

Нагрузочное тестирование с использованием захвата потоков по WebRTC/RTMP

Описание

Возможность [захвата потока с другого WCS-сервера по WebRTC](#) может быть полезна для нагрузочного тестирования. Тест проводится по следующему сценарию:

1. На сервере 1 [публикуется](#) поток
2. Сервер 2 создает заданное количество соединений (например, 100) с сервером 1 по WebSocket, как браузерный клиент
3. Сервер 2 [захватывает](#) поток в заданном количестве экземпляров, как зритель, создавая нагрузку на сервер 1.

Тестирование

1. Для теста используем:

- два WCS-сервера: `demo.flashphoner.com` и `wcs5-us.flashphoner.com`;
- веб-приложение [Two Way Streaming](#) для публикации потока;
- веб-приложение [Console](#) для проведения теста;
- браузер Chrome и расширение [Allow-Control-Allow-Origin](#) для работы веб-приложения Console.

2. Откройте приложение Console через HTTP (не через

HTTPS!) <http://demo.flashphoner.com:9091/client2/examples/demo/streaming/console/console.html>

Node ip/domain name	Add node	#	CPU	MEM	TH	CONN	IN	OUT

3. Установите расширение ACAO, разрешите Cross-Origin-Resource-Sharing

Settings

Enable cross-origin resource sharing

Access-Control-Expose-Headers

comma-separated list of headers ...

Intercepted URLs or URL patterns

URL or URL pattern

+

*-/**/*

4. Введите имя сервера `wcs5-us.flashphoner.com` и нажмите `Add node`. Данный сервер будет выполнять роль подписчика, захватывая потоки. Затем аналогично добавьте сервер `demo.flashphoner.com`, который будет выполнять роль тестируемого источника потоков

demo.flashphoner.com

Add node

wcs5-us.flashphoner.com

demo.flashphoner.com

#	CPU	MEM	TH	CONN	IN	OUT
wcs5-us.flashphoner.com	14.47	3717416	66	0	0	0
demo.flashphoner.com	9.32	1870944	85	4	2	1

Pull stream

Pull RTSP stream

Pull streams

Register

Unregister

Call

Hangup

Stress Register

Stress Call

Stress Play Stream

Show

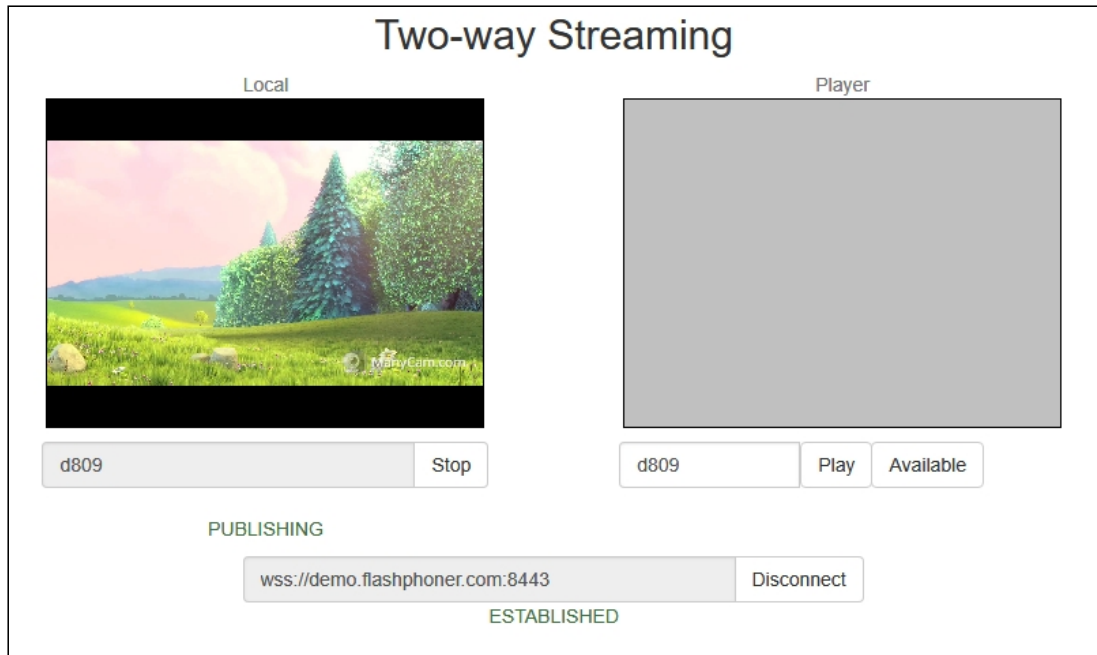
All

Apply

#	NAME	TECH	STATUS	TYPE	AUDIO	VIDEO	ACT
---	------	------	--------	------	-------	-------	-----

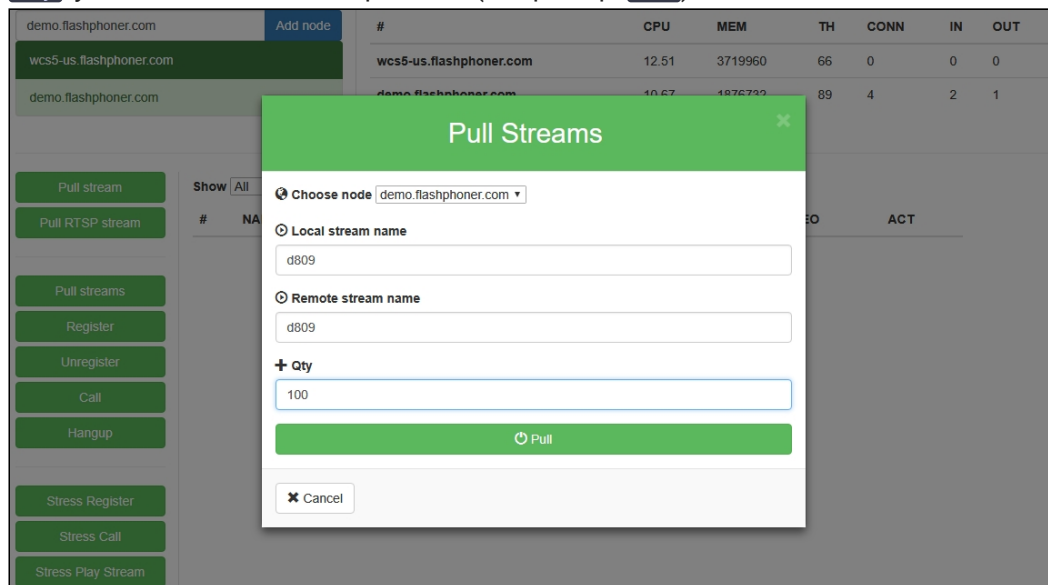
5. Откройте приложение `Two Way Streaming` https://demo.flashphoner.com/client2/examples/demo/streaming/two_way

[_streaming/two_way_streaming.html](#), опубликуйте поток с веб-камеры



6. В приложении **Console** выберите сервер **wcs5-us.flashphoner.com**, нажмите кнопку **Pull streams**, задайте параметры теста:

- **Choose node**: выберите тестируемый сервер **demo.flashphoner.com**
- **Local stream name**, **Remote stream name**: укажите имя опубликованного потока
- **Qty**: укажите количество зрителей (например, **100**)



7. Нажмите **Pull**. Начнется тест

demo.flashphoner.com

+

Add node

wcs5-us.flashphoner.com

+

demo.flashphoner.com

+

#	CPU	MEM	TH	CONN	IN	OUT
wcs5-us.flashphoner.com	54.68	3721380	336	0	21	0
demo.flashphoner.com	10.67	1876732	89	4	2	1

Pull stream

Pull RTSP stream

Pull streams

Register

Unregister

Call

Hangup

Stress Register

Stress Call

Show All

Apply

#	NAME	TECH	STATUS	TYPE	AUDIO	VIDEO	ACT
14f5f3a3-3ff2-4050-90c3-ddc51074c066	d80962	WebRTC	NEW	IN	opus		<div>TERMINATE</div> <div>PUSH</div>
27d95a09-a112-4513-baf0-38eda7df3767	d80969	WebRTC	NEW	IN	opus		<div>TERMINATE</div> <div>PUSH</div>
42a6fe57-9bfb-4fae-9669-39c4f914d615	d80961	WebRTC	NEW	IN	opus		<div>TERMINATE</div> <div>PUSH</div>
f087c3e6-3f0e-4397-9c71-31be6d9a5614	d80965	WebRTC	NEW	IN	opus		<div>TERMINATE</div> <div>PUSH</div>

8. Выберите сервер **demo.flashphoner.com**. На странице отображается список медиасессий, в которых воспроизводится опубликованный поток. В правом верхнем углу отображается информация о текущей нагрузке тестируемого сервера

demo.flashphoner.com

+

Add node

wcs5-us.flashphoner.com

+

demo.flashphoner.com

+

#	CPU	MEM	TH	CONN	IN	OUT
wcs5-us.flashphoner.com	75.18	3721736	1078	0	100	0
demo.flashphoner.com	20.79	1962612	795	104	2	101

Pull stream

Pull RTSP stream

Pull streams

Register

Unregister

Call

Hangup

Stress Register

Stress Call

Show All

Apply

#	NAME	TECH	STATUS	TYPE	AUDIO	VIDEO	ACT
54f48484-f5a4-42da-8069-a2aadfded2d6	d809	WebRTC	PLAYING	OUT	opus	H264	<div>TERMINATE</div>
dd346b9a-dc52-492d-a434-4e913bd2e0d6	d809	WebRTC	PLAYING	OUT	opus	H264	<div>TERMINATE</div>
917109ad-eb1b-4457-919d-92a0bc08c4bf	d809	WebRTC	PLAYING	OUT	opus	H264	<div>TERMINATE</div>
6c6e0068-cc9f-40df-87b1-6776a3401aef	d809	WebRTC	PLAYING	OUT	opus	H264	<div>TERMINATE</div>

Тестирование Edge сервера в CDN

Тестирование Edge сервера в **CDN** проводится по следующему сценарию:

1. На Origin сервере **публикуются** потоки
2. Тестирующий сервер создает заданное количество соединений (например, 100) с сервером Edge по Websocket, как браузерный клиент
3. Тестирующий сервер **захватывает** потоки, доступные Edge серверу, в заданном количестве экземпляров, как зритель, создавая нагрузку на сервер Edge и на CDN

в целом, в части передачи потока с Origin на Edge.

Краткое руководство по тестированию Edge сервера

1. Для теста используем:

- два WCS-сервера для развертывания CDN: `test1.flashphoner.com` и `test2.flashphoner.com`;
- WCS-сервер для проведения теста `demo.flashphoner.com`;
- веб-приложение [Two Way Streaming](#) для публикации потока;
- веб-приложение [Console](#) для проведения теста;
- браузер Chrome и расширение [Allow-Control-Allow-Origin](#) для работы веб-приложения `Console`.

2. Разверните CDN, назначив серверам роли:

- test1 - Origin
 - test2 - Edge
- На Edge сервере добавьте настройку

```
wcs_activity_timer_timeout=86400000
```

3. Откройте приложение `Console` через HTTP (не через HTTPS!) <http://demo.flashphoner.com:9091/client2/examples/demo/streaming/console/console.html>

Node ip/domain name	Add node	#	CPU	MEM	TH	CONN	IN	OUT

4. Разрешите Cross-Origin-Resource-Sharing

Settings

Enable cross-origin resource sharing

Access-Control-Expose-Headers

comma-separated list of headers ...

Intercepted URLs or URL patterns

URL or URL pattern

.//*

5. Введите имя Edge сервера `test2.flashphoner.com` и нажмите `Add node`. Данный сервер будет тестироваться. Затем аналогично добавьте сервер `demo.flashphoner.com`, который будет выполнять роль подписчика, захватывая ПОТОКИ

demo.flashphoner.com

Add node

test2.flashphoner.com

demo.flashphoner.com

#	CPU	MEM	TH	CONN	IN	OUT
test2.flashphoner.com	66.67	1132284	62	0	0	0
demo.flashphoner.com	0.00	NaN	undefined	42	14	14

Pull stream

Pull RTSP stream

Pull streams

Register

Unregister

Call

Hangup

Stress Register

Stress Call

Stress Play Stream

Stress Publish Stream

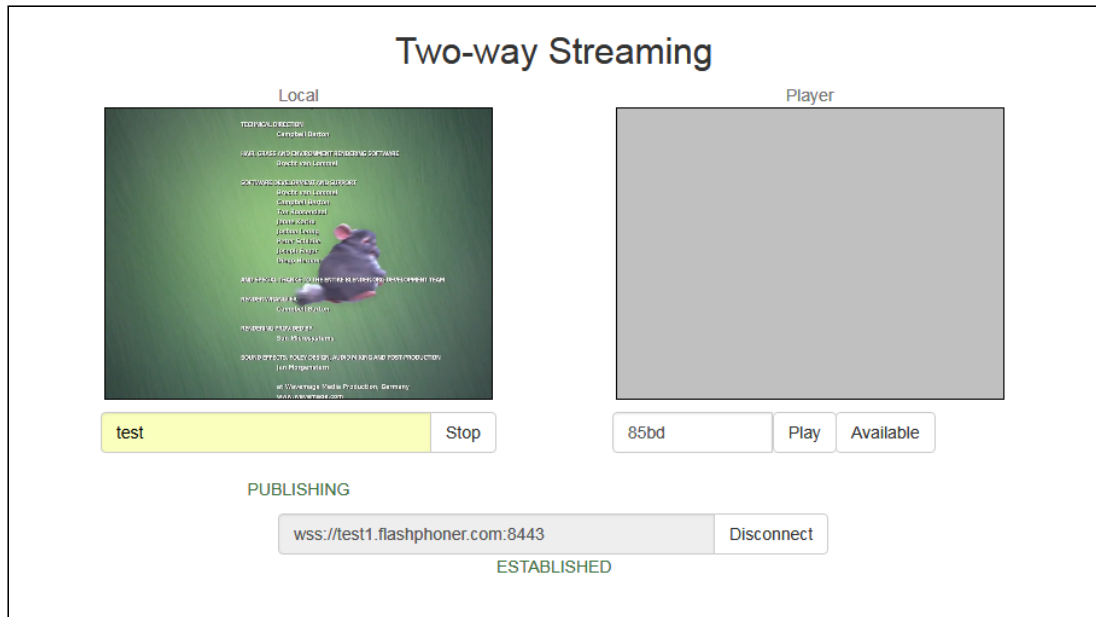
Show

All

Apply

#	NAME	TECH	STATUS	TYPE	AUDIO	VIDEO	ACT
---	------	------	--------	------	-------	-------	-----

6. Откройте приложение **Two Way Streaming**, опубликуйте поток с веб-камеры



7. В приложении **Console** выберите сервер **demo.flashphoner.com**, нажмите кнопку **Stress play stream**, задайте параметры теста:

- **Choose node**: выберите тестируемый сервер **test2.flashphoner.com**
- **Choose test mode**: **Random**
- **CDN**: установите переключатель

- `Max streams`: укажите количество зрителей (например, `100`)

Stress Play Stream

Choose node

test2.flashphoner.com

Choose test mode

Random

In this mode stream name will be fetched randomly from target node

☒

CDN

Stream life time

1 min

Fake stream requests

0 %

Max streams

100

Rate

1

Start

Cancel

8. Нажмите `Start`. Начнется тест

Захват потоков по RTMP

В сборке [5.2.767](#) добавлена возможность захвата потоков при нагрузочном тестировании по RTMP. Чтобы такой тест работал, необходимо в настройках тестирующего сервера указать параметр

```
rtmp_pull_allow_to_reuse_uri=true
```

В конфигурации теста необходимо выбрать `Proto pull: RTMP`

Pull Streams

Choose node

demo.flashphoner.com

Proto pull

RTMP

Local stream name

test

Remote stream name

test

+ Qty

100

⏻ Pull

✕ Cancel

Сам тест работает так же, как [WebRTC-тест](#)

Рекомендации по настройке

Если тест не был пройден успешно, рекомендуется на тестируемом сервере изменить следующие настройки:

1. В файле [flashphoner.properties](#) расширить диапазон UDP-портов и запретить быстрый старт декодера

```
media_port_from = 20000
media_port_to = 39999
streaming_video_decoder_fast_start=false
```

2. В файле [wcs-core.properties](#) увеличить предельный размер памяти для кучи. Рекомендуется задавать этот размер в половину физической памяти, например, 16 Гб при объеме физической памяти в 32 Гб. Убедитесь, что памяти на сервер достаточно для этого:

```
-Xmx16g -Xms16g
```

Известные проблемы

1. По умолчанию, захватывается не более 1000 потоков

Bug

При указании количества зрителей более 1000, захватывается 998 потоков

Success

По умолчанию, максимальное количество портов агентов ограничено 999:

```
wcs_agent_port_from=34001  
wcs_agent_port_to=35000
```

Для расширения данного ограничения, необходимо увеличить значение

```
wcs_agent_port_to=35000
```