация о сервере (server)

Информация о сервере Flussonic Media Server.

URL: `/flussonic/api/server`

Пример: `http://example.flussonic.com:8080/flussonic/api/server`

Параметры: нет

Ответ: JSON вида

```
{
  "version": "4.6.1", // версия сервера Flussonic Media Server.
  "hostname": "streamer.example.com", // hostname
  "uptime": 43373, // Время работы Flussonic в секундах.
  "total_clients": 1592, // Общее число клиентов.
  "total_streams": 100, // Общее количество потоков.
  "online_streams": 95, // Общее количество активных потоков.
  "input_kbit": 1234, // Текущая скорость входящего трафика.
  "output_kbit": 123456 // Текущая скорость исходящего трафика.
}
```

Список потоков, их клиенты и состояние (media)

Состояние и настройки потоков, количество клиентов.

Вы можете запросить все потоки или указать только несколько потоков через запятую.

Чтобы получить список всех потоков:

URL: `/flussonic/api/media`

Параметры: нет

Чтобы получить список только указанных потоков:

URL: `/flussonic/api/media?name=STREAMNAME`

URL: `/flussonic/api/media?name=STREAMNAME1,STREAMNAME2`

Параметры:

name=STREAMNAME — имя потока.

Необязательный параметр

Ответ: JSON вида

```
[ //Список транслируемых потоков
{
  "entry": "stream",
  "value": {
    "name": "euro", //Название потока
    "urls": [ //Список всех источников
      {
        "value": "tshttp://192.168.1.2:6502", //URL источника
        "options": [ //Опции резервирования источника
          [
            "priority",
            "1"
          ],
          [
            "source_timeout",
            "30"
          ]
        ]
      }
    ],
    "stats": {
      "alive": true, //Отдавал ли поток кадры в последнее время
      "bitrate": 3690, //Битрейт
      "bufferings": 0,
      "client_count": 0, //Количество клиентов, смотрящих поток
      "dash": true, //Включен ли DASH
      "hds": true, //Включен ли HDS
      "hls": true, //Включен ли HLS
      "input_error_rate": 0, //количество регистрируемых ошибок в секунду
      "last_access_at": 1493279230436,
      "media_info": { //Информация о содержимом потока
        "height": 576, //Высота изображения
        "streams": [
          {
            "bitrate": 191, //Битрейт
            "codec": "mp2a", //Кодек
            "content": "audio", //Тип содержимого:аудио
            "lang": "eng", //Язык
            "track_id": "a1" //Номер трека
          },
          {
            "bitrate": 3256, //Битрейт
            "codec": "mp2v", //Кодек
            "content": "video", //Тип содержимого: видео
```



```

        "size": "1024x576", //Размеры изображения
        "track_id": "v1" //Номер трека
    }
],
    "width": 1024 //Ширина изображения
},
    "out_bandwidth": 4002, //Скорость отдачи по сети
    "push_stats": { //Статистика копирования потока, количество байт
        "tshttp://container4:8080/static1/mpegts": 2000918592
    },
    "remote": false, //Зеркалирование потока с удаленного сервера
    "retry_count": 0, //Количество автоматических перезапусков
    "running": true, //Ведется вещание потока, не обязательно означает, что в потоке есть кадры
    "start_running_at": 1493279194382,
    "ts_delay": 113, //Миллисекунды с момента когда в потоке был кадр
    "url": "tshttp://192.168.1.2:6502" //URL текущего источника
},
    "options": { //Информация о конфигурации потока
        "static": false,
        "retry_limit": 10,
        "clients_timeout": 60,
        "source_timeout": 60,
        "pushes": [
            [
                "tshttp://container4:8080/static1/mpegts"
            ]
        ],
        "add_audio_only": false,
        "dash_off": false,
        "dvr_protected": false,
        "hds_off": false,
        "hls_off": false,
        "m4f_off": false,
        "m4s_off": false,
        "mpegts_off": false,
        "pulse_off": false,
        "rtmp_off": false,
        "rtsp_off": false,
        "webrtc_off": false
    }
}
},
...
]

```

Информация о потоке (media_info)

Информация о конкретном потоке: ширина, высота, описание дорожек.

URL: /flussonic/api/media_info/STREAM_NAME

Параметры:

STREAM_NAME — имя потока.

Обязательный параметр

Ответ: JSON вида

```
{
  "width":1024, //Ширина изображения
  "height":576, //Высота изображения
  "streams":[ //Список элементов, составляющих поток
    {
      "content":"video", //Тип содержимого. Видео.
      "codec":"h264", //Кодек h264
      "track_id":1 //Номер трека. Элементы отсортированы по нему.
    },
    {
      "content":"audio", //Тип содержимого. Аудио.
      "codec":"aac", //Кодек AAC
      "lang":"ukr", //Язык. Украинский.
      "track_id":2 //Номер трека.
    },
    {
      "content":"audio", //Тип содержимого. Аудио.
      "codec":"aac", //Кодек AAC
      "lang":"rus", //Язык. Русский.
      "track_id":3 //Номер трека.
    }
  ]
}
```

Информацию о потоке могут получать и клиенты, авторизованные для просмотра этого потока:

curl http://192.168.2.3:8080/ort/media_info.json

```
{
  "width":320,"height":240,"streams":[{"size":"320x240","content":"video","codec":"h264","bitrate":115,"track_id":1,"fps":25.0,"width":320,"height":240,"pixel_width":320,"pixel_height":240,"sar_width":1,"sar_height":1,"length_size":4,"profile":"Baseline","level":"2.1"},{"content":"audio","codec":"aac","bitrate":25,"track_id":2}]
}
```

Информация об исходном потоке (input_media_info)

Информация об оригинальном входящем потоке до транскодирования.

URL: /flussonic/api/input_media_info/STREAM_NAME

Параметры:

STREAM_NAME — имя потока.

Обязательный параметр

Ответ: JSON вида

```
{
  "height": 240,
  "streams": [
    {
      "codec": "h264",
      "content": "video",
      "fps": 25,
      "height": 240,
      "length_size": 4,
      "level": "2.1",
      "pixel_height": 240,
      "pixel_width": 320,
      "profile": "Baseline",
      "sar_height": 1,
      "sar_width": 1,
      "size": "320x240",
      "track_id": "v1",
      "width": 320
    },
    {
      "bitrate": 25,
      "codec": "aac",
      "content": "audio",
      "lang": "eng",
      "track_id": "a1"
    }
  ],
  "width": 320
}
```

Если поток не транскодируется, то в ответе будет ошибка:

```
{
  "error": "no_transcoder"
}
```

Проверка качества вещания потока (stream_health)

HTTP status code сигнализирует о том, когда в последний раз в потоке были кадры.

URL: /flussonic/api/stream_health/STREAM_NAME

Параметры:

STREAM_NAME — имя потока.

Обязательный параметр

Ответ запроса:

HTTP 200 — если поток по имени STREAM_NAME приходит и последний кадр на нём был относительно недавно (меньше секунды назад).

HTTP 424 — если в потоке давно не было кадров.

URL пригоден для настройки monit, например:

```
check process flussonic
start program = "/etc/init.d/flussonic start"
stop program = "/etc/init.d/flussonic stop"
if failed host localhost port 8080
protocol HTTP request "/flussonic/api/stream_health/cam0" then restart
if 5 restarts within 5 cycles then timeout
```

Список открытых файлов (files)

Список транслируемых файлов и количество клиентов, которые их просматривают.

URL: /flussonic/api/files

Параметры: нет

Ответ: JSON вида

```
{
  "files":[ //Список транслируемых файлов
    {
      "name":"vod/ir.mp4", //Внутренний идентификатор с префиксом
      "worker_count":1, //Сколько запущено потоков чтения этого файла
      "client_count":1, //Количество клиентов, смотрящих поток
      "url":"priv/ir.mp4", //Путь на диске или внешнем хранилище
      "bytes_out":1792522, //Количество отданных данных
      "bytes_in":1792522, //Количество считанных данных
      "prefix":"vod" //Префикс, в который добавлен файл
    }
  ]
}
```

Список открытых сессий (sessions)

Список открытых сессий, т.е. соединений клиентов с сервером. Если клиент нажал на паузу в проигрывателе или отключено автоматическое проигрывание видео (autoplay), то через некоторое время сессия будет закрыта.

URL: /flussonic/api/sessions

Параметры: нет

Ответ: JSON вида

```
{
  "sessions":[ //
    {
      "id":"3100c9414a8666450a47246520a921076f4738e5",
      //Неуникальный идентификатор сессии
      "session_id":
      "3100c9414a8666450a47246520a921076f4738e5-1396588629135",
      //Уникальный идентификатор сессии
    }
  ]
}
```

```

    "ip": "127.0.0.1", //IP клиента
    "name": "vod/ir.mp4", //Внутренний идентификатор с префиксом
    "created_at": 1396588629135, //Время старта сессии
    "duration": 5109, //Длительность сессии
    "type": "hds", //Тип потока
    "bytes_sent": 828010, //Количество посланных по сети байт
    "country": "NONE", //GeoIP location пользователя
  }
]
}

```

Для того, что бы вывести список подключений только одного потока или файла, надо указать имя потока в query string: /flussonic/api/sessions?name=vod/ir.mp4

Список открытых сессий для потока (sessions+stream_name)

Список сессий открытых к заданному потоку.

URL: /flussonic/api/sessions?name=STREAM_NAME

Параметры:

STREAM_NAME — имя потока.
Обязательный параметр

Ответ: JSON вида

```

{
  "name": "euro", //Имя потока
  "sessions": [ //Список сессий
    {
      "id": "e927bf4dca1ea90622318d4b4a0a60ab650bc6d9",
      //Уникальный идентификатор сессии
      "session_id":
      "e927bf4dca1ea90622318d4b4a0a60ab650bc6d9-1396867792624",
      //Уникальный идентификатор сессии
      "ip": "127.0.0.1", //IP клиента
      "name": "euro", //Имя потока
      "created_at": 1396867792624, //Время старта сессии
      "duration": 30614, //Длительность сессии
      "type": "hls", //Тип потока
      "bytes_sent": 5712330, //Количество посланных по сети байт
      "country": "NONE" //GeoIP location пользователя
    }
  ]
}

```

Закреть сессию (close_sessions)

Чтобы закрыть несколько активных сессий необходимо передать список их идентификаторов в

теле POST запроса. Используйте \n в качестве разделителя.

URL: /flussonic/api/close_sessions

Параметры:

HTTP request payload — список идентификаторов сессий через \n
Обязательный параметр

Ответ код 200

Информация о состоянии плейлиста (playlist)

HTTP API позволяет запросить текущее состояние серверного плейлиста и получить JSON ответ с данными.

URL: /flussonic/api/playlist/STREAM_NAME

Пример: http://example.flussonic.com:8080/flussonic/api/playlist/example_stream

Параметры:

STREAM_NAME — имя потока.
Обязательный параметр

Ответ: JSON вида

```
{
  "current_entry": "vod/ir.mp4", //Идентификатор текущего элемента в плейлисте
  "current_type": "file", //Тип текущего элемента в плейлисте
  "duration": null, //Длительность текущего элемента в миллисекундах (null - неизвестна)
  "position": 5.22e4 //Позиция проигрывания внутри текущего элемента, в миллисекундах
}
```

Карта записи архива за сутки (dvr_status)

Карта записи архива за определенные сутки, состоящая из сегментов и секунд.

URL: /flussonic/api/dvr_status/YEAR/MONTH/DAY/STREAM_NAME

Параметры:

STREAM_NAME — имя потока
Обязательный параметр

YEAR — год
Обязательный параметр

MONTH — месяц
Обязательный параметр

DAY — день
Обязательный параметр

Ответ: JSON вида

```
[ //Список блоков
{
  "timestamp": 1396864800, //Таймстемп блока
```

```

"path":"2014/04/07/10/00", //Путь до записи на жестком диске
"bitrate":2052, //Битрейт
"segments":[
  {
    "second":1, //Начиная с этой секунды от начала блока
    "utc":1396864801, //Таймстемп сегмента
    "duration":5357, //Длина сегмента
    "size":1383868, //Размер в байтах
    "bitrate":2066, //Битрейт
    "jpeg":"2014/04/07/10/00/01.jpg" //Скриншот сегмента
  },

  {"second":6,"utc":1396864806,"duration":7211,
  "size":1842964,"bitrate":2044,
  "jpeg":"2014/04/07/10/00/06.jpg"},

  {"second":13,"utc":1396864813,"duration":7109,
  "size":1739188,"bitrate":1957,
  "jpeg":"2014/04/07/10/00/13.jpg"}
]
}
{"timestamp":1396865400,"path":"2014/04/07/10/10","bitrate":1964,
"segments":[ {"second":0,"utc":1396865400,"duration":7194,
"size":1800664, "bitrate":2002,"jpeg":"2014/04/07/10/10/00.jpg"}]}
]

```

Удаление фрагмента DVR архива

Этот вызов позволяет удалить часть архива принудительно. Если какие-то сегменты защищены от удаления (lock), они удаляются всё равно.

Вызов удаляет фрагменты архива как локального, так и расположенного в облачном хранилище. Если участок архива скопирован в удаленное хранилище (Amazon S3) с помощью `сору=[REMOTE PATH]`, то он скачивается на Flussonic для удаления из него сегментов и затем копируется обратно.

URL: `/flussonic/api/dvr/delete`

Параметры POST запроса:

`stream` — имя потока, часть архива которого необходимо удалить.

`from` — время начала интервала, который необходимо удалить, в UTC.

`duration` — длительность интервала, который необходимо удалить, в секундах.

Пример вызова:

```
curl -u USER:PASS --data '{"stream":"ort","from":1483971680,"duration":1000}' http://FLUSSONIC-IP/flussonic/api/dvr/delete
```

Список файлов VOD (list_files)

Список имеющихся файлов для определенного префикса и источника.

URL:

/flussonic/api/list_files?prefix=VOD_LOCATION&path=VOD_ROOT&subpath=SUB_PATH_IN_VOD_ROOT
&from=FirstName&limit=COUNT

Параметры:

VOD_LOCATION — префикс VOD.

Обязательный параметр

Веб-интерфейс: Список префиксов отображается на главной странице в разделе Files (VOD).

Файл конфигурации: file vod {...};.

VOD_ROOT — корень VOD.

Обязательный параметр

Веб-интерфейс: Список источников (хранилищ) отображается на вкладке соответствующего префикса. Новый источник можно добавить в поле New Path.

Файл конфигурации: элементы path file vod { path priv };

Может быть адресом swift-хранилища и начинаться с swift://.

SUB_PATH_IN_VOD_ROOT — место внутри хранилища VOD_ROOT.

Обязательный параметр

Для корня хранилища его значение /

FirstName — список файлов начиная с этого имени. Можно указать неполное имя файла.

Необязательный параметр

Ответ запроса не будет включать сам файл FirstName.

COUNT — максимальное количество файлов в запросе.

Необязательный параметр

Если используется вместе с FirstName, ответ будет включать COUNT файлов начиная FirstName (не включая сам FirstName).

Ответ: JSON вида

```
{
  "files":[ //Список файлов
    {
      "name":"0.flac", //Название файла
      "type":"file", //Тип (file или directory)
      "prefix":"vod" //Префикс
    },
    {"name":"1.flac","type":"file","prefix":"vod"},
    {"name":"10.flac","type":"file","prefix":"vod"}
  ]
}
```

Сохранить новый файл конфигурации (save_config)

Для обновления конфигурации необходимо передать текст нового файла конфигурации в качестве тела POST запроса.

Важное отличие от update_config в том, что новая конфигурация не только применяется к запущенному серверу, но и заменяет собой существующий файл конфигурации в /etc/flussonic/flussonic.conf.

URL: /flussonic/api/save_config

Параметры:

HTTP request payload — текст нового файла конфигурации.

Обязательный параметр

В этом запросе вы передаете полный текст нового файла конфигурации как данные, приложенные к запросу.

Если использовать curl, то имеется в виду параметр --data-binary:

```
curl ... --data-binary '# Global settings:\nhttp 80;\nrtsp 554;\nrtmp 1935;\npulsedb /var/run/flussonic;'
```

Ответ: true в случае успешной обработки запроса.

Обновить конфигурацию (update_config)

Для обновления конфигурации необходимо передать текст нового файла конфигурации в качестве тела POST запроса.

Важное отличие от save_config в том, что новая конфигурация применяется к запущенному серверу, но сам файл конфигурации на жестком диске не изменяется.

URL: /flussonic/api/update_config

Параметры:

HTTP request payload — текст нового файла конфигурации.

Обязательный параметр

В этом запросе вы передаете полный текст нового файла конфигурации как данные, приложенные к запросу.

Если использовать curl, то имеется в виду параметр --data-binary:

```
curl ... --data-binary '# Global settings:\nhttp 80;\nrtsp 554;\nrtmp 1935;\npulsedb /var/run/flussonic;'
```

Ответ: true в случае успешной обработки запроса.

Удалить поток (config/stream_delete)

Для удаления потока необходимо передать его имя в качестве тела POST запроса.

URL: /flussonic/api/config/stream_delete

Параметры:

HTTP request payload — название потока.

Обязательный параметр

В этом запросе вы передаете название потока как данные, приложенные к запросу.

Если использовать curl, то имеется в виду параметр --data-binary:

```
curl ... --data-binary 'mystream'
```

Ответ: {"success":true} в случае успешной обработки запроса.

Создать или обновить поток (config/stream_create)

Для обновления конфигурации потока необходимо передать текст с настройками потока в качестве тела POST запроса. На вход передается в точности тот же текст, который пишется в файле конфигурации для создания потока. Важно: в данный момент не существует отдельной команды config/stream_update, поэтому уже созданный поток можно обновлять с помощью

запроса `config/stream_create`.

URL: `/flussonic/api/config/stream_create`

Параметры:

HTTP request payload — текст с конфигурацией потока.

Обязательный параметр

В этом запросе вы передаете конфигурацию потока как данные, приложенные к запросу.

Если использовать curl, то имеется в виду параметр `--data-binary`:

`curl ... --data-binary 'stream mystream { url hls://myvideo.com/mystream; dvr /storage 1d 1G; }'`

Ответ: `{"success":true}` в случае успешной обработки запроса.

Перечитать конфигурационный файл (reload)

Этот запрос необходимо выполнить для применения новой конфигурации к работающему серверу, если были внесены изменения в конфигурационный файл на диске `/etc/flussonic/flussonic.conf`.

URL: `/flussonic/api/reload`

Параметры: нет

Ответ: `true` в случае успешной обработки запроса.

Пример использования curl: `curl -u admin:pass0 http://flussonic:8080/flussonic/api/reload`

Перезапустить поток (stream_restart)

Полностью перезапускается один поток. Может помочь в случае проблем с источником.

URL: `/flussonic/api/stream_restart/STREAM_NAME`

Параметры:

STREAM_NAME — идентификатор потока.

Обязательный параметр

Ответ: `true` в случае успешной обработки запроса.

Переключить источник у потока (stream_switch_source)

Поток переключается на использование источника, заданного параметром.

URL: `/flussonic/api/stream_switch_source/STREAM_NAME?url=SOURCE_URL`

Параметры:

STREAM_NAME — идентификатор потока.

Обязательный параметр

SOURCE_URL — URL источника, на который надо переключиться.

Должен быть указан в конфигурации потока.

Обязательный параметр

Включить запись архива (dvr_enable)

В настройках потока обязательно должна быть опция `dvr` или `dvr_offline`.

URL: /flussonic/api/dvr_enable/STREAM_NAME

Параметры:

STREAM_NAME — идентификатор потока.

Обязательный параметр

Ответ: true в случае успешной обработки запроса.

Выключить запись архива (dvr_disable)

URL: /flussonic/api/dvr_disable/STREAM_NAME

Параметры:

STREAM_NAME — идентификатор потока.

Обязательный параметр

Ответ: true в случае успешной обработки запроса.

Серверы в кластере (cluster_servers)

URL: /flussonic/api/cluster_servers

Параметры: нет

Ответ: JSON вида

```
{
  "streamer-2": //Имя сервера в кластере
  {
    "client_count":0, //Текущее количество клиентов.
    "config2":true,
    "cpu_usage":1, //Использование процессора в процентах.
    "fetch_delay":0,
    "id":"91a25b1a-22f0-414b-858f-4abc727d0de2",
    "is_peer":true,
    "is_source":false,
    "key":"WzkF7f9VYndoH6G3VOVz6iSNIOh4h8CF", //Ключ кластера.
    "memory_usage":26, //Использование оперативной памяти в процентах.
    "output_bitrate":0, //Суммарный исходящий битрейт.
    "persistent":true,
    "ports":{ //Список используемых портов.
      "http":80,
      "rtmp":1935
    },
    "stream_count":0, //Текущее количество потоков.
    "uptime":251908, //Время с последнего перезапуска, в секундах.
    "version":"4.7.3" //Версия Flussonic Media Server.
  }
}
```

Events API

События Flussonic

В Flussonic Media Server есть удобная и гибкая система внутренних событий, с маршрутизацией и обработчиками, и возможностями настройки.

События инициируются в разных частях системы и могут использоваться для разных сценариев.

Чтобы задать настройки, связанные с событиями, в файл конфигурации Flussonic добавьте директиву `notify` и опцию `sink` (получатель события):

Чтобы использовать кастомный обработчик, в `sink` нужно указать путь до него.

Чтобы записать событие в журнал событий (лог), в `sink` указывают путь к файлу журнала.

Далее укажите различные опции, чтобы отфильтровать события до того, как они попадут в лог или в обработчик.

Содержание:

- Настройка логирования событий

- Настройка обработчиков событий

- Фильтрация событий

- Список доступных событий

- Пример настройки email нотификаций

- События производительности

- Гарантированная доставка уведомлений

Настройка логирования событий

В дополнение к основному журналу, Flussonic позволяет создавать столько лог-файлов, сколько может потребоваться, и записывать в них события, выбранные по разным критериям. Все это вы можете указать в настройках.

Чтобы вести свой файл с журналом событий, добавьте директиву `notify` и укажите путь к файлу журнала в опции `sink log://`:

```
notify log_name {  
    sink log://log/crash.log;  
    verbose debug;  
}
```

Здесь:

`log_name` — просто название настройки. Для удобства можно давать содержательные названия.

`sink` — получатель событий. В случае логов это файл, в который нужно записывать информацию о событиях.

`verbose` — уровень логирования по степени важности событий. `debug` (самое подробное логирование), `info`, или `alert` (логирование только важных событий).

Исключение событий из лога

Чтобы не записывать определенные события в данный файл, перечислите их в опции `except`. Например, следующая конфигурация игнорирует события, касающиеся потоков (но пишет другие события, такие как события сервера Flussonic):

```
notify log_name {
    sink log://log/crash.log;
    except media=*;
    verbose debug;
}
```

Настройка обработчиков событий

Каждый обработчик должен быть объявлен в конфигурации:

```
notify handler_name {
    sink http://backend.local/notify.php;
}
```

Такое объявление создаст обработчик событий с именем `handler_name` и он будет отсылать ВСЕ события на HTTP URL `http://backend.local/notify.php`.

В этой конфигурации все события Flussonic Media Server будут отправляться в JSON формате как список объектов.

В высоконагруженной системе может генерироваться огромное количество событий, большая часть которых не требуется.

Мы можем уменьшить поток событий улучшив конфигурацию:

```
notify handler_name {
    sink http://backend.local/notify.php;
    only event=stream_started,stream_stopped,source_ready,source_lost;
}
```

С такой конфигурацией на этот обработчик будут отправляться только определённые события.

Вызовы обработчика происходят синхронно: событие не будет отправлено в обработчик, пока не он не завершит обработку предыдущей порции событий.

В конфигурации обработчика событий можно указать следующие опции:

sink Получатель событий. Может быть задан HTTP URL: `http://URL`, `https://URL` или путь к LUA скрипту: `path_to_lua_script.lua`

only Белый список ограничений. Можно указать несколько `key=value` или `key=value1,value2` опций для каждой строки `only`. Вы можете фильтровать события по их типу `event`, по `media` или любому другому полю, такому как `country` или `ip`. Обычно это `event` и `media`. Ниже есть более полное описание поведения `only`.

except Черный список ограничений. События, совпавшие с одним из полей в `except` не будут переданы обработчику.

buffer Можно установить в `false` для имитации поведения до версии 4.6.14. Лучше не использовать этот параметр.

Все остальные опции в этом блоке будут переданы указанному приёмнику событий. В при LUA скрипте они доступны в таблице args. По HTTP они передаются вместе с другими параметрами.

Некоторые дополнительные опции:

sign_key Вы можете указать ключ подписи для HTTP приёмника событий. Когда Flussonic будет готовить HTTP POST запрос с данными в формате JSON, он добавит этот ключ к телу запроса, вычислит SHA1 хэш и добавит его в hex виде в http header ``X-Signature``. Это может использоваться для проверки, что событие отправил именно Flussonic.

Фильтрация событий

Вы можете фильтровать события перед тем, как они попадают в обработчик. Это очень важный механизм, попробуйте его использовать, потому что он уменьшает нагрузку на ваш обработчик событий. Каждое событие проходит через фильтр непосредственно в тред, создавшем это событие, перед тем, как попасть в обработчик.

Правила фильтрации:

если любая ехсерт директива полностью совпала с событием, событие выкидывается и не передаётся в обработчик;

если в объявлении обработчика нет only директив, события передаются в обработчик;

если директивы only есть, то событие передается в обработчик, если оно полностью совпало хотя бы с одной директивой only.

Полное совпадение директивы с событием означает, что все пары ключ=значение в директиве равны значениям в событии. Если в директиве указано ключ=значение1,значение2,значение3, то это означает, что поле ключ в событии должно быть равным какому-то одному значению из заданного списка.

Примеры:

only event=stream_started; совпадает с {event: "stream_started", media: "cbc"}

only event=stream_started,stream_stopped; совпадает с {event: "stream_started", media: "cbc"}

only event=stream_started,stream_stopped media=tnt; НЕ совпадает с {event: "stream_started", media: "cbc"}

only event=stream_started media=cbc group=news; НЕ совпадает с {event: "stream_started", media: "cbc"}

Список доступных событий

server_started Посылается после старта сервера

listener_start Flussonic открыл какой-то порт на приём подключений

listener_failure Ошибка открытия порта для приёма подключений

config_reloaded Конфиг был перечитан

session_opened Открытие сессии

session_closed Закрытие сессии

file_opened Открытие файла

file_closed Закрытие файла

stream_started Запуск потока

stream_stop Поток получил команду на закрытие через API

stream_stopped Остановка потока

stream_reconfigured Обновление конфигурации потока

stream_motion_started На IP камере было отмечено движение

stream_motion_stopped На IP камере было окончено движение

source_ready Поток принял первые кадры

stream_media_info Свойства потока (media_info) изменились

source_lost Источник потока потерян и требует перезапуска

source_switch Поток переключился на другой источник

frames_timed_out Источник потока перестал слать кадры (но ещё не перезапущен)

frames_restored Источник потока возобновил отсылку кадров после перерыва

stream_backup Начал проигрываться запасной файл в связи с потерей источника

publish_started Началась публикация в поток

publish_stopped Публикация в поток окончилась (вы можете получить много значимой информации с этим событием)

push_started Поток начал отправку (push) на другой источник

stream_jpeg Новый JPEG скриншот был сгенерирован

dvr_new_fragment Новый DVR фрагмент был записан на диск

dvr_deleted_fragments Старые фрагменты были удалены из архива

dvr_new_blob Новый часовой отрезок был открыт для записи архива

stream_force_close_gop Ошибка потока: получена неправильная временная отметка или слишком малое количество кадров в секунду

stream_rt_sync В потоке произошла пересинхронизация аудио и видео. Может быть признаком ошибок в потоке, если происходит слишком часто.

stream_broken_source Поток не может быть прочитан из текущего источника и было решено перезапустить поток dvr_replication_started Началась репликация архива

dvr_hour_replication_started Началась репликация часового отрезка архива

dvr_hour_replication_done Репликация часового отрезка архива закончилась

dvr_replication_progress Отображение процесса репликации

dvr_replication_done Процесс репликации закончен udp_pusher_does_not_fit_cbr Поток, отправленный по UDP с постоянным битрейтом (через настройку потока вида push udp://IP_ADDRESS?cbr=...), имеет слишком высокий битрейт, что может привести к переполнению буфера. В результате может возникнуть ошибка или рестарт потока. Чтобы такого не произошло, вы можете перекодировать поток в меньший битрейт с помощью транскодера.

Пример настройки email нотификаций

Давайте узнаем, что вы можете делать с помощью системы событий. Например, вы хотите получать письма, если поток остановился.

Простейшая конфигурация будет выглядеть так:

```
notify no_video {
    only event=stream_stopped,source_lost;
    sink /etc/flussonic/no_video.lua;
    from flussonic@streamer1.my.cdn;
    to admin@my.cdn;
    via smtp://127.0.0.1:587;
}
```

Этого конфига достаточно, если вы не хотите фильтровать события определённым по потокам.

Скрипт no_video.lua:

```
body = "Source lost on following streams: \n"
```

```
for _, event in pairs(events) do
    body = body.." "..event.media.." \n"
end
```

```
mail.send({from = args.from, to = args.to, subject = "Source lost", body = body})
```

Для корректной отправки почты необходимо установить утилиту sendmail:

```
apt-get install sendmail
```

Дополнительно необходимо убедиться, что sendmail слушает порт, указанный в файле конфигурации:

```
netstat -lntp
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	127.0.0.1:587	0.0.0.0:*	LISTEN	3507/sendmail

В качестве hostname сервера необходимо указывать РЕАЛЬНОЕ доменное имя:

hostname
streamer1.my.cdn

События производительности

Система событий Flussonic позволяет настроить оповещения об использовании ресурсов и проблемах с производительностью.

События `busy_port`, `busy_dist_port`, `long_gc`, `long_schedule_pid` и `long_schedule_port` соответствуют событиям системного монитора в Erlang. Их подробное описание, условия появления и передаваемые в событии параметры находятся в документации Erlang. Эти события говорят о возможном наличии проблем с производительностью.

У Flussonic все эти события принадлежат к группе `system_overloaded`. Группу событий следует указывать, используя ключевое слово `group`.

Пример настроек для получения событий о производительности:

```
notify performance_handler {  
    sink http://backend.local/notify.php;  
    only group=system_overloaded;  
}
```

В этом примере фильтр `group=system_overloaded` пропускает в обработчик только события о проблемах производительности.

Использование памяти

Событие `memory_usage` возникает при повышенном использовании памяти. В событии передаются параметры `total` и `used` — это общий и используемый размер памяти в байтах. Flussonic отсылает событие `memory_usage`, когда использует более половины всей памяти. При превышении порога в 80%, событие получает дополнительно признак `system_overloaded` и отсылается вместе с событиями группы `system_overloaded`.

Гарантированная доставка уведомлений

Чтобы не терять неотправленные уведомления, можно настроить Flussonic на отложенную повторную отправку. Если принимающий HTTP сервер или скрипт не отвечает, Flussonic накапливает события в специальном буфере и периодически повторяет попытки отослать нотификации. Когда принимающий сервер начнёт отвечать, Flussonic дошлёт накопленные нотификации.

Для этого вы должны указать в конфиге две опции:

```
notify watcher {  
    sink http://backend.local/notify.php;  
    resend_notifications_limit 1000;  
    resend_notifications_timeout 10;  
}
```

где:

`resend_notifications_limit` — количество последних событий, которое будет храниться с целью повторных попыток отослать их. Не может превышать 2000.

`resend_notifications_timeout` — интервал времени в секундах, через который Flussonic будет пытаться отослать события.

SQL API

К Flussonic Media Server можно подключиться по протоколу MySQL и запрашивать данные как у MySQL сервера.

Устанавливать MySQL не нужно. Достаточно включить поддержку MySQL API в файл конфигурации, указав порт, который Flussonic Media Server будет слушать.

Для этого надо в конфигурационном файле (или в веб-интерфейсе) включить порт для MySQL:

```
mysql 3306;
```

Если не указать эту строку, то по умолчанию будет использоваться порт 14406.

После этого с логином и паролем администратора (edit_auth или view_auth) можно подключиться с помощью клиента mysql или любой библиотеки.

База данных — flussonic:

```
mysql -u admin -h 127.0.0.1 -p flussonic
```

```
mysql> show tables;
```

```
+-----+
| Tables_in_flussonic |
+-----+
| streams              |
| files                |
| sessions             |
| stats                |
+-----+
4 rows in set (0.00 sec)
```

```
mysql> select * from streams;
```

```
+-----+-----+-----+-----+-----+-----+-----+
+-----+
| name | url | ts_delay | bytes_in | bytes_out | client_count | lifetime | retry_count | bitrate |
+-----+-----+-----+-----+-----+-----+-----+
+-----+
| channel | tshttp://transcoder:9000/ | 59 | 1003280707 | 0 | 0 | 46056397 | 0 | 647 |
+-----+-----+-----+-----+-----+-----+
+-----+
1 row in set (0.00 sec)
```

Аналогично есть доступ к таблице files.

Доступные таблицы

streams информация об активных потоках

files информация об открытых файлах

sessions информация об открытых сессиях

sessions_history информация об истории сессий

stats статистика по всевозможным метрикам

dvr_status статистика записи потока за указанное время. Обязательно указывать имя потока и время.

Советы

dvr_status

В запросе к dvr_status нужно обязательно использовать имя стрима (name) и время (utc).

Если их не указать, то запрос всегда будет возвращать ошибку «ERROR 1210 (HY000): name and utc conditions are required».

При этом значение name нужно указывать в одинарных кавычках, а utc без кавычек вообще.

Например: `select * from dvr_status where name='mystream' and utc > 1411084801;`

Если поставить неверные кавычки, то запрос будет всегда возвращать "Query OK, 0 rows affected (0.00 sec)".

Эта таблица предназначена для работы не с точным временем (utc=1411084801), а с диапазонами (utc > 1411084801).

В данный момент статистика хранится около суток. Это время нельзя увеличить.

Возможно, в будущих версиях время будет реализована возможность длительного хранения статистики, но сейчас этой возможности нет.

Отношение к MySQL database server

Часто задаваемый вопрос: Должен ли я для работы вашего MySQL API установить Oracle MySQL Database Server? Или необходимо будет вручную обходить конфликты с ней? Или нужно просто добавить поддержку MySQL API в файл конфигурации?

Устанавливать MySQL сервер не нужно. Просто добавьте поддержку MySQL API в файл конфигурации, указав порт, который Flussonic Media Server будет слушать.

Не забудьте использовать свободный порт, не занятый другой базой данных (3306 — это порт по умолчанию для MySQL, так что лучше использовать что-нибудь другое).

Flussonic Media Server не использует реального сервера баз данных, это просто эмуляция внутри Flussonic, позволяющая отвечать на простые SQL-запросы.

Так как это не полноценный сервер баз данных, для просмотра таблиц не получится использовать программы типа PHPMyAdmin. Данные нужно получать, выполняя простые SQL-запросы внутри MySQL клиента с помощью соответствующих библиотек.

Управление кластерами при помощи SQL API

Flussonic Media Server позволяет управлять потоками (создавать, обновлять, удалять и выбирать) по всему кластеру серверов через одну точку входа.

В этой статье мы подробно расскажем, как это сделать.

Не нужно устанавливать MySQL сервер. Flussonic Media Server не работает с MySQL сервером, но он ведет себя так, как будто он и является MySQL сервером.

Когда вы укажете `mysql 3306`; в конфигурационном файле `/etc/flussonic/flussonic.conf`, Flussonic Media Server привяжется к этому порту и будет работать в качестве сервера MySQL (с определенными функциональными возможностями). Не устанавливайте MySQL сервер!

Обычно клиент MySQL пытается подключиться к серверу MySQL через сокет домена Unix. Когда вы скажете ему подключиться к `localhost`, он попытается открыть что-то вроде `/tmp/mysql.sock` или что-то еще. Flussonic слушает строго сокет TCP, поэтому при попытке сделать это на `localhost`, вам следует использовать `127.0.0.1`. Как правило, это алиас, но в случае с клиентом MySQL существует разница.

Ключ кластера

Серверы в кластере используют `cluster_key` для авторизации друг у друга. Кластерный ключ это произвольная последовательность букв и цифр, которую необходимо придумать или сгенерировать. Он хранится в глобальных опциях сервера.

Чтобы настроить ключ кластера на текущем Flussonic Media Server, необходимо заявить `cluster_key` одним из способов:

В конфигурационном файле в глобальных опциях сервера.

Пример:

```
cluster_key wjFxFxSiSG6EC6e2;
```

В веб-интерфейсе Flussonic в `Cluster > Settings`.

Серверы в кластере

Расскажем о концепции сервера в кластере. Каждый Flussonic Media Server знает о себе, а о наличии других серверов в кластере Flussonic узнает из директивы `peer`. Серверы в кластере используют `cluster_key` для авторизации друг у друга.

Добавив `cluster_key` в директиву `peer`, вы сообщаете Flussonic Media Server, какой `cluster_key` следует использовать для подключения к другому Flussonic серверу. По умолчанию, для соединения с пирами Flussonic Media Server будет использовать свой собственный ключ кластера.

Пример

Если у вас есть два сервера: srv01.cdn.local и srv02.cdn.local, то их можно настроить следующим образом:

srv01.cdn.local:

```
http 80;
mysql 3306;
edit_auth admin secretpass;
cluster_key mcXpNJyZX3mSE3;
peer srv01.cdn.local;
peer srv02.cdn.local {
  cluster_key wjFxFxSiSG6EC6e2;
}
```

srv02.cdn.local:

```
http 80;
cluster_key wjFxFxSiSG6EC6e2;
```

Когда вы настроите их таким образом, srv01 будет периодически пинговать srv02 и извлекать информацию о состоянии потоков на нем.

Теперь давайте посмотрим, как получить такую же конфигурацию при помощи SQL API:

```
$ mysql -u admin -psecretpass -h srv01.cdn.local cluster
mysql> INSERT INTO peers (hostname,cluster_key) VALUES ('srv02.cdn.local','wjFxFxSiSG6EC6e2');
Query OK, 1 row affected (0.36 sec)
```

```
mysql> SELECT * FROM peers;
```

hostname	available	id	port	cluster_key	uptime	load
srv02.cdn.local	1	bcac6edd-46a7-40cc-970a-5da3bec8d94b	80	wjFxFxSiSG6EC6e2	78715	NULL
localhost	1	cd30c5a4-3dda-4ac1-a2eb-a9b765918082	80	mcXpNJyZX3mSE3	NULL	0

Теперь вы получите такую же конфигурацию, как описано выше. Если вы видите идентификатор, это означает, что данный Flussonic смог подключиться к удаленному Flussonic и установить пиринг. Вы сможете видеть время работы и даже нагрузку (использование выходной полосы пропускания в процентах).

Удалить этот сервер из конфигурации можно так:

```
mysql> DELETE FROM peers WHERE hostname='srv02.cdn.local';
Query OK, 1 row affected (0.68 sec)
```

Теперь этот сервер удален и вы не будете видеть потоки с данного сервера.

Управление потоками

Вы можете получить доступ ко всем потокам на всех серверах из одного места через обычный SQL API.

Flussonic дает возможность хранить некоторые метаданные потока прямо в разделе

конфигурации потока в файле flussonic.conf, поэтому вам, возможно, не понадобится вести таблицу streams в системе управления.

Поток идентифицируется по столбцу name и имеет столбец server с именем хоста того сервера, на котором "находится" поток.

Если вы создадите поток без указания сервера, он будет создан на том сервере, к которому вы подключены. Если же вы укажете сервер, поток будет создан на указанном сервере.

Не нужно указывать сервер при обновлении, удалении или выборе потока, потому что Flussonic Media Server автоматически найдет сервер, на котором выполняется поток, и выполнит на этом сервере необходимые действия.

Структура таблицы Peers

Таблица peers (сервера в кластере) содержит следующие поля:

Поле	Тип	Описание
------	-----	----------

hostname	string	
----------	--------	--

available	bool	
-----------	------	--

id	string	
----	--------	--

port	integer	
------	---------	--

https_port	integer	
------------	---------	--

rtmp_port	integer	
-----------	---------	--

rtsp_port	integer	
-----------	---------	--

cluster_key	string	
-------------	--------	--

uptime integer

meta string

bandwidth_usage integer исходящий трафик в процентах от total_bandwidth

cpu_usage integer использование процессора в процентах

memory_usage integer использование памяти в процентах от RAM

total_clients integer

online_streams integer

total_streams integer

version string

build integer

is_remote bool

is_peer bool

is_source bool

error string

fetch_at integer

vsaas bool служебное

vsaas_endpoint string служебное

rproxy bool служебное

rproxy_streampoint_key string служебное

rproxy_endpoint_auth string служебное

camera_alarm integer

camera_alarm_address string

camera_alarm_patterns string

Структура таблицы Streams

Таблица streams широкая, потому что каждый поток содержит много информации.

Поле Тип Описание

name string имя потока

title string mpegts title

provider string mpegts title

static bool static / ondemand

pulse_off bool не учитывать поток в статистике потребления ресурсов

position integer положение (для vod файлов)

disabled bool

retry_limit integer

source_timeout integer

password string

url_publish bool

prefix string http://yourwebsite.com/...

max_bitrate integer

on_publish string

motion_detector_enabled bool

motion_detector_notify string

motion_detector_tags string

transcoder string

dvr_root string путь к архиву на сервере или ссылка на RAID массив

hls_off bool включение / выключение протокола

hds_off bool включение / выключение протокола

rtmp_off bool включение / выключение протокола

rtsp_off bool включение / выключение протокола

dash_off bool включение / выключение протокола

mpegts_off bool включение / выключение протокола

webrtc_off bool включение / выключение протокола

m4s_off bool включение / выключение протокола

m4f_off bool включение / выключение протокола

mseld_off bool включение / выключение протокола

url_prefix string

segment_count integer

segment_duration integer

prepush integer

backup string

add_audio_only bool

max_sessions integer

thumbnails_enabled bool

thumbnails_url string адрес скриншотов (jpeg)

clients_timeout integer

push string

external_cache string служебное

auth_url string

domains string

allowed_countries string

disallowed_countries string

glue_ts bool

vsaas bool служебное

iptv bool служебное

comment string

coordinates string служебное

postal_address string служебное

owner string служебное

auth_token string служебное

mobile_token string служебное

access string служебное

onvif_url string служебное

onvif_profile string служебное

can_ptz bool служебное

dvr_protected bool служебное

agent_id string служебное

agent_pin string служебное

agent_key string служебное

agent_model string служебное

agent_serial string служебное

registered_at integer служебное

program_id integer служебное

stream_id integer служебное

extra string служебное

dvr string

dvr_no_index bool не создавать idx index

urls string

cluster_ingest bool

alive bool

remote bool

source_hostname string

lifetime integer служебное

start_running_at integer

ts_delay integer служебное

retry_count integer

client_count integer

last_dts integer служебное

last_dts_at integer служебное

last_access_at integer

input_error_rate integer служебное

bytes_in integer

bytes_out integer

out_bandwidth integer

bufferings integer служебное

bitrate integer

source_error string ошибка источника

url string

current_agent_id string служебное

agent_status string служебное

server string

published_from string IP адрес источника публикации

published_via string формат публикации

dvr_only bool служебное

running_transcoder bool

dvr_replication integer

dvr_replication_running bool

Здесь находятся не все параметры конфигурации. Мы работаем над заполнением таблицы.

Создание потока

При первом запуске свежего кластера у вас не будет никаких потоков:

```
mysql> select * from streams;  
Empty set (0.00 sec)
```

Здесь и далее мы предполагаем, что на всех серверах вы сделали следующие настройки:

```
file vod {  
    path priv;  
}
```

и файл bunny.mp4 находится в пути priv.

Создадим потоки:

```
mysql> insert into streams (name,urls,dvr) values ('bunny1','file://vod/bunny.mp4','movies 2d 20G');  
Query OK, 1 row affected (0.02 sec)
```

```
mysql> select name,static,urls,lifetime,server from streams;
```

name	static	urls	lifetime	server
bunny1	1	file://vod/bunny.mp4	27029	localhost

```
1 row in set (0.01 sec)
```

Теперь заставим работать кластер:

```
mysql> insert into streams (name,urls,dvr,server) values ('bunny2','file://vod/bunny.mp4','movies 2d 20G','srv02.cdn.local');
```

Query OK, 1 row affected (0.03 sec)

```
mysql> select name,static,urls,lifetime,server from streams;
```

name	static	urls	lifetime	server
bunny1	1	file://vod/bunny.mp4	108025	localhost
bunny2	1	file://vod/bunny.mp4	3989	srv02.cdn.local

2 rows in set (0.00 sec)

Таким образом, мы можем создавать потоки на всех наших серверах из единой точки входа. При этом нужно явно указать на каком сервере хотим создать поток.

Обновление потока

При обновлении информации о потоке не нужно явно указывать сервер. Это очень удобно, поскольку не нужно помнить, где находится ваш поток.

```
mysql> update streams set url_prefix='http://srv02.cdn.local' where name='bunny2';
```

Query OK, 1 row affected (0.02 sec)

```
mysql> select name,static,urls,lifetime,server,url_prefix from streams;
```

name	static	urls	lifetime	server	url_prefix
bunny1	1	file://vod/bunny.mp4	429009	localhost	NULL
bunny2	1	file://vod/bunny.mp4	319980	srv02.cdn.local	http://srv02.cdn.local

2 rows in set (0.01 sec)

Flussonic Media Server "знает", на каком сервере работает поток, поэтому он не потребует от вас указывать его.

Однако в пределах кластера поле name не уникально. Поэтому если вы дублировали имена потоков на нескольких серверах, то тогда нужно указать сервер, на котором вы собираетесь обновить поток.

Удаление потока

Давайте избавимся от одного из этих потоков:

```
mysql> delete from streams where name='bunny1';
```

Query OK, 1 row affected (0.02 sec)

```
mysql> select name,static,urls,lifetime,server,url_prefix from streams;
```

name	static	urls	lifetime	server	url_prefix
------	--------	------	----------	--------	------------


```
+-----+-----+-----+-----+-----+-----+
| bunny2 | 1 | file://vod/bunny.mp4 | 593231 | srv02.cdn.local | http://srv02.cdn.local |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

и удалим все остальные потоки:

```
mysql> delete from streams where name='bunny2';
Query OK, 1 row affected (0.02 sec)
```

```
mysql> select name,static,urls,lifetime,server,url_prefix from streams;
Empty set (0.00 sec)
```

При удалении потока Flussonic Media Server следует тому же правилу, что и для обновления или выбора. Если у вас в кластере один поток, он обнаружит его и изменит файл конфигурации этого удаленного сервера.

Миграция потоков между серверами

Вот тут и начинается волшебство. Если вы настроили поток на одном сервере, для него достаточно просто указать другой сервер, чтобы автоматически произошла миграция:

```
mysql> insert into streams (name,urls,dvr,server) values ('bunny2','file://vod/bunny.mp4','movies 2d 20G','s
rv02.cdn.local');
Query OK, 1 row affected (0.03 sec)
```

```
mysql> select name,static,urls,lifetime,server from streams;
+-----+-----+-----+-----+-----+-----+
| name | static | urls          | lifetime | server      |
+-----+-----+-----+-----+-----+-----+
| bunny2 | 1 | file://vod/bunny.mp4 | 3889 | srv02.cdn.local |
+-----+-----+-----+-----+-----+-----+
1 rows in set (0.00 sec)
```

```
mysql> update streams set server='srv03.cdn.local' where name='bunny2';
Query OK, 1 row affected (0.02 sec)
```

```
mysql> select name,static,urls,lifetime,server from streams;
+-----+-----+-----+-----+-----+-----+
| name | static | urls          | lifetime | server      |
+-----+-----+-----+-----+-----+-----+
| bunny2 | 1 | file://vod/bunny.mp4 | 3989 | srv03.cdn.local |
+-----+-----+-----+-----+-----+-----+
1 rows in set (0.00 sec)
```

Перемещать потоки можно с любого сервера на любой другой. Flussonic Media Server будет создавать их на новых серверах и удалять старые.

Таблица GPU

Таблица gpu содержит информацию о текущей загрузке GPU.

Примеры запросов:

```
SELECT * FROM gpu
```

```
MySQL [cluster]> select * from gpu;
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| server | hwid | power | temperature | memory | encoder | decoder | sm |
+-----+-----+-----+-----+-----+-----+-----+-----+
| encoder | nv1 | 8 | 37 | 0 | 0 | 0 | 0 |
| encoder | nv0 | 20 | 41 | 0 | 0 | 0 | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
SELECT * FROM gpu ORDER BY sm DESC - отсортировать gpu по загрузке
```

```
mysql> select * from gpu order by sm desc;
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| server | hwid | power | temperature | memory | encoder | decoder | sm |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 4gpu | nv2 | 46 | 36 | 2 | 11 | 7 | 7 |
| 4gpu | nv3 | 45 | 32 | 2 | 10 | 7 | 6 |
| 4gpu | nv0 | 43 | 38 | 2 | 7 | 3 | 4 |
| 4gpu | nv1 | 44 | 34 | 1 | 2 | 3 | 3 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Описание полей:

server - имя (hostname) сервера

hwid - ID карты, оно же используется в настройках транскодера(deviceid=1)

power - текущее потребление карты в ваттах

temperature - температура карты в градусах

memory - утилизация памяти в %

encoder - утилизация энкодера в %

decoder - утилизация декодера в %

sm - загрузка CUDA в %.

SNMP

Во Flussonic Media Server есть базовая реализация SNMP. Она позволяет отслеживать разные параметры, например, сколько ресурсов потребляют видео-потoki Flussonic.

Она включается в конфигурационном файле Flussonic следующим способом:

```
snmp 4000;
```

```
edit_auth ADMIN PASSWORD;
```

Примените настройки:

```
/etc/init.d/flussonic reload
```

Теперь на порту 4000 Flussonic будет отвечать по SNMP.

snmpwalk

Чтобы прочитать данные SNMP, выполните такие команды:

```
apt-get -y install snmp snmp-mibs-downloader
```

```
cd /opt/flussonic
```

```
snmpwalk -c ADMIN -v 2c -M +lib/snmp2/mibs/ -m +FLUSSONIC-MIB 127.0.0.1:4000 .
```

Замените ADMIN на логин администратора Flussonic.

Здесь snmpwalk — утилита для диагностики установленной системы SNMP.

Опция -c ADMIN означает "community" в терминах SNMP. SNMP community равен логину администратора Flussonic.

Пример

Если все настроено правильно, то утилита snmpwalk даст ответ такого вида:

```
# snmpwalk -c flussonic -v 2c -M +lib/snmp2/mibs/ -m +FLUSSONIC-MIB 127.0.0.1:4000 .
```

```
Created directory: /var/lib/snmp/mib_indexes
```

```
SNMPv2-MIB::sysDescr.0 = STRING: Flussonic 19.05
```

```
SNMPv2-MIB::sysObjectID.0 = OID: FLUSSONIC-MIB::flussonicModule
```

```
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (81590) 0:13:35.90
```

```
SNMPv2-MIB::sysContact.0 = STRING: info@erlyvideo.org
```

```
SNMPv2-MIB::sysName.0 = STRING: Flussonic
```

```
SNMPv2-MIB::sysLocation.0 = STRING: Erlang
```

```
SNMPv2-MIB::sysServices.0 = INTEGER: 72
```

```
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (0) 0:00:00.00
```

```
SNMPv2-MIB::snmplnPks.0 = Counter32: 1186
```

```
SNMPv2-MIB::snmplnBadVersions.0 = Counter32: 0
```

```
SNMPv2-MIB::snmplnBadCommunityNames.0 = Counter32: 6
```

```
SNMPv2-MIB::snmplnBadCommunityUses.0 = Counter32: 0
```

```
SNMPv2-MIB::snmplnASNParseErrs.0 = Counter32: 0
```

```
SNMPv2-MIB::snmpEnableAuthenTraps.0 = INTEGER: enabled(1)
```

```

SNMPv2-MIB::snmpSilentDrops.0 = Counter32: 0
SNMPv2-MIB::snmpProxyDrops.0 = Counter32: 0
FLUSSONIC-MIB::streamsNum.0 = Gauge32: 2
FLUSSONIC-MIB::sIndex.1 = INTEGER: 1
FLUSSONIC-MIB::sIndex.2 = INTEGER: 2
FLUSSONIC-MIB::sName.1 = STRING: mc1
FLUSSONIC-MIB::sName.2 = STRING: tra
FLUSSONIC-MIB::sClientCount.1 = Gauge32: 1
FLUSSONIC-MIB::sClientCount.2 = Gauge32: 0
FLUSSONIC-MIB::sRetryCount.1 = Gauge32: 0
FLUSSONIC-MIB::sRetryCount.2 = Gauge32: 0
FLUSSONIC-MIB::sLifeTime.1 = Counter64: 809863
FLUSSONIC-MIB::sLifeTime.2 = Counter64: 804257
FLUSSONIC-MIB::sBitrate.1 = Counter64: 8657
FLUSSONIC-MIB::sBitrate.2 = Counter64: 4178
FLUSSONIC-MIB::sBytesIn.1 = Counter64: 900131893
FLUSSONIC-MIB::sBytesIn.2 = Counter64: 453771011
FLUSSONIC-MIB::sBytesOut.1 = Counter64: 0
FLUSSONIC-MIB::sBytesOut.2 = Counter64: 0
FLUSSONIC-MIB::sStatus.1 = INTEGER: active(1)
FLUSSONIC-MIB::sStatus.2 = INTEGER: active(1)
FLUSSONIC-MIB::totalClients.0 = Gauge32: 1
FLUSSONIC-MIB::schedulerLoad.0 = Gauge32: 0
SNMPv2-MIB::snmpSetSerialNo.0 = INTEGER: 362871239
SNMP-FRAMEWORK-MIB::snmpEngineID.0 = STRING: "flussonicID"
SNMP-FRAMEWORK-MIB::snmpEngineBoots.0 = INTEGER: 1
SNMP-FRAMEWORK-MIB::snmpEngineTime.0 = INTEGER: 816 seconds
SNMP-FRAMEWORK-MIB::snmpEngineMaxMessageSize.0 = INTEGER: 484
SNMP-MPD-MIB::snmpUnknownSecurityModels.0 = Counter32: 0
SNMP-MPD-MIB::snmpInvalidMsgs.0 = Counter32: 0
SNMP-MPD-MIB::snmpUnknownPDUHandlers.0 = Counter32: 0
SNMP-MPD-
MIB::snmpUnknownPDUHandlers.0 = No more variables left in this MIB View (It is past the end of the MIB tree)

```

В данной SNMP таблице есть объекты, которые относятся к Flussonic (FLUSSONIC-MIB) и содержат данные о потоках, такие как имя потока (FLUSSONIC-MIB::sName), количество запросивших клиентов (FLUSSONIC-MIB::sClientCount), время активности потока (FLUSSONIC-MIB::sLifeTime) и т.д.

Потоки пронумерованы .1, .2, и т.д.

Поясним объекты, которые могут вызвать вопросы:

FLUSSONIC-MIB::sStatus

Возвращает целые числа, соответствующие следующим значениям:

active = 1

```
notInService = 2
notReady = 3
active = 1
notInService = 2
notReady = 3
FLUSSONIC-MIB::schedulerLoad
```

Потребление в % ресурса Erlang scheduler (среднее за последнюю минуту). Соответствует среднему значению из Pulse > Scheduler utilization for last minute.

snmptranslate

Чтобы получить информацию об объектах и идентификаторах (OID), воспользуйтесь утилитой snmptranslate с флагом -Tz:

```
snmptranslate -m lib/snmp2/mibs/FLUSSONIC-MIB.mib -Tz
```

Утилита дает ответ такого вида:

```
"org"          "1.3"
"dod"          "1.3.6"
"internet"     "1.3.6.1"
"directory"    "1.3.6.1.1"
"mgmt"         "1.3.6.1.2"
"mib-2"        "1.3.6.1.2.1"
"transmission" "1.3.6.1.2.1.10"
"experimental" "1.3.6.1.3"
"private"      "1.3.6.1.4"
"enterprises"  "1.3.6.1.4.1"
"flussonicModule" "1.3.6.1.4.1.36342"
"flussonic"     "1.3.6.1.4.1.36342.1"
"streams"       "1.3.6.1.4.1.36342.1.1"
"streamsNum"    "1.3.6.1.4.1.36342.1.1.1"
"streamsTable"  "1.3.6.1.4.1.36342.1.1.2"
"streamsEntry"  "1.3.6.1.4.1.36342.1.1.2.1"
"sIndex"        "1.3.6.1.4.1.36342.1.1.2.1.1"
"sName"         "1.3.6.1.4.1.36342.1.1.2.1.2"
"sClientCount"  "1.3.6.1.4.1.36342.1.1.2.1.3"
"sRetryCount"   "1.3.6.1.4.1.36342.1.1.2.1.4"
"sLifeTime"     "1.3.6.1.4.1.36342.1.1.2.1.5"
"sBitrate"      "1.3.6.1.4.1.36342.1.1.2.1.6"
"sBytesIn"      "1.3.6.1.4.1.36342.1.1.2.1.7"
"sBytesOut"     "1.3.6.1.4.1.36342.1.1.2.1.8"
"sStatus"       "1.3.6.1.4.1.36342.1.1.2.1.9"
"accounting"    "1.3.6.1.4.1.36342.1.2"
"totalClients"  "1.3.6.1.4.1.36342.1.2.1"
"serverStatus"  "1.3.6.1.4.1.36342.1.3"
"schedulerLoad" "1.3.6.1.4.1.36342.1.3.1"
"flussonicConformance" "1.3.6.1.4.1.36342.2"
"streamGroup"   "1.3.6.1.4.1.36342.2.1"
"statGroup"     "1.3.6.1.4.1.36342.2.2"
"statusGroup"   "1.3.6.1.4.1.36342.2.3"
```

"security"	"1.3.6.1.5"
"snmpV2"	"1.3.6.1.6"
"snmpDomains"	"1.3.6.1.6.1"
"snmpProxys"	"1.3.6.1.6.2"
"snmpModules"	"1.3.6.1.6.3"
"zeroDotZero"	"0.0"

Описание объектов можно посмотреть в файле
/opt/flussonic/lib/snmp2/mibs/FLUSSONIC-MIB.mib
в поле DESCRIPTION.

Защита контента с помощью DRM

DRM (Digital Rights (Restrictions) Management) — это способ защиты видео-контента при помощи шифрования парой ключей. Ключи выдаются

сервером ключей

DRM-системы.

Ниже на этой странице приведены настройки, одинаковые для всех поддерживаемых DRM систем.

По ссылкам приведены настройки для конкретных DRM. Flussonic Media Server поддерживает работу со следующими системами DRM:

DRM Conax

BuyDRM (KeyOS)

Widevine

PallyCon

EzDRM

Irdeto

PlayReady DRM

GS DRM

Описание механизма защиты DRM

Apple в спецификации протокола HLS описывает два штатных механизма шифрования: AES-128 и SAMPLE-AES. Flussonic Media Server поддерживает оба механизма, а также ряд DRM систем.

Механизмы отличаются лишь непосредственным способом шифрования данных и работают по трехсторонней схеме:

Flussonic запрашивает и получает ключ для шифрования контента от сервера ключей и URL этого ключа.

Клиент получает от Flussonic зашифрованный контент и URL ключа для дешифрования.

Сервер ключей получает от клиента запрос на ключ для дешифрования и решает: отдавать ключ или нет.

Если клиент получает контент от Flussonic по безопасному соединению и общается с сервером ключей по HTTPS, то можно надеяться на то, что он сможет расшифровать видео и проиграть его, не открыв доступ к нешифрованному контенту неавторизованным пользователям.

Механизмы получения ключа для видео-поточков и для файлов не отличаются.

Настройка шифрования в общем случае

Flussonic Media Server хранит контент в нешифрованном виде. Шифруется контент при отдаче клиенту.

Чтобы включить шифрование, добавьте строку `drm` в настройки потока или зоны файлов,

укажите метод шифрования и сервер ключей.

```
stream channel0 {  
    url udp://239.0.0.1:1234;  
    drm aes128 keyserver=http://192.168.0.80:4500/;  
}
```

Примеры настроек для отдельных DRM приведены в одноименных разделах (см. ссылки выше на этой странице).

После сохранения настроек Flussonic будет шифровать контент для всех протоколов, которые могут работать с указанным методом шифрования.

Внимание! Необходимо отключить все протоколы, которые несовместимы с выбранной DRM ситсемой. Если указанный метод шифрования поддерживается в протоколе HLS, но у вас остался работающим протокол HDS, то пользователь сможет без проблем проиграть видео по HDS без шифрования.

Чтобы этого не произошло, надо отключить все лишние протоколы для нужного потока или зоны файлов:

```
stream channel0 {  
    url udp://239.0.0.1:1234;  
    rtsp off;  
    rtmp off;  
    hds off;  
    mpegts off;  
    dash off;  
    drm aes128 keyserver=http://192.168.0.80:4500/;  
}
```

```
file vod {  
    rtsp off;  
    rtmp off;  
    hds off;  
    mpegts off;  
    dash off;  
    drm aes128 keyserver=http://192.168.0.80:4500/;  
}
```

Теперь доступ к контенту доступен только по протоколу HLS.

DRM для файлов

В случае с файлами, внешний сервер ключей не может явно указывать ключ, потому что не знает, когда будет открыт на чтение файл.

Необходимо сконфигурировать файл на явное обращение к серверу ключей:

```
file drm {  
    path priv;  
    hds off;
```



```
rtmp off;
rtsp off;
dash off;
drm aes128 keyserver=http://192.168.0.80:4500/;
}
```

При такой конфигурации Flussonic будет делать HTTP GET запросы к серверу ключей с параметром ?name=: <http://192.168.0.80:4500/?name=drm/bunny.mp4&number=1>

В ответ Flussonic ожидает ответ, в котором первые 32 байта будут HEX представлением ключа, используемого для шифрования. Также в этом ответе ожидается заголовок X-Key-Url, который будет сообщен клиенту. Поэтому X-Key-Url клиент будет ожидать 16-байтный ключ (не в HEX виде) для дешифровки.

Защита DVR архива с помощью DRM

Архив шифруется посегментно одним ключом, и каждые 10 минут Flussonic запрашивает новый ключ для следующей группы сегментов.

Внимание! Чтобы DRM защита DVR архива работала, сервер ключей должен хранить старые ключи (по старым URL) не менее по времени, чем глубина архива.

Включение шифрования всех кадров

По умолчанию Flussonic шифрует только ключевые кадры. Как правило, этого достаточно для защиты от несанкционированного доступа к потоку, а также снижает энергопотребление при дешифровании на стороне клиента. Однако некоторые Smart TV и приставки требуют, чтобы шифровались все кадры в потоке.

Чтобы включить шифрование всех кадров, используйте опцию encryption=full при настройке DRM.

Для включения режима шифрования только ключевых кадров используйте опцию encryption=sparse.

```
stream channel0 {
  url udp://239.0.0.1:1234;
  rtsp off;
  rtmp off;
  hds off;
  mpegts off;
  dash off;
  drm aes128 keyserver=http://192.168.0.80:4500/ encryption=full;
}
```

Ротация ключей шифрования

Многие серверы DRM периодически меняют (ротируют) ключи с целью более надежной защиты. По умолчанию Flussonic ротирует ключи сам — он запрашивает у DRM сервера новый ключ каждые 10 минут. Для изменения интервала ротации ключей используйте опцию expires и укажите требуемое время в минутах.

```
stream channel0 {  
    url udp://239.0.0.1:1234;  
    rtsp off;  
    rtmp off;  
    hds off;  
    mpegts off;  
    dash off;  
    drm aes128 keyserver=http://192.168.0.80:4500/ expires=60;  
}
```

Conax DRM

Настройка DRM для потока или VOD локации выглядит таким образом:

```
stream ort {  
    url udp://239.0.0.1:1234;  
    hds off;  
    rtmp off;  
    rtsp off;  
    mpegts off;  
    meta drm_id ev0234; # опционально, по умолчанию ContentID - имя потока  
    drm conax keyserver=https://uSeR:Passw0rd@cas-gateway:12346;  
}
```

```
file drm {  
    path /storage/vod;  
    hds off;  
    rtmp off;  
    rtsp off;  
    mpegts off;  
    drm conax keyserver=https://uSeR:Passw0rd@cas-gateway:12346;  
}
```

где

*https://uSeR:Passw0rd@cas-gateway:12346 - сервер ключей с учетной записью предоставленной Conax. *drm_id — параметр, который называется ContentId в Conax. Если вам необходимо указать ContentId, используйте meta drm_id <ContentId> для потоков. Для VOD файлов, например, для /path/to/vod/content.mp4 Flussonic будет читать ContentID из файла /path/to/vod/content.mp4.conax_id.

Замечание. Вам нужно указать только учетные данные, имя хоста и порт в директиве drm. Flussonic сам добавит путь /ca-server/webservices/key-server/conax.

Важно!

Если вы получили от Conax пароль, содержащий специальные символы, например aaa%bbb, будьте внимательны: вам может потребоваться экранировать некоторые специальные символы, когда вы добавляете пароль в файл конфигурации.

Приведенный в примере пароль следует переписать как aaa%25bbb, так как символ % пишется в адресах HTTP как %25.

BuyDRM KeyOS

Настройка DRM для потока или VOD локации выглядит таким образом:

```
stream ort {  
    url udp://239.0.0.1:1234;  
    hds off;  
    rtmp off;  
    rtsp off;  
    mpegts off;  
    meta drm_id RANDOM_UUID_VALUE;  
    drm keyos userkey=YOUR_USER_KEY;  
}
```

```
file drm {  
    path /storage/vod;  
    hds off;  
    rtmp off;  
    rtsp off;  
    mpegts off;  
    drm keyos userkey=YOUR_USER_KEY;  
}
```

Параметр userkey предоставляет KeyOS.

Для каждого файла и потока вам необходимо указать уникальный KeyID в формате UUID. Убедитесь, что при публикации нового файла или потока вы генерируете новый KeyID для него. Для генерации KeyID можно воспользоваться генератором UUID.

Если вы используете один и тот же KeyID для нескольких файлов/потоков, они будут зашифрованы одним и тем же ключом, и, получив одну лицензию, можно будет расшифровать всю группу.

Для потоков используйте строку meta drm_id <KeyID>.

Для VOD файла /path/to/vod/content.mp4 Flussonic будет читать KeyID из файла /path/to/vod/content.mp4.keyos_id.

Widevine DRM

Чтобы настроить Widevine DRM для потока или файлов, находящихся в определенной VOD-локации, добавьте параметр

drm

:

```
stream ort {  
  url udp://239.0.0.1:1234;  
  hds off;  
  rtmp off;  
  rtsp off;  
  mpegts off;  
  drm widevine aes_key=1234512345...123451234 iv=12345as...45asdfg12 signer=widevine_test;  
}
```

Где:

aes_key и iv — обязательные параметры. Их можно получить через ваш аккаунт на <https://www.widevine.com/>.

content_id - необязательный параметр. Задает уникальный идентификатор содержимого. По умолчанию content_id равен названию потока.

При использовании тестового ключа необходимо указать опцию signer:

signer — ваш уникальный ID, который Flussonic будет использовать для соединения с сервером ключей <http://license.uat.widevine.com/cenc/getcontentkey/<signer>>.

В случае использования production-ключа необходимо указать опцию keyserver:

keyserver — задает кастомный URL для сервера ключей Widevine. Например, keyserver=<http://license.widevine.com/cenc/getcontentkey/<signer>>. В этом случае параметр signer - необязательный, поскольку signer берется из URL, заданного в параметре keyserver.

PallyCon DRM

Чтобы настроить PallyCon DRM для потока или файлов, находящихся в определенной VOD-локации, добавьте параметр

drm

:

```
stream ort {  
  url udp://239.0.0.1:1234;  
  hds off;  
  rtmp off;  
  rtsp off;  
  mpegts off;  
  drm pallycon enc_token="12345asdfg...12345asdfg12";  
}
```

Где:

enc_token — токен API аутентификации, который генерируется, когда вы логинитесь в PallyCon сервис на <https://login.pallycon.com/>.

PallyCon использует CPIX для обмена ключами.

EzDRM

Flussonic поддерживает два варианта обмена ключами в системе EzDRM:

EzDRM с использованием CPIX протокола для передачи ключей

EzDRM с использованием собственного формата EzDRM для передачи ключей.

Настройка EzDRM (обмен ключами по CPIX)

Чтобы настроить EzDRM с обменом ключами в формате CPIX для потока или файлов, находящихся в определенной VOD-локации, добавьте параметр `drm ezdrm`:

```
stream ort {  
    url udp://239.0.0.1:1234;  
    hds off;  
    rtmp off;  
    rtsp off;  
    mpegts off;  
    drm ezdrm password="password" user="user@ezdrm.com";  
}
```

```
file drm {  
    path /storage/vod;  
    hds off;  
    rtmp off;  
    rtsp off;  
    mpegts off;  
    drm ezdrm password="password" user="user@ezdrm.com";  
}
```

Где:

`user` и `password` — обязательные параметры. Их можно получить через ваш аккаунт на <https://www.ezdrm.com>.

Для VOD файла `/storage/vod/content.mp4` Flussonic будет читать `ContentId` из файла `/storage/content.mp4.ezdrm_id`.

Если указать дополнительно опцию `meta drm_id` — DRM ID будет использоваться в запросах потока вместо его имени.

```
stream ort {  
    url udp://239.0.0.1:1234;  
    hds off;  
    rtmp off;  
    rtsp off;  
    mpegts off;  
    meta drm_id 12345asdfg12345asdfg12345asd;  
    drm ezdrm password="password" user="user@ezdrm.com";  
}
```

```
}
```

Настройка EzDRM (обмен ключами в формате EzDRM)

Чтобы настроить EzDRM с обменом ключами в собственном формате для потока или файлов, находящихся в определенной VOD-локации, добавьте параметр `drm ezdrm_classic`:

```
stream ort {  
    url udp://239.0.0.1:1234;  
    hds off;  
    rtmp off;  
    rtsp off;  
    mpegts off;  
    drm ezdrm_classic password="password" user="user@ezdrm.com";  
}
```

Где:

`user` и `password` — обязательные параметры. Их можно получить через ваш аккаунт на <https://www.ezdrm.com>.

Если указать дополнительно опцию `meta drm_id` — DRM ID будет использоваться в запросах потока вместо его имени.

```
stream ort {  
    url udp://239.0.0.1:1234;  
    hds off;  
    rtmp off;  
    rtsp off;  
    mpegts off;  
    meta drm_id 12345asdfg12345asdfg12345asd;  
    drm ezdrm password="password" user="user@ezdrm.com";  
}
```


Веб-интерфейс

Список активных каналов

По закладке «Media» доступен список запущенных потоков. Можно проиграть каждый из потоков по желаемому протоколу, посмотреть архив (если он включен), посмотреть в реальном времени графики трафика и количества клиентов потока, посмотреть количество и список клиентов (кликнув на количество). Так же можно увидеть, как давно поток перестал получать видео и рестартовать его.

Общий мониторинг

Во вкладке общего мониторинга можно увидеть графики сетевой активности, клиентов, потребления памяти, использования диска и т.п.

Конфигурация

Раздел конфигурации позволяет настроить все параметры сервера из веб-интерфейса. Подробнее в разделе про настройку.

Помощь

При обращении в техподдержку надо зайти в раздел Upload debug и загрузить логи отсюда.

Инсталляция

Содержание:

Прочитайте перед установкой Flussonic: как поменять пароль?

Установка под Debian/Ubuntu

Установка RPM под CentOS/RedHat

Количество открытых дескрипторов

Обновление Flussonic Media Server

Какая версия Flussonic стоит сейчас?

Как откатиться на предыдущую версию?

Запуск и остановка Flussonic Media Server

Системные требования

Прочитайте перед установкой: Как поменять пароль?

Чрезвычайно важно!

Сразу после установки измените логин/пароль администратора для управления Flussonic Media Server. По умолчанию, логин/пароль: flussonic и letmein!.

Чтобы изменить пароль администратора, отредактируйте в конфигурационном файле `/etc/flussonic/flussonic.conf` значение директивы `edit_auth`.

После редактирования конфигурационного файла вручную необходимо перечитать настройки. Выполните команду:

```
/etc/init.d/flussonic reload
```

Другой способ — смените пароль через веб-интерфейс в Config > Access:

Установка под Debian/Ubuntu

Поддерживаемые архитектуры: amd64, armhf, arm64. Архитектура i686 не поддерживается.

Требуемая версия ОС: Ubuntu 16.04 и новее, Debian 8 Jessie и новее.

Установите Flussonic Media Server из Debian package:

```
wget -q -O /etc/apt/trusted.gpg.d/flussonic.gpg http://apt.flussonic.com/binary/gpg.key;
```

```
echo "deb http://apt.flussonic.com binary/" > /etc/apt/sources.list.d/flussonic.list;
```

```
apt update;
```

```
apt -y install flussonic flussonic-transcoder;
```

Замечание. Пакет flussonic-transcoder необходим только в случае, если вы планируете использовать CPU для выполнения транскодирования.

Чтобы узнать, какие пакеты можно установить, используйте следующую команду вместо той, где пакеты перечислены:

```
apt-get -y --install-recommends --install-suggests install flussonic
```

Запустите Flussonic Media Server:

```
/etc/init.d/flussonic start
```

Теперь необходимо открыть в браузере веб-интерфейс Flussonic Media Server по адресу `http://flussonic-ip:8080/` и скопировать туда полученный от нас ключ (замените "flussonic-ip" на адрес вашего сервера).

На этой же странице измените пароль администратора для управления Flussonic Media Server.

Лицензионный ключ хранится в файле `/etc/flussonic/license.txt`, его можно вписать туда вручную перед запуском.

Проверить правильную установку Flussonic Media Server можно по адресу `http://flussonic-ip:8080/`, где flussonic-ip — адрес того сервера, на который вы поставили ПО. Также можно выполнить команду:

```
/etc/init.d/flussonic status
```

Установка завершена, Flussonic Media Server готов к работе. Однако, для большого количества клиентов необходимо произвести тюнинг ОС.

Также придется полностью отключить swar, так как его наличие несовместимо с видеостримингом.

Если на сервере не хватает оперативной памяти, её нельзя расширять с помощью swar.

Установка RPM на CentOS/RedHat или подобные системы

Важно!

Мы очень не рекомендуем использование rpm-дистрибутивов: CentOS, RedHat, Suse и т.п. Это связано с целым перечнем причин.

Мы не оказываем поддержки по проблемам, связанным с использованием rpm-пакетов и rpm-дистрибутивам пользователям, покупающим меньше 10 лицензий.

Устанавливаем Flussonic Media Server из Yum repository:

```
cat > /etc/yum.repos.d/Flussonic.repo <<EOF
[flussonic]
name=Flussonic
baseurl=http://apt.flussonic.com/rpm
enabled=1
gpgcheck=0
EOF
yum -y install flussonic-erlang flussonic flussonic-transcoder
# Положить ключ в /etc/flussonic/license.txt
/etc/init.d/flussonic start
```

Количество открытых дескрипторов

По умолчанию, в OS Linux количество максимально открытых файлов для процесса часто бывает выставлено в 1024. Это означает, что больше 1000 пользователей подключиться не

сможет даже, если ресурсов ещё достаточно. Добавьте следующие строки в `/etc/security/limits.conf`:

```
root      soft  nofile 65536
root      hard  nofile 65536
```

Количество открытых дескрипторов превышает количество подключенных клиентов.

Важно!

Синтаксис файла `limits.conf` может меняться между разными ОС. В этой документации приведен работающий вариант синтаксиса для Debian/Ubuntu. Под другими ОС синтаксис может быть другой.

Обновление Flussonic Media Server

Для обновления достаточно ещё раз обновить пакеты и установить Flussonic Media Server:

```
apt-get update
apt-get -y install flussonic
/etc/init.d/flussonic restart
```

Какая версия стоит сейчас?

```
dpkg -l | grep flussonic
# ii flussonic 3.71.14 amd64 Videostreaming server
```

Как откатиться на предыдущую версию?

Иногда может потребоваться установить предыдущую версию. Для этого необходимо указать версию пакета `flussonic` и его зависимостей.

Предположим, что нужна версия 19.06.1.

Узнаем версии зависимостей с помощью `apt-cache`:

```
apt-cache show flussonic=19.06.1 | egrep '^(Depends|Suggests):'
Depends: flussonic-erlang (=21.3.6) | flussonic-erlang_21.3.6, flussonic-transcoder-base (=4.6.2) |
flussonic-transcoder-base_4.6.2
Suggests: flussonic-transcoder (>= 4.5)
```

Установим с указанием полученных версий:

```
apt-get install flussonic=19.06.1 flussonic-erlang=21.3.6 flussonic-transcoder-base=4.6.2
```

Важно!

Перед установкой пакетов обязательно делайте резервную копию конфигурационных файлов из директории `/etc/flussonic` и `.db` файлов из директории `/opt/flussonic/priv` (эта директория используется по умолчанию, в конфиге вы можете задать произвольный путь).

Важно!

На других дистрибутивах Linux, для которых нет пакетов, мы не можем гарантировать работоспособность сервера.

Запуск и остановка Flussonic Media Server

Необходимо использовать следующие команды:

для запуска сервиса:

```
/etc/init.d/flussonic start
```

для остановки сервиса:

```
/etc/init.d/flussonic stop
```

для перезапуска сервиса:

```
/etc/init.d/flussonic restart
```

для переконфигурации без отключения клиентов:

```
/etc/init.d/flussonic reload
```

Системные требования

Минимальные системные требования к конфигурации сервера для работы Flussonic Media Server приведены в таблице ниже. Реальные требования могут различаться в зависимости от количества одновременных подключений, которое будет обслуживать Flussonic Media Server.

Важно!

При планировании мощностей сервера, должны быть также учтены ресурсы, необходимые для работы операционной системы и других сервисов, которые будут функционировать параллельно с Flussonic Media Server.

Минимальные системные требования

Кол-во одновременных подключений		10	100	1 000	5 000+	
Процессор	Любой	1-ядерный	4-х ядерный (Xeon / Core i7)		2-х процессорный	
Xeon E5	Оперативная память		128 Мб	256 Мб	1024 Мб	16 Гб
Место на жестком диске		40 Мб	40 Мб	40 Мб	40 Мб	Сетевой адаптер
100 Мбит/с	1 Гбит/с	1 Гбит/с	серверный	10 Гбит/с	Intel	Операционная
система	Debian Linux, Ubuntu Linux					

Для стабильного воспроизведения видео потока у клиентов при большом количестве одновременных подключений рекомендуется организовать распределение сетевого трафика по нескольким физическим серверам. Подробная информация о кластеризации серверов Flussonic Media Server находится в разделе о кластеризации.

Необходимо учитывать, что при передаче мультимедиа-данных из файлов на жестком диске, основная нагрузка ложится на дисковую подсистему. Соответственно, при планировании конфигурации Flussonic Media Server, особое внимание должно быть уделено производительности используемых жестких дисков. Подробнее можно прочитать в разделе про вещание файлов.

Если на сервере имеется Firewall, необходимо настроить соответствующие разрешающие правила для возможности установки соединений по протоколам, которые будет обслуживать Flussonic Media Server.

Так же важно помнить, что при использовании коммерческой версии, важно что бы сам Flussonic Media Server имел доступ в интернет по протоколам HTTP и HTTPS.

Ночные сборки

Мы выпускаем новую версию Flussonic каждый месяц, но у нас также есть репозиторий с ночными сборками. Каждый день он обновляется новой сборкой Flussonic, которая содержит новые функции и исправления ошибок. Ночные сборки — это Release Candidate-версии, которые мы используем в нашей лаборатории и предлагаем некоторым клиентам, которые хотят получить обновления до выхода следующего официального релиза.

Каждая такая сборка проходит тысячи автотестов и должна быть пригодна для использования на боевом сервере, однако мы не гарантируем этого, так как ночные сборки не проходят ручное тестирование и имеют недокументированные функции.

Важно! Помните, что вы используете ночные сборки на свой страх и риск, даже если инженеры нашей службы поддержки предложили вам их использовать. Будьте осторожны с использованием ночных сборок в производственной среде: делайте ежедневные резервные копии и будьте готовы вернуть ваш Flussonic к официальной версии.

Служба поддержки всегда ответит на ваш вопрос, но в рамках базового плана мы не можем немедленно реагировать на каждый инцидент. Наш опыт показывает, что большинство проблем можно решить самостоятельно. Однако если вы не уверены в своих навыках работы с Linux, не пытайтесь устанавливать ночные сборки.

Эта инструкция поможет вам установить ночную сборку, вернуться к официальному выпуску, создать резервную копию конфигурации и подготовить информацию для группы поддержки в случае сбоя при установке.

Установка ночной сборки

Удалите установленную версию Flussonic и ее зависимости:

```
apt remove flussonic
```

Измените используемый репозиторий на репозиторий с ночными сборками и установите Flussonic:

```
rm -f /etc/apt/sources.list.d/erlyvideo.list;  
echo "deb http://apt.flussonic.com/repo master/" > /etc/apt/sources.list.d/flussonic.list;  
apt update;  
apt install flussonic;  
service flussonic restart
```

Возврат к официальному релизу

Удалите установленную версию Flussonic и ее зависимости:

Важно! Перед удалением пакетов создайте резервную копию файлов конфигурации, расположенных в директории /etc/flussonic и файлов .db из директории /opt/flussonic/priv (эта директория используется по умолчанию, но можно прописать другую в конфигурационном файле Flussonic).

```
apt remove flussonic
```

Измените используемый репозиторий на репозиторий с официальными выпусками и установите Flussonic:

```
rm -f /etc/apt/sources.list.d/erlyvideo.list;
```

```
echo "deb http://apt.flussonic.com binary/" > /etc/apt/sources.list.d/flussonic.list;
```

```
apt update;
```

```
apt install flussonic;
```

```
service flussonic restart
```

Важно! Если Flussonic не запускается, выполните команды `systemctl status flussonic.service` и `journalctl -xe` и покажите результат службе поддержки.

Обновление Flussonic

Обновите пакет Flussonic, когда вышла версия с новыми возможностями или с исправлениями ошибок более ранних версий. Мы рекомендуем обновляться более или менее регулярно. Вы сможете откатить изменения, если необходимо.

Об изменениях в новых версиях читайте в нашем блоге: <https://erlyvideo.ru/blog>

Обновление на Debian и Ubuntu

```
apt-get update
apt-get -y install flussonic
/etc/init.d/flussonic restart
```

Важно. Flussonic после обновления надо перезапускать вручную (это сделает третья команда в примере).

Менеджер пакетов может сообщить об изменении файла /etc/flussonic/flussonic.conf. В таком случае мы советуем сохранить уже установленную версию, нажав клавишу N:

Файл настройки «/etc/flussonic/flussonic.conf»

==> Изменён с момента установки (вами или сценарием).

==> Автор пакета предоставил обновлённую версию.

Что нужно сделать? Есть следующие варианты:

Y или I : установить версию, предлагаемую сопровождающим пакета

N или O : оставить установленную на данный момент версию

D : показать различия между версиями

Z : запустить оболочку командной строки для проверки ситуации

По умолчанию сохраняется текущая версия файла настройки.

*** flussonic.conf (Y/I/N/O/D/Z) [по умолчанию N] ? N

Конфигурационный файл из пакета будет сохранён в файле: /etc/flussonic/flussonic.conf.dpkg-dist. После изучения изменений его можно удалить.

Замечание. Дополнительный пакет flussonic-transcoder необходим только в случае, если вы планируете использовать CPU для выполнения транскодирования. Если вы используете Nvidia NVENC, то он не нужен.

Обновление под CentOS

Обновление Flussonic под CentOS выполняется аналогично установке:

```
yum -y install flussonic flussonic-erlang flussonic-transcoder
```

Менеджер пакетов может создать файл /etc/init.f/flussonic.rpmnew. Переименуйте его:

```
mv /etc/init.f/flussonic.rpmnew /etc/init.f/flussonic
```

Перезапустите Flussonic после обновления:

/etc/init.d/flussonic restart

Конфигурирование Flussonic Media Server

Содержание:

- Управление конфигурацией
- Перезапуск сервера Flussonic
- Глобальные опции сервера
- Настройки потока или группы
- Настройки VOD
- HTTP API для управления конфигурацией

Управление конфигурацией Flussonic Media Server

Настройки Flussonic Media Server хранятся в конфигурационном файле `/etc/flussonic/flussonic.conf`.

Есть несколько способов задать настройки Flussonic:

Отредактировать файл `/etc/flussonic/flussonic.conf` и применить настройки — см. ниже.

Задать настройки через веб-интерфейс. Веб-интерфейс поддерживает большинство необходимых настроек, но иногда настройку можно указать только через файл.

Воспользоваться HTTP API для управления настройками.

Воспользоваться Cluster SQL API.

После изменения параметров в конфигурационном файле `flussonic.conf`, необходимо применить настройки сервиса Flussonic Media Server следующей командой:

```
/etc/init.d/flussonic reload
```

Также для применения новых настроек можно воспользоваться HTTP API :

```
curl -u user:pass http://localhost:8080/flussonic/api/reload
```

Где `user:pass` — логин и пароль из настройки `edit_auth` (см. Другие опции сервера ниже на странице).

Перезапуск сервера Flussonic

Для рестарта сервера выполните команду:

```
/etc/init.d/flussonic restart
```

Глобальные опции сервера

В глобальных опциях указываются порты для протоколов и общие настройки.

Опции сервера (Протоколы и порты) Описание `https 443;` включает приём запросов по протоколу HTTPS на указанном порту. Можно указывать несколько раз. `http 80;` включает приём запросов по протоколу HTTP на указанном порту. Можно указывать несколько раз. `http 127.0.0.1:80;` включает приём запросов по протоколу HTTP на указанном порту

и заданном IP адресе. Можно указывать несколько раз.
запросов по протоколу RTMP на указанном порту.
запросов по протоколу RTMPS на указанном порту.
по протоколу RTSP на указанном порту.
протоколу RTSPS на указанном порту.
протоколу MySQL на указанном порту.

rtmp 80; включает приём
rtmps 1443; включает приём
rtsp 554; включает приём запросов
rtsp 1554; включает приём запросов по
mysql 3306; включает приём запросов по

Замечания

При конфигурации протоколов HTTPS, RTMPS, и RTSPS Flussonic Media Server ожидает наличие сертификатов в каталоге /etc/flussonic.

Для того, чтобы протокол RTMPS заработал, необходимо наличие валидного сертификата, который работает без каких-либо предупреждений или ошибок.

Flussonic Media Server ожидает приватный ключ сервера в файле /etc/flussonic/flussonic.key с паролем flussonic. Сертификат сервера будет читаться из файла /etc/flussonic/flussonic.crt. Промежуточный и CA сертификаты будут браться из /etc/flussonic/flussonic-ca.crt.

Например, при получении купленного комплекта из ключа и сертификатов, надо сделать следующие действия:

bash

```
cat intermediate.crt ca.crt > /etc/flussonic/flussonic-ca.crt
cp server.crt flussonic.crt
openssl rsa -des3 -in server.key -out flussonic.key
```

Опция сервера	Описание	loglevel info;	Меняется объём записываемых в журнал логов. Варианты: debug, info, alert.
		logrequests true;	Включается запись всех HTTP запросов в /var/log/flussonic/access.log
		total_bandwidth 10G	Указывается максимальная ёмкость канала. Нужно для балансировки серверов в кластере.
		auth_token TOKEN;	Имя параметра в query string, который будет интерпретироваться как авторизационный токен.
		auth false;	Глобальное отключение авторизации.
		auth http://backend/auth.php;	Глобальное включение авторизационного бекенда.
		max_sessions 1000;	Глобальный лимит по количеству сессий.
		no_auto_token; auto_token false;	При указании этой опции запросы пришедшие без опции token в query string будут немедленно отключаться.
		auto_token UUID;	При указании этой опции авторизационный token будет автоматически генерироваться, если он не указан в query string.
		auto_token blank;	При указании этой опции авторизационный token будет учитываться как пустой, если он не указан в query string.
	Режим по умолчанию.	cluster_key SECRETKEY;	Строка, используемая для авторизации других серверов Flussonic Media Server входящих в кластер.
		view_auth USER PASSWORD;	Включается авторизация для readonly доступа к API.
		edit_auth USER PASSWORD;	Логин-пароль для административного доступа к серверу.
		api_allowed_from 10/8 192.168/16;	Можно указать с каких адресов или сетей можно разрешать доступ к API.
		notify HANDLER_NAME { sink http://backend/event.php;}	notify HANDLER_NAME { sink /etc/flussonic/events.lua;}
	События Flussonic будут приходить на указанный URL или в указанный скрипт. Подробнее в Events API	pulsedb /var/lib/flussonic;	Указывается, куда будет писаться статистика потоков.
		session_log /var/lib/flussonic;	Указывается, куда будет писаться история сессий.
		session_log false;	Отключает запись сессий.
	url_prefix PREFIX; url_prefix http://my.domain.address.com:8080;		При использовании протокола HLS, для всех потоков на сервере, адреса отдельных сегментов и плейлистов в вариантном плейлисте будут начинаться с указанного префикса. Эта опция может быть использована не только в глобальной части файла конфигурации, но и как часть настроек конкретного потока. Прописанная в настройках потока, опция применяется только к одному этому потоку.

`source SOURCE/PREFIX; source SOURCE/PREFIX { }source origin1.tv { }` Директива, включающая автоматическое зеркалирование потоков с другого сервера на локальный.

`stream ntv { url tshttp://source/ntv.ts; }` Директива `stream` включает перманентный поток, который будет поддерживаться рабочим всё время жизни сервера, даже если ни один из источников недоступен. Опции директивы `stream` описаны ниже.

`ondemand ntv { url tshttp://source/ntv.ts; }` Директива `ondemand` описывает поток, который будет запускаться по запросу. Если он какое-то время недоступен, то он будет автоматически выключаться. Опции директивы `ondemand` описаны ниже.

`rewrite client16/* { url rtmp://origin/%s; }` Директива `rewrite` включает динамический запуск потоков по запросу, чьи имена попадают под маску `clients16/*`. Опции директивы `rewrite` описаны ниже.

`live published { }` Директива `live` включает возможность публикации на сервер потоков с именами, начинающимися с `published`. Опции директивы `live` описаны ниже.

`file vod { path /storage; }` Директива `file` включает вещание файлов, лежащих в каталоге `/storage` под именами, начинающимися с `vod/`. Опции директивы `file` описаны ниже.

`cache globalcache /var/www misses=4 2d 40G;` Конфигурируется глобальный кэш по имени `globalcache` в каталоге `/var/www` с лимитом на 40 гигабайт и 2 дня. Файлы начнут кэшироваться только после 4 запросов от клиентов.

`nvidia_monitor true;` Директива `nvidia_monitor` включает (`true`) или выключает (`false`) сохранение статистики о производительности Nvidia. Статистику можно посмотреть, выполнив запрос в разделе Pulse в UI, например: ``sum:1m-avg:gpu_dec{from=-2h,gpu=nv0}``. Подробнее

`geoip PATH_TO_DATABASE;` Директива `geoip` задаёт путь до файла базы данных GeoLite2, которую Flussonic будет использовать для геолокации вместо встроенной базы. Подробнее

Настройки потока или группы

Эти настройки можно указывать внутри директив `stream`, `ondemand`, `rewrite` и `live`. Мы называем их опции.

auth

`auth http://backend/;` Включение авторизации для потока. Более подробно в разделе про авторизацию.

domains

`domains host1.ru *.host1.ru;` Указание доменов, на которых можно проигрывать это видео. Не работает для клиентов, которые не передают `Referer`. Для правильной работы в WEB в списке доменов должен присутствовать домен, с которого загружается `embed` плеер.

allowed_countries

`allowed_countries RU UA KZ;` Список стран (двухбуквенные коды как в БД `maxmind`) с которых разрешен доступ.

url

`url tshttp://transcoder:port/;` Адрес источника. Можно перечислить несколько директив `url` для выбора источника по кругу.

Важно! Если используется UDP-источник, то в файле конфигурации этот UDP-адрес должен использоваться только один раз. Если несколько потоков будут использовать один и тот же адрес, то, скорее всего, это просто не будет работать.

urls

`urls source1 source2;` Список адресов источников. Подробнее про механизм переключения

источников.

url_prefix

`url_prefix prefix` При использовании протокола HLS, адреса отдельных сегментов и плейлистов в вариантном плейлисте будут начинаться с указанного префикса. Эта опция может быть использована не только как часть настроек конкретного потока, но и в глобальной части файла конфигурации. Прописанная глобально опция применяется ко всем потокам на этом сервере. Пример: `url_prefix http://my.domain.address.com:8080`

dvr

`dvr /storage 1d 50% schedule=8:00-16:00;` `dvr @my RAID 1d 50% schedule=8:00-16:00;`
Включение архива. Flussonic Media Server будет писать в каталог `/storage/streamname` или дисковый массив `@my RAID`. Очищать архив он будет либо через день, либо когда диск, на который пишут, заполнится на 50%. Вместо дней можно указывать часы: 20h. Параметр `schedule` позволяет задать расписание для записи архива в виде интервалов. Время указывается в UTC в часах и опционально минутах, интервал может переключаться через полночь: 22-1:30. Расписание может содержать несколько интервалов, разделенных запятой: 8:00-16:00,22-1:30.

dvr_offline

`dvr_offline /storage 1d 50%;` При указании этой опции, поток не будет включать запись на старте. Её надо будет включать явно через API. Эта опция указывается вместо опции ``dvr``.

udp

`udp 239.0.0.1:4501 multicast_loop;` `udp 239.0.0.1:4501;` Flussonic Media Server будет посылать поток по UDP MPEG-TS. Чтобы задать UDP сокету параметр `MULTICAST_TTL`, используйте следующий синтаксис: `udp 239.0.0.1:4501?ttl=8;`. Чтобы задать постоянный битрейт (CBR), используйте следующий синтаксис: `udp 239.0.0.1:4501?cbr=2000;`, где 2000 — битрейт в kbit/sec.

thumbnails

`thumbnails;` Включает генерацию JPEG-скриншотов потока..

retry_limit

`retry_limit 10;` Количество раз, которое Flussonic Media Server будет пытаться подключиться к источникам, прежде чем отключить не статический поток.

clients_timeout

`clients_timeout 10;` Количество секунд которое Flussonic Media Server будет держать нестатический стрим после последнего обращения клиента.

source_timeout

`source_timeout 10;` Количество секунд, которое Flussonic Media Server будет ожидать кадры от источника. По прошествии этого времени Flussonic Media Server переподключится к источнику. По умолчанию значение `source_timeout` равно 60 секундам.

frames_timeout

`frames_timeout 3;` Время в секундах, в течение которого Flussonic Media Server ожидает кадры от источника прежде чем генерировать событие `frames_timed_out`. Это время должно быть

меньше, чем `source_timeout`. Событие `frames_timed_out` уведомляет вас о том, что источник, возможно, скоро отключится. Если от источника снова начали поступать кадры, до наступления `source_timeout`, Flussonic генерирует событие `frames_restored`.

password

`password secret`; Пароль, который должен передаваться в query string (`http` или `rtmp`) для публикации в поток или в группу.

push

`push rtmp://destination-server/name`; При указании этой опции Flussonic Media Server будет публиковать поток на другой сервер.

backup

`backup vod/blank.mp4`; Указание этой опции в потоке будет запускать указанный файл `vod/blank.mp4` пока нет видео от источника.

url publish://

`url publish://`; Указание этой опции в потоке позволяет публиковать в него видео. Не используется для группы потоков.

on_publish

`on_publish http://host/publish.php`; `on_publish /etc/flussonic/publish.lua`; При публикации в этот поток или группу потоков будет вызывать скрипт или делаться HTTP запрос с параметрами в виде названия потока, IP адреса публикующего и т.п. В ответ можно либо отказать в публикации, либо разрешить: — HTTP бэкенд должен вернуть либо 200 OK, либо 403 Forbidden; — .lua скрипт должен вернуть либо `{true, {}}`, либо `{false, {}}`.

max_sessions

`max_sessions 1000`; Лимит по количеству сессий на поток.

settings_rtp

`rtp udp`; Включает принудительное использование UDP для работы с RTSP камерами.

add_audio_only

`add_audio_only`; Добавляет в HLS плейлист ссылку на поток только с аудио. Требуется для прохождения валидации приложения в Apple.

prepush off

`prepush off`; Отключается `[prepush](#prepush)`, т.е. более быстрый старт. Может быть нужно для сохранения риа тайма в потоке.

prepush

`prepush 10`; Включает буфер заданной продолжительности. Если соединение клиента с сервером прерывается или замедляется, то он проигрывает видео из буфера, что позволяет плееру стартовать быстрее, но с отставанием.

max_bitrate

`max_bitrate 1000`; Лимит битрейта публикуемого потока.

logo

В версии 4.6.15 и выше.

logo path=flu/embed-logo.png height=100 width=100 left=0 top=0; Добавить логотип при воспроизведении потока. Такой логотип не будет отображаться на мобильных устройствах и в DVR плеере. Чтобы добавить логотип к видео используйте транскодер. path (Обязательный параметр) — путь относительно директории wwwroot. height, width — размер изображения логотипа в пикселях. Если задан только один из этих параметров, то второй будет изменен пропорционально. Не указывайте эти параметры, чтобы оторазить логотип в исходном размере. left, top, right, bottom — положение логотипа заданное в виде смещения в пикселях от левого, верхнего, правого и нижнего края видео. Например, чтобы отобразить логотип в правом нижнем углу: `right=0, bottom=0`. Не используйте одновременно параметры left, top, right и bottom.

mpegts_pids

mpegts_pids pmt=4095 sdt=0x12 v1=211 v2=212 a0=220 t0=16#fb; С помощью этого параметра задаются значения PID исходящему MPEG-TS потоку. Можно задать PID для PMT, SDT, дорожек с видео, аудио и субтитрами. Дорожки нумеруются начиная с 1: `a1=123` установит PID для первой аудио дорожки. Можно задать базовый PID для дорожек определённого типа с помощью индекса 0. Например, `t0=100` установит первой дорожке субтитров PID равный 101, второй - 102 и т.д. Числа могут быть заданы в десятичном виде (по умолчанию) или префиксом `0x` для шестнадцатеричных чисел.

segment_duration

segment_duration 4; Регулирует длительность сегмента в HLS и HDS потоках (в секундах). Не для всех получаемых потоков Flussonic сможет применить указанную длительность сегмента. Всё зависит от длины GOP у потока. Нужно, чтобы сегмент был кратен GOP, потому что GOP нельзя порезать на части. Например, для потока с GOP в 4 секунды допустима длина сегмента 4 секунды, 8 секунд, 12 секунд и т.д. Если это не так, то Flussonic создаёт сегменты равные GOP.

segment_count

segment_count 4; Регулирует количество сегментов в HLS и HDS плейлистах.

group

group sport; Используется только на сервере-источнике, чтобы отнести поток к одной или нескольким группам ТВ-каналов. Подробнее

disabled

disabled; Отключить поток.

Настройки VOD

Эти настройки можно указывать внутри директивы file, которая хранит настройки вещания файлов. Мы называем их опции.

file

file vod { path /storage;} Конфигурации локации для проигрывания файлов.

cache

cache /ssd misses=5 2d 40G; Все запросы к файлам будут кешироваться в папке /ssd сроком не более чем 2 дня и с лимитом в 40G. Кеширование включится, если к одному файлу было более 5 некешированных запросов.

domain

domain host.ru; Указание доменов, на которых можно проигрывать это видео. Не работает для клиентов, которые не передают Referer.

domains

domains host1.ru *.host1.ru; Указание доменов, на которых можно проигрывать это видео. Не работает для клиентов, которые не передают Referer.

path

path /storage; path s3://key:secret@s3.amazonaws.com/bucket/; Путь для поиска файлов. Можно указать несколько путей.

read_queue

read_queue 100; Количество одновременных обращений к диску для этого префикса.

download

download; Эта опция разрешает скачивание файла и обращение к нему с Range запросами.

max_readers

max_readers 10; Опция настраивает максимальное количество одновременных дисковых запросов ко всему префиксу.

thumbnails

thumbnails offset=10; Опция включает генерацию скриншотов к файлам с опциональным указанием секунды видео.

auto_mbr

auto_mbr; Включает автоматическое создание мультибитрейтного HLS плейлиста из нескольких файлов с разным битрейтом.

Мониторинг

Для мониторинга можно использовать Monit. Monit — это самостоятельный демон, работающий на операционных системах GNU/Linux (например, Ubuntu), который может проверять существование процесса по PID, работу портов, ресурсы, используемые процессом (CPU, RAM) и многое другое.

Установка программ

Вам нужно установить monit и postfix. Postfix используется для рассылки писем.

В Ubuntu для установки используются команды:

```
sudo apt-get install monit
sudo apt-get install postfix
```

Конфигурация Monit

Файл конфигурации Monit находится здесь: /etc/monit/monitrc PID-файл Flussonic находится здесь: /var/run/flussonic/pid

Вот пример конфигурации, применительно к Flussonic:

```
set mailserver localhost
set mail-format { from: monit@example.com }
set mail-format { from: support@example.com }
set alert sys@example.com
set httpd port 2812 and allow localhost

check process flussonic with pidfile /var/run/flussonic/pid
  start program = "/etc/init.d/flussonic start"
  stop program = "/etc/init.d/flussonic stop"
  if failed host 127.0.0.1 port 1935 type tcp
  with timeout 5 seconds
  then restart

check system 127.0.0.1
  if memory usage > 75% for 4 cycles then alert
  if cpu usage (user) > 80% for 4 cycles then alert
```

(example.com и почтовые адреса даны для примера, их нужно заменить на настоящие)

Более подробную информацию о конфигурации Monit можно узнать на официальном сайте. Также в интернете есть статьи о нем и на русском языке, например, вот эта статья на Хабрахабре.

Оптимизация Flussonic Media Server и операционной системы

В этом разделе будут приведены некоторые часто встречающиеся проблемы и возможности детальной настройки ОС и Flussonic Media Server для больших нагрузок.

Настройка UDP захвата

При захвате UDP под линуксом необходимо увеличить размер памяти под UDP буферы:

```
sysctl -w net.core.rmem_max=1048576
sysctl -w net.core.rmem_default=1048576
sysctl -w net.ipv4.udp_mem="8388608 12582912 16777216"
```

Эти настройки будут работать до перезагрузки. Чтобы сохранить эти параметры навсегда, нужно отредактировать файл /etc/sysctl.conf.

С самый конец файла добавить:

```
net.core.rmem_max = 1048576
net.core.rmem_default=1048576
net.ipv4.udp_mem = 8388608 12582912 16777216
```

Чтобы применить изменения из файла, нужно выполнить команду `sudo sysctl -p`.

Работа с большим количеством памяти

При наличии более 60 гигабайт памяти рекомендуется зарезервировать 10 гигабайт под Linux:

```
sysctl vm.min_free_kbytes=10240000
```

Настройка TCP/IP стека

Если вы используете Flussonic Media Server для вещания более чем на 3-4 Гбит/с для вас могут стать ощутимыми тонкости настройки TCP стека в Linux.

Во-первых, требуется увеличить количество доступной памяти для буферов соединений:

```
sysctl -w net.core.wmem_max=16777216
sysctl -w net.ipv4.tcp_wmem="4096 4194394 16777216"
sysctl -w net.ipv4.tcp_congestion_control=htcp
sysctl -w net.ipv4.tcp_slow_start_after_idle=0
```

Эти настройки будут работать до перезагрузки. Чтобы сохранить эти параметры навсегда, нужно отредактировать файл /etc/sysctl.conf, в самый конец файла вставить:

```
net.core.wmem_max = 16777216
net.ipv4.tcp_wmem = 4096 4194394 16777216
```

Чтобы применить изменения из файла, нужно выполнить команду `sudo sysctl -p`.

Так же надо поменять настройки сетевого интерфейса: `ifconfig eth0 txqueuelen 10000`.

Обязательно надо проверить версию драйвера сетевой карты. Желательно использовать самую свежую версию. Выяснить версию драйвера и firmware можно так:

```
# ethtool -i eth2
```

```
driver: ixgbe
```

```
version: 3.15.1
```

```
firmware-version: 0x61c10001
```

```
bus-info: 0000:04:00.0
```

Важно! При обновлении файла firmware в каталоге /lib/firmware необходимо перезагружать сервер. При этом может остаться старый firmware. Не забудьте запустить утилиту update-initramfs перед рестартом сервера.

Настройка сетевой карты

Прерывания сетевой карты

Современные 10-гигабитные сетевые адаптеры имеют несколько очередей для входящих и исходящих пакетов, которые иногда приходится вручную привязывать к ядрам процессоров.

Сервер на котором такую оптимизацию не сделали, обрабатывает всю сетевую подсистему на одном ядре. Выглядит это так:

```
# cat /proc/interrupts
```

	CPU0	CPU1	CPU2	CPU3	CPU4	CPU5	CPU6	CPU7		
0:	2097	0	0	0	0	0	0	0	IR-IO-APIC	timer
...										
66:	2072120005	0	0	0	0	0	0	0	IR-PCI-MSI	eth2-TxRx-0
67:	1562779	0	0	0	0	0	0	0	IR-PCI-MSI	eth2-TxRx-1
68:	1830725	0	0	0	0	0	0	0	IR-PCI-MSI	eth2-TxRx-2
69:	1504396	0	0	0	0	0	0	0	IR-PCI-MSI	eth2-TxRx-3
70:	5112538	0	0	0	0	0	0	0	IR-PCI-MSI	eth2-TxRx-4
71:	2229416	0	0	0	0	0	0	0	IR-PCI-MSI	eth2-TxRx-5
72:	1686551	0	0	0	0	0	0	0	IR-PCI-MSI	eth2-TxRx-6
73:	1217916	0	0	0	0	0	0	0	IR-PCI-MSI	eth2-TxRx-7
74:	2358	0	0	0	0	0	0	0	IR-PCI-MSI	eth2

Для сетевых карт Intel производитель предлагает скрипт set_irq_affinity который раскидывает очереди по ядрам. После его запуска статистика прерываний выглядит так:

```
# cat /proc/interrupts
```

	CPU0	CPU1	CPU2	CPU3	CPU4	CPU5	CPU6	CPU7		
0:	2097	0	0	0	0	0	0	0	IR-IO-APIC	timer
...										
66:	2072120005	0	0	0	0	0	0	0	IR-PCI-MSI	eth2-TxRx-0
67:	1562779	1162738082	0	0	0	0	0	0	IR-PCI-MSI	eth2-TxRx-1
68:	1830725	0	1133908105	0	0	0	0	0	IR-PCI-MSI	eth2-TxRx-2
69:	1504396	0	177620	1123678951	0	0	0	0	IR-PCI-MSI	eth2-TxRx-3
70:	5112538	0	0	0	1638450740	0	0	0	IR-PCI-MSI	eth2-TxRx-4

```

71: 2229416 130189 0 0 0 1441511712 0 0 IR-PCI-MSI eth2-TxRx-5
72: 1686551 0 0 0 0 0 1402472725 0 IR-PCI-MSI eth2-TxRx-6
73: 1217916 0 0 66145 0 0 0 1380402032 IR-PCI-MSI eth2-TxRx-7
74: 2358 0 0 0 0 0 0 0 IR-PCI-MSI eth2

```

Эта настройка становится критичной примерно в районе 3-5 Гбит/с трафика.

Соединение со свичем

При соединении сетевой карты сервера со свичем, проверьте, что с обеих сторон установлены совместимые настройки. Т.е. с обеих сторон должно быть либо auto select, либо строго одинаковая скорость и дуплекс.

Оптимизация сервера для VOD вещания

Отдельный, более детальный раздел посвящен оптимизации сервера для вещания фильмов.

Перевод процессора в режим высокой производительности

Для экономии энергии в ОС Linux по умолчанию регулятор `scaling_governor` находится в режиме энергосбережения (`powersave`). В таком случае сервер не будет использовать все свои аппаратные ресурсы. Чтобы сервер работал в режиме высокой производительности, необходимо:

Отключить регулятор `ondemand`:

```
systemctl disable ondemand
```

Перезагрузить сервер:

```
reboot
```

Проверить текущее значение `scaling_governor`:

```
cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor
```

Использование лицензионного ключа

В этом разделе описывается как использовать полученный лицензионный ключ.

Работа с лицензионным ключом

Лицензионный ключ необходимо положить в файл `/etc/flussonic/license.txt`

Ключ выглядит примерно так: `5d6b1420-4093-012e-832e-0949543365b9`

Важно не забыть, что сервер должен иметь доступ к интернету по HTTP и HTTPS.

Перенос ключа на другой сервер

Flussonic связывается с сервером лицензий через интернет, поэтому перенести лицензионный ключ на другой сервер очень легко.

Просто выключите Flussonic на первом сервере и включите на втором.

Старые версии (до 4.5.20) требовали небольшого таймаута около 4 минут между выключением и запуском, новые этого не требуют.

Скрипты на Lua

Во Flussonic Media Server есть встроенная

экспериментальная

возможность писать скрипты на lua. В качестве интерпретатора работает luerl, поэтому сообщения об ошибках могут отличаться от стандартных.

Важно! Вся описанная в этом разделе функциональность является экспериментальной и может поменяться без предупреждения.

В дополнение к стандартной библиотеке lua, Flussonic Media Server добавляет некоторые возможности к скриптам:

HTTP клиент

`http.get(url)` `http.get(url, headers)` `http.get(url, headers, body)` `http.get(url, headers, timeout)`

`http.get(url, headers, body, timeout)` Посылается HTTP GET запрос. В ответ приходит таблица с ключами: `code`, `headers`, `body`.

`http.post(url)` Посылается HTTP POST запрос. Аргументы и ответ аналогично `http.get`.

`http.qs_encode(table)` Таблица кодируется в query string.

JSON модуль

`json.encode(table)` Таблица переводится в JSON

`json.decode(text)` JSON переводится в таблицу

Flussonic API

`flussonic.config()` возвращается частичный конфиг

`flussonic.streams()` текущий список потоков

`flussonic.files()` текущий список открытых файлов

`flussonic.caches()` текущий список дисковых кешей

`flussonic.clients()` `flussonic.clients(stream)` текущий список либо всех сессий, либо сессий конкретного потока

`flussonic.log(text)` вывод в лог средствами Flussonic

`flussonic.debug(text)` отладочный вывод в лог средствами Flussonic

`flussonic.now()` текущее время UTC

`flussonic.uuid()` генерируется uuid

`table.tostring(table)` возвращает таблицу в текстовом представлении

Comet-сервер

`comet.create_channel("channel")` `comet.create_channel("channel", 100)` создается канал во внутреннем комет-сервере. Опционально указывается таймаут жизни канала

comet.send("channel", "message") посылается сообщение по каналу внутреннего комет-сервера

SWIFT клиент

auth_info = swift.auth("http://proxy-server/", "account", "password") возвращаются авторизационные данные для дальнейших обращений

swift.list_containers(auth_info) список контейнеров авторизованного аккаунта

swift.create_container(auth_info, 'videos') swift.create_container(auth_info, 'videos', {}) создание контейнера, возможно с метадатой

swift.delete_container(auth_info, 'videos') удаление контейнера

swift.list_objects(auth_info, 'videos') список объектов в контейнере

swift.create_object(auth_info, 'videos', 'file.txt', 'contents') создание объекта

swift.upload_file(auth_info, 'videos', 'remote_path.mp4', 'local_path.mp4') swift.upload_file(auth_info, 'videos', 'remote_path.mp4', 'local_path.mp4', 'local_callback_name') загрузка файла. Так же можно указать в виде строки имя локальной функции, которая будет вызываться во время загрузки для индикации загрузки на сервер

FTP клиент

ftp.list("ftp://user:password@host/path") список файлов в каталоге

ftp.upload("local.mp4", "ftp://user:password@host/path/remote.mp4") function progress(p) end

ftp.upload("local.mp4", "ftp://user:password@host/path/remote.mp4", "progress") локальный файл загружается на FTP. Опционально вызывается коллбек для статуса загрузки.

Crypto API

crypto.md5("Hi") md5 в hex виде

crypto.sha1("Hi") sha1 в hex виде

crypto.sha256("Hi") sha256 в hex виде

Авторизация с помощью Lua

Авторизационный бекенд на Lua должен представлять из себя обычный скрипт, результат которого должен возвращаться в конце оператором return. В этот скрипт передается дополнительная глобальная таблица req со следующими полями, часть из которых опциональная:

token - token из query string или автосгенерированный

ip - IP-адрес пользователя

name - имя потока/файла

referer - опционально referer плеера (адрес страницы на которой он вставлен)

Должен быть ответ return true, {user_id = 15, unique = true}, return "redirect", "http://someotherserver/path" либо return false, {code = 403}

Обработчики событий

Пример использования Lua для фильтрации и отправки внутренних событий Flussonic Media Server приведен в разделе Events API.

Веб-скрипты

Lua можно использовать для генерации веб-страниц с использованием инфраструктуры Flussonic Media Server.

Для этого надо указать в конфиге: `web_script mytest priv/myscripts;`

После этого при обращении к адресу `http://192.168.2.3:8080/mytest/counter` будет вызван lua-скрипт `priv/myscripts/web.lua`, в нём будет вызвана функция `http_handler.counter(req)`:

```
http_handler = { }
```

```
http_handler.counter = function(req)
  if not req.cookies.flusession then
    session_id = flussonic.uuid()
    headers = { }
    headers["Set-Cookie"] = "flusession="..session_id
    headers["Location"] = "/mytest/counter"
    return "http", 302, headers, "auth\n"
  else
    session_id = req.cookies.flusession
    value = flussonic.session.get(session_id, "key1")
    if not value or value == "undefined" then
      flussonic.session.set(session_id, "key1", 1)
    end
    value = flussonic.session.get(session_id, "key1")
    flussonic.session.set(session_id, "key1", value + 1)
    return "http", 200, { }, tostring(value).."\n"
  end
end
```

Теперь к этому скрипту можно обратиться:

```
$ curl -v http://localhost:8080/mytest/counter
```

```
...
```

```
< HTTP/1.1 302 Found
```

```
< Connection: keep-alive
```

```
< Server: Cowboy
```

```
< Date: Fri, 06 Mar 2015 10:20:13 GMT
```

```
< Content-Length: 5
```

```
< Location: /mytest/counter
```

```
< Set-Cookie: flusession=4a41cc61-b089-4cd5-9c4a-28402c6db525
```

```
$ curl -s http://localhost:8080/mytest/counter -H 'Cookie: flusession=4a41cc61-b089-4cd5-9c4a-28402c6db525'
```

```
1
```

```
$ curl -s http://localhost:8080/mytest/counter -H 'Cookie: flusession=4a41cc61-b089-4cd5-9c4a-28402c6db525'
```

```
2
```

```
$ curl -s http://localhost:8080/mytest/counter -H 'Cookie: flusession=4a41cc61-b089-4cd5-9c4a-28402c6db525'
```

```
3
```

```
$ curl -s http://localhost:8080/mytest/counter -H 'Cookie: flusession=4a41cc61-b089-4cd5-9c4a-28402c6db525'
```


В этом скрипте будут доступны все описанные выше функции. Так же доступна таблица req:

req.query - разобранная query string

req.headers - HTTP заголовки

req.method - HTTP метод заглавными буквами

req.body - тело HTTP заголовка. Если это тело идет в формате www-form-encoded, то его надо самостоятельно в lua разобрать с помощью http.qs_decode:

```
if req.method == "POST" then
  post = http.qs_decode(req.body)
end
```

Lua скрипт может вернуть следующие варианты ответа:

```
return "http", 200, [{"Content-Type"} = "text/plain"], "Hello, world!\n" - Можно отдать http ответ
напрямую
return "json", {key = "value"} - таблица будет упакована в JSON
return "template", {var1 = "value1", var2 = "value2"} - в этом случае Flussonic возьмет файл,
который называется
priv/cameras/list.html, который должен представлять из себя валидный DTL шаблон,
в который подставляются значения из скрипта
return "template", {headers = {}}, {var1 = "value1"} - то же самое, но с возможностью выставить
заголовки
```

Примеры

Иногда возникает задача отключить авторизацию для вторичных серверов, т.е. для нескольких IP адресов. Что бы не вносить эту логику в бекенд, можно написать собственный скрипт авторизации:

```
if req.ip == "94.95.96.97" then
  return true, {}
end
```

```
reply = http.get("http://backend/script.php")
```

```
if not reply.code == 200 then
  return false, {code = reply.code}
end
```

```
opts = {}
if reply.headers["x-userid"] then
  opts.user_id = reply.headers["x-userid"]
end
```

```
if reply.headers["x-unique"] then
  opts.unique = true
```

end

return true, opts

Безопасность Flussonic Media Server

В этом разделе мы расскажем, как ограничить доступ к панели администратора Flussonic Media Server.

Чрезвычайно важно! Если злоумышленник получит доступ к Flussonic Media Server, то он сможет прочесть и перезаписать любой файл на жестком диске.

Логин и пароль

В конфигурации Flussonic Media Server настраиваются два разных уровня доступа: `view_auth` и `edit_auth`.

`view_auth user password;` — используется для предоставления доступа на чтение к API Flussonic Media Server: получение информации о потоках, состоянии записи и т.п.

`edit_auth user password;` — используется для предоставления полного доступа администратору. С этим логином и паролем можно делать с сервером практически всё.

Ограничение по IP адресу

Для того, чтобы ограничить доступ к API по IP адресам, надо включить `white list` директивой в конфигурационном файле:

```
api_allowed_from 10.0.0.0/8 192.168.4.15;
```

Отдельный IP порт для API

Существует возможность задать для HTTP API отдельный IP порт:

```
admin_http 8090;  
admin_http 127.0.0.1:8091;  
admin_https 8092;
```

Теперь веб-интерфес HTTP API будет доступен только через заданные порты.

Это значит, что при настройке кластера в директивах `peer` и `source` для узла с включенным `admin_http(s)` надо указывать эти порты.

SSL сертификаты

Чтобы перевести веб-интерфейс администратора на HTTPS, нужно включить порт для HTTPS в конфигурации Flussonic. Откройте веб-интерфейс и укажите порт для HTTPS в `Config > SSL-tunneled protocols`, например, 443.

Можно сгенерировать собственный SSL сертификат сервера Flussonic. Для этого надо сгенерировать ключ и сертификат с паролем `flussonic` и положить их в `/etc/flussonic/` (`/etc/flussonic/flussonic.crt` и `/etc/flussonic/flussonic.key`).

Для генерации самоподписанных сертификатов Flussonic Media Server выполните по порядку команды, приведенные ниже.

Каждый раз, когда система будет запрашивать пароль для сертификата, вводите flussonic.

```
openssl genrsa -des3 -out flussonic.key 1024
```

```
openssl req -new -key flussonic.key -out flussonic.csr
```

```
mv flussonic.key flussonic.key.org
```

```
openssl rsa -in flussonic.key.org -out flussonic.key
```

```
openssl x509 -req -days 365 -in flussonic.csr -signkey flussonic.key -out flussonic.crt
```

Промежуточный и CA сертификаты будут братья из /etc/flussonic/flussonic-ca.crt.

Самое актуальное описание команд OpenSSL доступно в manpages в документации OpenSSL.

LetsEncrypt сертификаты

Компания LetsEncrypt с апреля 2016 года предлагает бесплатные SSL сертификаты сроком действия на месяц. Создание сертификата происходит в автоматическом режиме.

Мы добавили в Flussonic Media Server поддержку Letsencrypt. Как получить LetsEncrypt

Защита конфигурационного файла от изменения

Можно запретить изменение конфигурационного файла через API (т.е. веб-интерфейс). Для этого создайте файл /etc/flussonic/flussonic.conf.locked, выполнив в командной строке команду:

```
touch /etc/flussonic/flussonic.conf.locked
```

Теперь изменить настройки Flussonic через веб-интерфейс нельзя.

Запуск Flussonic от имени непривилегированного пользователя

Чтобы запустить Flussonic Media Server под обычным пользователем, выполните следующие настройки:

```
adduser flussonic --home /var/lib/flussonic --disabled-password
```

```
chown -R flussonic /etc/flussonic/
```

```
chown -R flussonic /var/lib/flussonic/
```

```
echo flussonic > /etc/flussonic/run_as
```

```
chown root /etc/flussonic/run_as
```

```
chmod 0644 /etc/flussonic/run_as
```

```
chown -R flussonic /var/run/flussonic /var/log/flussonic /opt/flussonic/.erlang.cookie
```

```
setcap cap_net_bind_service=+ep /opt/flussonic/lib/erlang/erts-*/bin/beam.smp
```

Теперь после перезапуска сервер Flussonic Media Server будет работать с правами пользователя flussonic.

Чтобы снова запускать Flussonic Media Server под суперпользователем, удалите файл

/etc/flussonic/run_as.

Активация Flussonic через SOCKS5 прокси

Flussonic может использовать SOCKS5 прокси сервер для общения с сервером лицензий. Чтобы включить его, используйте возможности systemd override:

```
# systemctl edit flussonic
```

Эта команда откроет текстовый редактор (обычно это nano). Затем введите:

```
[Service]
```

```
Environment=PROXY="socks5://172.20.10.1:1080"
```

Нажмите Ctrl-X, затем Y, чтобы сохранить и выйти.

Перезапустите Flussonic:

```
# /etc/init.d/flussonic restart
```

Теперь Flussonic будет использовать указанный прокси-сервер.

Защита видео от просмотра администратором

По умолчанию пользователи с правами администратора Flussonic могут проигрывать любой поток в веб-интерфейсе администратора. Для этого используется специальный авторизационный токен администратора.

Возможно, вы захотите запретить просмотр некоторых потоков администратором — тех потоков, для которых нужна авторизация пользователя.

Чтобы запретить администратору Flussonic проигрывать в веб-интерфейсе потоки, защищенные авторизацией:

Отредактируйте unit файл сервиса Flussonic (/lib/systemd/system/flussonic.service). Следует делать это, используя systemd override:

```
# systemctl edit flussonic
```

Эта команда откроет текстовый редактор (обычно nano).

Добавьте строки:

```
[Service]
```

```
Environment=ADMIN_VIEW_DISABLE=true
```

Нажмите Ctrl-X, затем Y, чтобы сохранить и выйти.

Перезапустите Flussonic:

```
# /etc/init.d/flussonic restart
```

Теперь, если поток требует авторизации, плеер в веб-интерфейсе вернет ошибку 403 при попытке проиграть поток с токеном администратора.

Потоки без настроенной авторизации будут воспроизводиться как обычно.

Защита файловой системы от доступа из UI

В веб-интерфейсе пользователь (администратор Flussonic) указывает расположение директорий

для хранения DVR, VOD и кэш. Вы можете ограничить администратора, указав директорию, выше которых Flussonic не разрешит размещать VOD, DVR и кэш. Например, важно сделать директорию /root недоступной.

Настройте переменную окружения FLUSSONIC_DATAPATH и укажите самую верхнюю директорию, в которой допускается создавать VOD локации, DVR и кэш.

Важно! Чтобы Flussonic запустился с новыми настройками, в текущей конфигурации не должно быть путей к директориям, находящимся выше указанный в переменной FLUSSONIC_DATAPATH.

Для добавления FLUSSONIC_DATAPATH можно использовать возможности systemd override:

```
# systemctl edit flussonic
```

Эта команда откроет текстовый редактор (обычно это nano). В редакторе введите, например, такие директории:

```
[Service]
```

```
Environment=FLUSSONIC_DATAPATH=/storage:/mount:/copy
```

Нажмите Ctrl-X, затем Y, и Enter чтобы сохранить и выйти.

Перезапустите Flussonic:

```
# /etc/init.d/flussonic restart
```

Теперь в настройках можно будет использовать только в /storage, в /mount и /copy и вложенных директориях.

Let's Encrypt и Flussonic Media Server

Сервис Let's Encrypt позволяет получать сертификаты для настройки HTTPS в автоматическом режиме.

Flussonic Media Server имеет встроенную поддержку Let's Encrypt, поэтому не требуется установка дополнительных пакетов и ручной настройки веб-сервера.

Достаточно зайти в админку и указать порт для HTTPS, нажать на кнопку «Let's Encrypt».

Дальше Flussonic Media Server сделает все сам — получит сертификат и будет обновлять его.

Вам не нужно беспокоиться о сроке жизни сертификата и редактировать конфигурацию.

HTTPS нужен, чтобы:

- никто не мог перехватить управление сервером, узнать ваш пароль или ссылки на потоки;

- защитить видео с камер наблюдения;

- вставить ссылку на сайт, работающий по https (иначе браузеры будут ругаться на незащищенный контент).

Ниже более подробное описание процесса настройки и механизма работы сервиса Let's Encrypt.

Let's Encrypt: как это работает

Подробное описание на официальном сайте: <https://letsencrypt.org/how-it-works/>.

Чтобы Let's Encrypt выдал вам валидный сертификат, нужно доказать, что вы владеете доменом. Когда вы нажимаете «Issue by Letsencrypt» в административной панели, Flussonic Media Server сообщает доменное имя, для которого требуется сертификат. В ответ получает ключ, который нужно будет отдать, когда проверяющий бот обратится к серверу по HTTP (именно на 80-й порт) по адресу `http://your-domain.com/.well-known`.

Проверяющий бот обращается к вашему домену, поэтому домен должен быть делегирован, а DNS записи настроены на IP-адрес, где работает Flussonic Media Server. Бот подтверждает владение доменом, а Flussonic Media Server сохраняет сертификат.

Чтобы продлить сертификат, придется повторить проверку, а значит Flussonic Media Server всегда должен слушать порт `http 80`; . Перенести проверку на другой порт не получится — такие правила у Let's Encrypt. Продление происходит автоматически, когда срок действия сертификата подходит к концу, но можно и вручную обновить сертификат через административную панель Flussonic Media Server.

Настройка

Зайдите в административную панель Flussonic Media Server с помощью доменного имени, а не IP-адреса (например, `http://your-domain.com/admin`)

Перейдите во вкладку «Config» и в SSL-tunneled protocols в качестве HTTPS ports укажите 443.

После этого в панели появится кнопка «Issue by LetsEncrypt», запускающая процесс получения сертификата.

Нажмите на нее и дождитесь, когда появится срок действия сертификата (как правило, это занимает до 10 секунд).

Вот так выглядит меню, когда сертификат не получен:

А вот после получение сертификата:

Сохраните настройки нажатием на «Save». Flussonic Media Server перекинет ваш браузер на <https://> — теперь можно предоставлять сервис по HTTPS.

Миграция

Важно! При миграции настроек Flussonic Media Server с сервера на сервер исполняемые файлы и установленные библиотеки переносить нельзя, используйте пакетный менеджер для установки на новом сервере. Переносится только конфигурация и лицензия.

Список файлов для переноса:

`/etc/flussonic/flussonic.conf` — основной файл конфигурации.

`/etc/flussonic/license.txt` — лицензия.

`/opt/flussonic/priv/iptv.db` — файл базы данных IPTV плагина.

Копирование файлов конфигурации возможно только с root-правами.

Способы перенести файлы:

Перенос конфигурации с помощью SCP

Перенос конфигурации с помощью USB-носителя

Перенос конфигурации с помощью SCP

SCP (Secure CoPy) — программа для удаленного копирования файлов по сети между хостами. Она использует SSH для передачи данных, в том числе аутентификацию и меры безопасности, которые реализованы для SSH.

Для копирования файла с одного удаленного сервера «remote.host1» на другой удаленный сервер «remote.host2» необходимо выполнить команду вида:

```
$ scp user@remote.host1:/directory/file.txt user@remote.host2:/some/directory/
```

Установите Flussonic Media Server на новый сервер:

```
curl -sSf https://flussonic.com/public/install.sh | sh
```

После этого перенесите на новый сервер файлы конфигурации:

```
scp user@remote.host1:/etc/flussonic/flussonic.conf user@remote.host2:/etc/flussonic/
```

```
scp user@remote.host1:/etc/flussonic/license.txt user@remote.host2:/etc/flussonic/
```

```
scp user@remote.host1:/opt/flussonic/priv/iptv.db user@remote.host2:/opt/flussonic/priv/
```

Запустите Flussonic Media Server:

```
/etc/init.d/flussonic start
```

Готово!

Перенос конфигурации с помощью USB-носителя

Если вы хотите перенести файлы конфигурации с помощью какого-либо USB-носителя, то воспользуйтесь следующей инструкцией.

Монтирование USB

Создайте директорию, в которую будем монтировать:

```
mkdir -p /mnt/usb
```

Вставьте носитель в USB порт и узнайте имя устройства:

```
fdisk -l
```

Результатом этой команды будет:

```
Disk /dev/sdb: 4008 MB, 4008706048 bytes
```

```
118 heads, 53 sectors/track, 1251 cylinders, total 7829504 sectors
```

```
Units = sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disk identifier: 0x74a37a4d
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1	*	63	7829503	3914720+	b	W95 FAT32

Здесь имя устройства: /dev/sdb1.

Смонтируйте носитель:

```
mount /dev/sdb1 /mnt/usb
```

Проверьте подключение:

```
mount
```

Копирование конфигурации

```
cp /etc/flussonic/flussonic.conf /mnt/usb/flussonic.conf
```

```
cp /etc/flussonic/license.txt /mnt/usb/license.txt
```

```
cp /opt/flussonic/priv/iptv.db /mnt/usb/iptv.db
```

После копирования не забудьте отмонтировать накопитель:

```
sudo umount /dev/sdb1
```

Установка конфигурации на новый сервер

Установите Flussonic Media Server на новый сервер:

```
curl -sSf https://flussonic.com/public/install.sh | sh
```

Создайте директорию, в которую будем монтировать USB-носитель:

```
mkdir -p /mnt/usb
```

Вставьте носитель в USB порт и узнайте имя устройства:

```
fdisk -l
```

Смонтируйте:

```
mount /dev/sdb1 /mnt/usb
```

Перенесите файлы конфигурации:

```
cp /mnt/usb/flussonic.conf /etc/flussonic/flussonic.conf
```

```
cp /mnt/usb/license.txt /etc/flussonic/license.txt
```

```
cp /mnt/usb/iptv.db /opt/flussonic/priv/iptv.db
```

Запустите Flussonic Media Server:

/etc/init.d/flussonic start

Готово!

Захват спутникового видео

Для захвата видео со спутника используется так называемое IRD (Integrated Receiver-Decoder) оборудование и головные станции. Flussinic Media Server умеет принимать видео по IP протоколам с любых IRD устройств и систем.

Рассмотрим принципы захвата видео со спутников и вопросы выбора оборудования для этой цели.

Содержание:

- Долгота
- Настройка ресивера
- Дескремблинг
- Выбор оборудования
- Получение каналов
- Итоги

Долгота

Телевизионные спутники висят на геостационарной орбите Земли над экватором. Их положение над Землей постоянное, поэтому спутники идентифицируются долготой, над которой они висят. Так, например, спутник ABS 2 вращается вокруг Земли, постоянно находясь над 75° восточной долготы.

Геостационарное размещение делает очень простым монтаж спутниковых тарелок: достаточно направить на нужную точку и не нужно поворачивать тарелку.

Поскольку спутник находится над конкретной долготой, то, как правило, разные спутники используются для вещания определенного набора каналов. Например, спутник Appstar-7 76.5°E наполнен индийскими каналами, а спутник Galaxy-17 91.0°W больше используется для вещания каналов для США.

Важно отметить, что у некоторых спутников есть несколько лучей, т.е. зон максимального приёма. Каналы в лучах могут отличаться. Например, один и тот же спутник может на Россию вещать один набор каналов, а на Тайланд другой.

Настройка ресивера

С одного спутника в одном луче уходит множество транспондеров. Транспондер — это один цифровой канал, для приёма которого надо настроить карту захвата на определенную частоту и поляризацию.

Т.е. с одного спутника одновременно вещается множество частот в нескольких поляризациях.

Поляризация бывает правой/левой или горизонтальной/вертикальной. Бытовые спутниковые антенны (точнее головки этих антенн или конвертеры) умеют принимать сразу обе поляризации на один выходящий кабель, но делают это плохо. Профессиональные конвертеры принимают все поляризации, но на разные выходы.

Такое разделение обусловлено тем, что для переключения приёма поляризации плата захвата

подает напряжение 13 или 18 вольт. Ниже таблица соответствия поляризации и напряжения:

Напряжение (В)	13-14	17-18
----------------	-------	-------

Линейная поляризация	вертикальная	горизонтальная
Круговая поляризация	правая	левая

В некоторых ресиверах указывается напряжение, в некоторых указывается поляризация. По сути всё сводится к подаче напряжения.

Если простым сплиттером соединить две платы захвата и на одной подать 18 вольт, а на другой 13, то вторая не получит сигнал.

Некоторые ресиверы умеют отключать напряжение. В этом случае они должны быть сплиттером соединены с другим ресивером/картой захвата, который всё таки подает напряжение.

Диапазоны частот со спутника разделяются условно на верхний и нижний диапазон. Граница проходит примерно на 11700 МГц.

При захвате частот ниже 11700 обычно выставляется частота гетеродина, она же LNB Frequency в 9750 МГц. При захвате частот выше 11700 настраивается частота гетеродина в 10600 МГц

После того, как правильно скоммутировали провода, не напутав с напряжением, настроили нужную частоту на приёмнике и частоту гетеродина и приёмник автоматически выбрал FEC (количество битов для контроля ошибки) и модуляцию (QPSK, 8PSK и т.п.), то ресивер начинает получать битовый поток, т.е. транспондер.

Транспондер представляет из себя MPEG-TS поток и позволяет в один поток упаковывать много разных телеканалов с разными языковыми дорожками и субтитрами.

Бытовой спутниковый приёмник позволяет забрать только один канал из транспондера, но профессиональные приёмники и DVB PCI платы захвата позволяют снимать все каналы с транспондера. Ниже будет подробнее описана структура MPEG-TS потока в транспондере.

Дескремблинг

Большинство каналов на спутнике вещаются зашифрованными. Шифрование используется для управления доступа к каналам различных пользователей: кто проплатил следующий месяц, тот и смотрит каналы.

Процедура расшифрования спутникового канала называется дескремблингом, а сам зашифрованный канал называют скремблированным.

Механизм управления доступом к спутниковым телеканалам называется Conditional Access (CA), в русской терминологии используется термин «условный доступ».

На сегодняшний день используются различные схемы шифрования, но в основном все современные схемы работают примерно следующим образом:

- абонент получает карту доступа (похожа на большую, необрезанную SIM-карту с таким же чипом);

- на карте доступа есть приватный ключ;

- раз в месяц меняется ключ пакета;

через спутник рассылается для каждого абонента ключ пакета, зашифрованный публичным ключом карты;

карта доступа запоминает ключ пакета;

раз в минуту на каждом канале меняется ключ потока, который зашифрован ключом пакета;

если пользователю пришел ключ пакета и он смог его расшифровать, то он может получить доступ к каналу.

У этой схемы есть вариации и усложнения, но концептуально схема именно такая. Если не заплатить денег спутниковому оператору, то в конце месяца он не разошлет обновленный ключ и карта не сможет расшифровывать.

С технической точки зрения с дескремблированием сложилась грустная ситуация. Спутниковые операторы и пираты ведут многолетнюю и безуспешную борьбу друг с другом, в которой страдают операторы связи.

Когда обычный абонент покупает спутниковую тарелку для дома, он получает карту доступа и бытовой спутниковый приёмник, который имеет чип для дескремблирования одного канала по той схеме шифрования, которая выбрана у этого оператора. Условно говоря, приёмник для НТВ+ не подойдет к Континент-ТВ.

Оператор связи не может использовать 200 бытовых приёмников чисто технически, поэтому используются профессиональные приёмники, которые снимают не по одному каналу с транспондера, а все каналы. Однако официальный метод дескремблирования, который предлагают операторы связи, подразумевает использование специальных СА модулей. Это плата, похожая на РСМCIA модуль, в который втыкается карта доступа.

СА модуль самостоятельно дескремблирует каналы. Для этого он забирает каналы с головной станции, дескремблирует и отдает обратно.

Проблема в том, что даже для профессионального СА модуля дескремблирование 8 каналов — это предельная нагрузка. Учитывая, что на многих транспондерах сегодня бывает до 30 каналов, получается, что один и тот же транспондер надо захватывать через сплиттер несколько раз, используя дорогостоящие карты захвата или головные станции.

Детальное описание процесса дескремблирования на головной станции или с помощью компьютера находится за рамками этого описания, особенно учитывая, что допустимые схемы дескремблирования должны быть согласованы с поставщиком контента, чтобы не нарушить уголовный кодекс.

Выбор оборудования

Традиционный способ захвата спутникового эфира — использование так называемых головных станций.

Головная станция — это специализированный спутниковый приёмник, умеющий захватывать больше одного канала (до сотен каналов).

Более дорогие головные станции, такие как WIS Compact Headend, предлагают более плотную компоновку по сравнению с недорогими станциями:

Вместо 1-2 транспондеров в 1U корпусе можно захватывать вплоть до 24. Правда дескремблировать их все не получится, потому что в этом случае количество транспондеров сократится до 12 (нужно место под CI модули), а профессиональная головная станция не может дескремблировать больше 8-10 каналов с одного транспондера, потому что это по большому счёту дорогой, но слабый компьютер.

Альтернативный вариант — использование компьютера для захвата со спутника.

В обычный PC помещается до 7 подобных плат (главное подобрать материнскую плату). Разумнее всего брать либо с 4 выходами, либо с 2 но с CI модулем.

При покупке подобных плат очень важно свериться с поддержкой их в Linux на сайте проекта LinuxTV, кому в конце концов может понадобится плата захвата, которая поддерживается только Windows?

Flussonic сможет работать с видео независимо от того, на каком оборудовании оно захвачено. Поэтому при выборе оборудования достаточно учесть различия в обслуживании, ориентируясь на выделенный бюджет и технические предпочтения.

При построении головной станции можно выбрать отдельное полностью аппаратное решение либо решение, построенное на x86 платформе.

Аппаратные решения отличаются большей ценой и несколько меньшей гибкостью. Для настройки таких решений, как правило, используется веб-интерфейс, что удобно, но может замедлить решение разных проблем. Взамен вендоры таких решений гарантируют стабильную емкость, длительную непрерывную работу и превосходную поддержку.

Программные решения, которые вы будете использовать с платами захвата под PC, в свою очередь обладают непревзойденной гибкостью в настройке и работе. Настройки можно задавать через конфигурационные файлы, а Linux предоставляет более широкие возможности для отладки проблем, чем закрытые аппаратные решения.

Получение каналов

Как было выше сказано, транспондер — это MPEG-TS поток. Транспортный контейнер MPEG-TS позволяет в один поток байт упаковывать много одновременно идущих потоков, предоставляя стандартизованный способ для выбора нужного под-потока.

Один телеканал называется программой (program). MPEG-TS в котором идет только одна программа называется SPTS — single program transport stream. Со спутника идет MPEG-TS в котором много программ, это называется MPTS — multiple program transport stream.

MPTS удобен для передачи в носителях типа спутника или кабеля, когда полоса фиксированная и для выравнивания трафика поток даже приходится добивать ненужными байтами. SPTS удобен при передаче по IP, когда весь огромный транспондер клиенту не нужен, а нужен только один канал.

Процесс смешивания нескольких SPTS в MPTS называется мультиплексированием и как правило производится перед подачей потока на спутник или кабель. Процесс расщепления MPTS на несколько SPTS называется демультиплексированием и происходит при приеме со спутника.

По IP MPTS передают очень редко, например для транзита со спутника в кабель.

Сам MPEG-TS представляет из себя последовательность пакетов по 188 байт. Первый байт всегда 0x47 и используется для статистически достоверной синхронизации в потоке.

В следующих трех байтах закодирован 13-битный номер потока внутри MPEG-TS. Этот номер называется Pid и поэтому в профессиональном жаргоне зачастую под-поток так же называют пидом по его номеру.

Есть несколько стандартных номеров Pid, которые зарезервированы под служебные нужды. Условно говоря, это все номера до 32.

В потоке с Pid 0 идет информация о существующих программах в этом MPEG-TS потоке. Эта информация упакована в PAT — program adaptation table. PAT — это один из вариантов PSI информации. PSI или PSI таблицы — это метainформация, идущая в MPEG-TS потоке и

необходимая либо для получения доступа к аудио/видео, либо для получения дополнительной информации, как например, расписание передач или информация о каналах на других транспондерах.

Важно понимать, что все PSI таблицы проектировались для случаев, когда кроме как со спутника, никаких данных ресивер больше не получает. Поэтому большинство PSI таблиц для IPTV OTT сервиса бессмысленны: расписание там зачастую паршивое, а информация о других транспондерах вообще бессмысленна.

В PAT записано, какие номера программ (pnr, program number, service id) идут на каких пидлах. На этих пидлах будут идти не аудио/видео потоки, а PMT — program mapping table. В PMT уже будет написано на каком пиде идет видео поток и на каких пидлах идут разные языки аудио.

При настройке демультимплексирования можно настраивать по пидлам, а можно по pnr. Второе более предпочтительно, потому что пиды на спутнике могут перенастроиться без предупреждения, а номера программ как правило не меняются.

Итоги

Процесс захвата видео со спутника выглядит следующим образом:

- администратор настраивает захват на головной станции или программе с нужного входа, на нужной частоте и с нужной поляризацией;

- поток дескреблируется либо через CA модуль, вставленный в CI слот, либо программно с использованием Phoenix card reader;

- поток демультимплексируется из MPTS в несколько разных SPTS, согласно настройкам (скорее всего по pnr);

- несколько отдельных SPTS вещаются в сеть по мультикасту.

В классическом IPTV на этом всё и заканчивается, клиенты получают свой мультикаст через каскад роутеров, общающихся по PIM протоколу, но в нашем случае всё только начинается, потому что дальше результирующее видео надо транскодировать

Транскодирование

Со спутника видео передается либо в кодеке MPEG-2, либо в H.264 (он же AVC или MPEG-4 part10). Как правило для простоты MPEG-4 part 10 сокращают до MPEG-4, но тут важно не спутать с MPEG-4 part 2, который совершенно никак не совместим и не похож на H.264 и использовался в старых IP камерах.

Аудио передается в MPEG audio layer 2 (сокращенное mp2), либо в ac3 (a/52).

Причём важно понимать, что сегодня H264 как правило сжимается с intra-refresh, т.е. в видео потоке нет опорных кадров (IDR или keyframe). Такой метод сжатия позволяет сгладить скачки битрейта.

В результате ни один из передаваемых со спутника вариантов аудио или видео не проигрывается на айфоне. В браузере проигрывается только H264.

При передаче через интернет как правило можно смело сжимать видео из mpeg2 в h264 с трехкратным снижением трафика.

При передаче HD каналов через интернет сегодня приходится сжимать поток в несколько разных качеств: от HD с максимальным качеством до стандартного SD для компенсации перегруженных каналов.

В итоге видео со спутника для предоставления качественного ОТТ сервиса надо транскодировать в другие кодеки и качества.

Важно не путать транскодирование с перепакровкой. Транскодирование — крайне ресурсоёмкая операция, включающая в себя:

- распаковку потока до кодированного видео/аудио
- декодирование до сырого видео/аудио
- изменение размеров и прочих параметров
- кодирование обратно
- упаковка в транспорт для потока

Упаковка и распаковка относительно легкие операции, стриминговый сервер может обрабатывать до 1000 каналов на одном компьютере. Транскодировать на одном компьютере можно от 1 до 30 каналов в зависимости от размера и мощности компьютера.

Для транскодирования можно использовать специализированные выделенные устройства, центральный процессор или видеоплату: внешнюю или встроенную в процессор.

Специализированные устройства мы рассматривать не будем, потому что в своей массе это либо компьютер с какой-то программой, либо крайне дорогостоящее и очень специализированное оборудование, или же либо попросту необоснованно дорогое устройство, реализуемое исключительно за счёт маркетинговых усилий компании производителя и не позволяющее достигнуть сколь-либо значимых результатов.

H.264

Для обработки видео на CPU существует несколько разных программ, но по большому счёту на сегодняшний день существует лишь две библиотеки, которые имеет смысл использовать для сжатия в кодек H.264 на CPU: это бесплатная libx264 и платная MainConcept. Всё остальное либо

хуже, либо сильно хуже, причём как по выходному результату, так и по использованию ресурсов.

Практика работы с MainConcept в этой статье рассматриваться не будет, будет упомянута только libx264

Кодек H.264 является стандартом де-факто на сегодняшний день для видео, потому что он поддерживается во всех современных устройствах, за исключением разве что некоторых устройств от Google.

Альтернатив ему практически нет. Сегодня появился и развивается H.265, у него уже есть большая поддержка, но пока что работа с ним — это инвестиции в будущее.

Кодеки от Google: VP8 и VP9 являются больше желанием гугла перетянуть одеяло на себя, нежели чем-то реально полезным. Результирующее качество хуже, поддержки аппаратного декодирования нет, а следовательно растёт цена устройства.

При кодировании видео надо понимать, что приходится балансировать между такими параметрами:

- задержка внутри энкодера в кадрах
- использование CPU (сколько миллисекунд требуется на сжатие одного кадра)
- выходное качество картинки (насколько пиксельная и какие цвета)
- выходной битрейт

Для всех видов эфира абсолютно критичным является использование CPU. Если настройки энкодера требуют полной загрузки CPU или больше, то видео не будет успевать кодироваться в реальном времени и следовательно потоковость видео пропадет.

Для VOD такого жесткого ограничения нет и фильм длиной в час вполне можно кодировать три часа, если хочется понизить битрейт. При этом для эфирного видео обычно всё таки стараются использовать не всю мощность процессора, что бы обрабатывать на одном компьютере не 4 канала, а 10.

Что касается задержки внутри энкодера, то она критична для видеоконференций, но совершенно не критична для IPTV. Даже 5 секунд задержки при вещании телевидения не меняют качество сервиса.

У битрейта и качества связь достаточно четкая: чем больше информации о картинке мы передаем, тем лучше она будет отображаться. Повысить качество картинки, снизив битрейт, как правило можно за счёт выбора более результативных инструментов компрессии, которые требуют большей задержки и большего количества тактов.

Понимание этой сложной взаимосвязи нужно для того, что бы лучше воспринимать заверения о том, что «наш энкодер самый лучший энкодер в мире». Сравнивать приходится минимум по 4-м параметрам, но в итоге всё сводится к тому: сколько денег стоит разово и в месяц транскодирование одного канала с желаемым качеством и выходным битрейтом.

Flussonic Media Server для транскодирования

Отдельным пакетом к Flussonic Media Server идет транскодер.

Flussonic Media Server может декодировать видео из UDP/HTTP MPEG-TS, RTMP источников и кодировать его в несколько качеств и размеров.

Эта возможность становится нужна, когда возникает необходимость показывать видео не только на приставках, но и на планшетах: там выбор доступных кодеков существенно меньше, чем на приставке.

Важно отметить, что для того, что бы видео игралось на айфоне, надо даже H264 со спутника транскодировать, потому что как правило на спутнике для плавного битрейта используется intra-refresh режим кодирования, создающий видео, которое не играется на айфоне.

Flussonic Media Server удобнее чем VLC или другие варианты для организации транскодирования, потому что управляется одним конфигурационным файлом и автоматически следит за состоянием транскодирования. VLC же требует написания большого количества мониторинговых скриптов для отслеживания состояния транскодирования.

Следующая важная возможность Flussonic Media Server для транскодирования — автоматическая перебалансировка потоков при падении одного из серверов. Если один из 20 транскодеров ночью сломается, то остальные транскодеры можно настроить на автоматический захват потоков для транскодирования, причём стример сам заберет потоки с резервных транскодеров.

Middleware в IPTV OTT

В старом аналоговом эфирном телевидении пользователь должен был настраивать самостоятельно все нужные каналы: первый канал на 1-ю кнопку, второй канал на 2-ю. Пока каналов меньше 10 такой вариант всех устраивал. Никакого контроля доступа или детализированного учета: что чего смотрит в древнем телевидении не было.

С переходом на IPTV технология настройки не сильно отличалась: в приставку при заливке прошивки добавляется плейлист, т.е. список каналов с их multicast UDP адресами. Контроль доступа осуществляется либо с помощью шифрования (CAS), либо с помощью сетевых методов, например авторизация IGMP запросов, которая возможна только в простой локальной сети.

При этом все функции реализуются на приставке. Например, такая услуга как PVR (personal video recorder) осуществляется на приставке: пользователь заранее заказывает запись передачи и в нужный момент приставка сама начинает на свой жесткий диск записывать нужную передачу.

С развитием IPTV пользователям начали предлагать такие новые сервисы, как:

- EPG (electronic program guide), т.е. расписание передач.

- Организация каналов по платным пакетам.

- Каталогизация каналов (по жанрам).

- Родительский контроль (не показывать детям эротику).

- Просмотр записанных передач.

- Сопутствующие сервисы типа прогноза погоды или курса валют.

- Подписка на платный пакет с пульта.

- Предоставление VOD, т.е. просмотр фильмов.

Важно понимать, что большинство решений в сфере IPTV разрабатывалось при постоянных мыслях о том, как бы это реализовать в условиях спутникового вещания, т.е. когда связи у приставки с каким-то центральным сервером нет. Из-за этого в транспортных протоколах, используемых в телевидении есть очень много деталей бизнес-логики, которые были нужны раньше, но становятся не очень актуальны сегодня.

С развитием разнообразия услуг в IPTV возникло понимание, что развивать услуги за счёт усложнения приставки не очень удобно или не всегда возможно, потому что когда программная часть услуги реализована на приставке, её обновление и поддержка требуют небезопасной процедуры обновления прошивки на приставке.

Для упрощения ввода в строй новых услуг и управления ими, а так же для реализации услуг, невозможных в классическом телевидении, в IPTV появилось звено под названием Middleware.

За этим страшным термином прячется обычный веб-сайт (отдающий HTML, javascript или отвечающий на HTTP API запросы), на который приставки заходят либо обычным веб-браузером, либо чем-то экзотическим типа SVG браузера. На приставке размещается доработанный веб-браузер (обычно Opera или webkit), который умеет проигрывать все варианты видео, доступных приставке (десктопные браузеры обычно не умеют и 5% от видео-возможностей приставки) и умеет работать с пультом, превращая нажатия кнопок пульта в Javascript-события.

Важный момент: на приставках не используется Java (за исключением андроидных приставок, которые по ряду причин очень непопулярны). Обычно люди путают Java и Javascript, не надо так делать.

До сих пор существует дискуссия среди специалистов, что лучше: веб-браузер или специализированное приложение на устройстве. Этот выбор совершенно аналогичен выбору технологии на мобильных устройствах: делать HTML приложение или писать на С.

Многие современные middleware предлагают оба варианта для покрытия максимального количества устройств. Так, например, для приставок Amino (с браузером opera), Mag250, tvip и т.п. (с браузером на базе webkit) будет отдаваться HTML с интерфейсом.

Обычно middleware практически не взаимодействует с видеопотоком, потому что лишь предоставляет приставкам урлы для просмотра каналов и фильмов: либо мультикаст урлы, либо уникаст. Иногда в Middleware реализован механизм мониторинга каналов, что бы не показывать клиентам канал или явно сообщать, что просмотр канала невозможен из-за аварии.

Ниже мы рассмотрим, каким образом происходит интеграция Flussonic Media Server и Middleware для улучшения качества сервисов пользователям.

Авторизация пользователей

В IPTV ограничение доступа к видео используется для того, что бы:

- управлять пакетами каналов. Не подписался на футбол — смотри что попало
- усложнять задачу воровства контента неавторизованными пользователями
- усложнять задачу несанкционированной записи передач

Тут смешиваются две системы: CAS и DRM. CAS (conditional access system) — это механизм технического ограничения доступа к контенту. DRM — механизм для запутывания пользователя, что бы он никак не добрался до расшифрованного видео.

CAS системы работают хорошо и исправно. DRM системы в силу своей природы ненадежны, глючат и создают массу проблем всем, кроме их продавцов.

В Flussonic Media Server реализована CAS с помощью авторизации доступа к потокам и файлам.

Интеграция с Middleware выглядит следующим образом:

Middleware при формировании HTML страницы или ответа на API, отдает ссылку на HLS (или HTTP MPEG-TS) поток с уникальным ключом.

Приставка получает этот уникальный урл для просмотра, обращается с ним к Flussonic Media Server.

Flussonic Media Server при первом запросе обращается обратно к Middleware с вопросом: можно ли этой приставке смотреть этот канал по этому адресу.

Middleware проверяет, не подсмотрен ли этот адрес (проверяется IP клиента, user agent, протокол и время) и разрешает или запрещает.

Если злоумышленник подсмотрит в сети адрес, то он им не сможет воспользоваться, потому что не совпадет какой-нибудь из параметров и Middleware скажет стримеру не отдавать ему видео.

Работа с EPG

EPG, он же электронный телегид, оно же расписание передач. Обычно представляет из себя источник головной боли, если хочется точно попадать в передачу, потому что практически никто на российском рынке не предоставляет точного расписания передач.

Телеканалы не особо следят за точностью программы передач (погрешность до 15-20 минут) и постфактум точное время начала и конца передач не сообщают.

EPG можно получать со спутника в MPEG-TS потоке, но там информация достаточно ограничена, а можно забирать из интернета у поставщиков программы передач, таких как teleguide.info

Частый формат для программы передач — XMLTV.

Традиционно приставки повторяют функциональность древних видеомагнитофонов: пользователь заранее заказывает желаемую передачу и приставка записывает её. И так же традиционно из-за ошибок в расписании передач, пользователь записывает хвост предыдущей рекламы, 20 минут рекламы, потом нужную ему передачу, обрезанную в самом конце, потому что расписание поехало.

Flussonic Media Server предлагает совершенно другой подход к записи передач. Не нужно издеваться над пользователем и заставлять его заранее вспоминать о нужной передаче. Middleware должна предоставлять возможность пользователю посмотреть передачи, которые уже прошли и сформировать правильную ссылку к Flussonic Media Server для просмотра уже записанной передачи.

Тут есть два механизма работы: для уже прошедшей передачи и для ещё идущей.

Если передача уже прошла и закончилась, то Middleware на основании EPG формирует ссылку для просмотра из архива (которая так же проходит через механизм авторизации).

Пользователь получает возможность посмотреть записанную передачу, как обычный файл.

Например, если передача началась в 18:15 по Москве (14:15 UTC) 27 августа и длилась час, то Middleware должен при выборе передачи в списке прошедших сформировать url вида <http://streamer/ort/index-1409148900-3600.m3u8>

Если передача сейчас всё ещё идет, то Middleware может сформировать специальный url к архиву, позволяющий отматывать назад прямой эфир на начало передачи. Данная функциональность, к сожалению, поддерживается далеко не на всех устройствах и приставках, но тем не менее она существует.

Url для такой незакончившейся передачи будет выглядеть <http://streamer/ort/index-1409148900-now.m3u8>

Важный момент здесь в том, что информация о записанных передачах и об их времени хранится в middleware, а Flussonic предоставляет доступ к своему архиву как к бесконечной ленте.

В документации более подробно описана работа с архивом DVR

В Flussonic Media Server есть поддержка и других вариантов доступа к записанным передачам. Это:

проигрывание архива по HTTP MPEG-TS с определенного момента:

http://streamer/ort/timeshift_abs-1409148900.ts

проигрывание архива в режиме потока по HLS с определенного момента:

http://streamer/ort/timeshift_abs-1409148900.m3u8

Так же Middleware может обратиться к Flussonic Media Server к API о состоянии записи потока, чтобы показать в интерфейсе передачи, которые можно посмотреть и которые нельзя записать.

Таймшифт

Под термином таймшифт подразумевают две разных функции: возможность отмотки назад прямого эфира и постоянный сдвиг эфира в другую временную зону.

Эта услуга нужна, когда видео захватывается в одном часовом поясе, а показывать хочется в другом для того, что бы, например, пользователи в США видели утреннюю передачу в 9 утра, а не в час ночи.

Flussonic предлагает два варианта таймшифта: запуск постоянного потока, который отстает на фиксированное время от эфира и предоставление ссылок на просмотр архива в режиме потока.

Разница между ними в количестве чтения с диска. Если к каналу редко обращаются, логичнее воспользоваться вторым вариантом. Если канал смотрят со сдвигом часто, то нужно запускать постоянный поток.

Задача Middleware здесь — знать как настроен канал и выдавать ссылки либо к смещенному каналу, либо индивидуальные ссылки на просмотр архива вида:

http://streamer/ort/timeshift_rel/7200 — проигрывание архива по HTTP MPEG-TS с отставанием в 2 часа

http://streamer/ort/timeshift_rel-7200.m3u8 — проигрывание архива по HLS с отставанием в 2 часа

Интеграция Flussonic с Middleware

На сегодняшний день поддержка возможностей Flussonic Media Server есть в следующих Middleware:

iptvportal, они же распространяют Flussonic Media Server в составе пакета

Stalker

CloudWare

С другими Middleware можно провести интеграцию по вашему запросу.

На стороне Flussonic Media Server реализованы всё, что нужно для интеграции с Middleware. Просите вашего поставщика Middleware проверить совместимость с Flussonic Media Server самостоятельно. Также вы можете протестировать следующие Middleware:

www.magoware.tv

www.abvtc.com

Получение EPG из MPEG-TS потоков

Об экспорте EPG

EPG (электронный телегид) это важный компонент услуг цифрового телевидения. Есть разные способы предоставить EPG пользователям. Например, спутниковое ТВ передает EPG для вещаемых каналов в MPEG-TS потоках, и эта услуга бесплатна. Но можно монетизировать ее.

Flussonic может извлекать расписание передач из метаданных MPEG-TS потоков, полученных со спутникового приемника мультикастом по UDP. Он экспортирует данные о EPG в файлы, которые вы можете получать через HTTP API. По мере поступления новых данных расписание на Flussonic обновляется, и эти обновления можно отслеживать и получать.

Расписание можно использовать на стороннем middleware для приставок, чтобы отдавать его абонентам. Другое преимущество в удобстве - EPG в JSON формате можно использовать для интеграции с приложениями в ваших сервисах. Все это значит, что абоненты могут получать расписание через Интернет как часть платных услуг.

Flussonic экспортирует телепрограмму в два формата. Они удобны каждый для своих целей:

XMLTV. Это стандартный формат для описания телевизионных передач, который вы можете загружать в IPTV middleware. Позволяет просматривать расписание и формировать ссылки на передачи из архива.

JSON. Файл, имеющий собственную структуру Flussonic. JSON файлы удобно использовать на веб-страницах.

Flussonic формирует расписание для отдельных каналов, для всех каналов и для группы каналов, например, Спорт.

Как получить EPG

Flussonic начнет формировать расписание, если вы включите IPTV плагин (небольшое встроенное во Flussonic middleware, которое помогает в решении некоторых задач).

Чтобы включить IPTV плагин через веб-интерфейс, перейдите в IPTV и щелкните Enable IPTV.

Теперь можно:

Получать EPG в виде XMLTV или JSON файлов через API, чтобы затем использовать их в ваших сервисах (см. далее на этой странице).

Подписаться на событие `mpegts_epg_changed`, чтобы получать обновленное расписание по мере его обновления.

Обновить программу передач это значит получить новый файл с ней. О том, как подписаться на события, см. Events API.

IPTV API для получения EPG

После того, как вы включили IPTV плагин, вы можете:

Импортировать EPG в файл

Получать обновленный файл по событию Flussonic `mpegts_epg_changed`

EPG получают при помощи IPTV API, который представляет собой часть HTTP API, которая относится к IPTV плагину.

Чтобы получить EPG в формате XMLTV, используйте URL:

`/tv/all/epg.xml`

EPG для всех каналов, доступных абоненту.

`/tv/channel/канал/epg.xml`

EPG для одного канала с указанным названием.

`/tv/group/группа каналов/epg.xml`

EPG для указанной группы каналов.

Для получения EPG в формате JSON, используйте URL:

`/tv/all/epg.json`

EPG для всех каналов, доступных абоненту.

`/tv/channel/канал/epg.json`

EPG для одного канала с указанным названием.

`/tv/group/группа каналов/epg.json`

EPG для указанной группы каналов.

Пример ссылки для загрузки программы передач в XMLTV:

`http://<flussonic-ip>/tv/channel/ORT/epg.xml`

Группы потоков

Конфигурацию Flussonic с группами потоков удобно использовать в IPTV сервисе, где под группами потоков мы понимаем группы ТВ-каналов. Поделив каналы на группы, вы можете предлагать абонентам пакеты услуг, в которые включены разные наборы каналов. Flussonic умеет готовить расписание передач (EPG) для отдельных групп каналов.

Группы потоков задаются в кластерной установке Flussonic Media Server. Группы определяете вы сами в настройках потока на сервере-источнике.

Примеры сценариев использования:

Есть кластер Flussonic серверов, и мы хотим на сервере-рестримере сохранять в архив только избранные потоки. По умолчанию, захватываются все потоки и пишутся в единое место (если опция dvr указана в настройках). Благодаря группам, потоки с сервера-источника можно записывать в разные директории – для каждой группы можно настроить свои опции DVR.

Вы настроили Flussonic забирать EPG из MPEG-TS потоков со спутника и хотите формировать расписание для отдельных групп каналов, чтобы предлагать его абонентам.

Настройка групп потоков

На источнике мы относим потоки к разным группам, а на приемнике (рестримере) задаем для каждой группы индивидуальные настройки, которые Flussonic применит только к потокам этой группы.

Порядок настройки:

На сервере-источнике настройте группы потоков. Для этого пропишите группу в настройках потока. Названия групп произвольные. Можно отнести канал к одной или нескольким группам в зависимости от тематики канала или по любым другим критериям.

Если группу не указать, то на сервере-приемнике (рестримере) к потоку применятся только общие настройки, заданные внутри source, но не настройки группы.

На сервере-приемнике настройте захват потоков с источника при помощи директивы source. Внутри директивы source задайте настройки потоков для разных групп. Flussonic позволяет задать здесь любые настройки потока.

Теперь выполним эти шаги по порядку:

На сервере-источнике в конфигурационном файле Flussonic добавьте директиву group в настройки каждого потока, который хотите включить в группу.

```
stream stream1 {  
  group sport;  
}  
stream stream2 {  
  group cartoons;  
}  
stream stream3 {  
}  
stream stream4 {
```

```
group sport;  
group cartoons;  
}
```

На сервере-приемнике захватывайте потоки с источника при помощи директивы source.
Например:

```
source src1 {  
}
```

На сервере-приемнике добавьте директивы group_config в настройки источника – это позволит задать уникальные настройки для потоков разных групп:

```
source src1 {  
  group_config sport {  
    dvr /dvr_sport;  
  }  
  group_config cartoons {  
    dvr /dvr_cartoons;  
  }  
}
```

Такая настройка говорит Flussonic записывать архивы групп в разные директории.

Замечание. Если поток принадлежит более чем к одной группе, опции, заданные в group_config, объединяются. Если одна опция задана более чем в одной группе, будет использоваться последнее значение.

Врезка рекламы

Flussonic Media Server позволяет работать с рекламой в потоковом видео несколькими способами. Можно организовать вставку рекламы:

Задав расписание показа рекламных роликов через авторизационный бэкенд. [Узнать больше](#)

Включив наш плагин Ad Injector и задав через веб-интерфейс расписание показа рекламных роликов. [Узнать больше](#)

Используя для врезки рекламы специальные метки, если таковые содержатся во входном потоке видеоданных. Flussonic прочитает метки и добавит их в выходной поток. Если вам требуются метки в определенном формате, Flussonic преобразует их в другой формат для выходного потока. [Узнать больше](#)

How-to руководства

Задачи

В этом разделе приведены ссылки на статьи о том, как решить конкретные задачи с помощью Flussonic.

- Как добавить видео с IP-камеры на сайт?
- Как сделать, чтобы звук с камеры был в MP4 (AAC)?
- Как опубликовать видео на Flussonic Media Server?
- Как отправить мультикаст в сеть?
- Как захватить мультикаст поток?
- Как вещать видео в социальные сети (Facebook, YouTube, OK)?
- Как совместить Middleware и Flussonic Media Server?

- Как сделать авторизацию на Flussonic Media Server через Middleware?
- Как организовать запись телепередач?
- Как организовать смещенный по часовому поясу канал?
- Как сделать авторизацию на Flussonic Media Server через Middleware?
- Как организовать запись телепередач?
- Как организовать смещенный по часовому поясу канал?
- Как передать UDP мультикаст через Интернет с помощью Flussonic?
- Как посмотреть файл?
- Каким должен быть источник?
- Как настроить транскодер?
- Как добавить два авторизационных бэкенда?
- Как ограничить доступ по IP адресу?

Flussonic сервер, аппаратная и программная часть

- Как выбрать аппаратную часть?
- Как узнать разрядность операционной системы
- Как уменьшить количество потребляемой памяти?
- Решение проблем с веб-интерфейсом
- Предоставление доступа по SSH для оказания технической поддержки