Скрипты для тестирования максимального количества WebRTC публикаций/подписчиков

- Тестирование максимального количества WebRTC публикаций
- Тестирование максимального количества WebRTC подписчиков

В сборке 5.2.1766 добавлены скрипты для нагрузочного тестирования WCS на максимальное число публикаций или подписчиков. Скрипты написаны на bash и используют REST API для публикации и захвата потоков с тестируемого WCS сервера. Скрипты располагаются в каталоге `/usr/local/FlashphonerWebCallServer/tools`

Для нагрузочного тестирования необходимы:

- тестируемый WCS сервер
- тестирующий WCS сервер
- средство публикации исходного потока (например, OBS для публикации RTMP, или пример Media Devices для публикации WebRTC)

Обратите внимание, что производительность тестирующего сервера должна быть минимум вдвое больше, чем тестируемого, в противном случает сам тестирующий сервер станет узким местом в нагрузочном тесте.

Тестирование максимального количества WebRTC публикаций

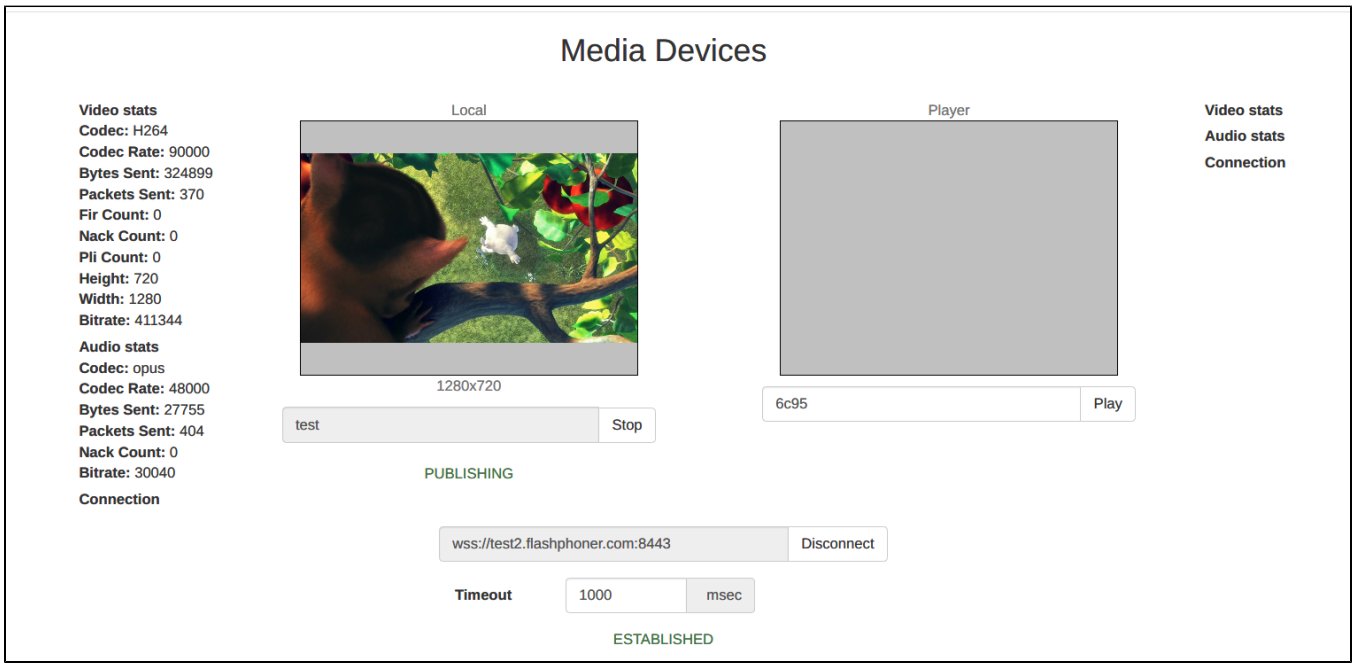
Тестирование максимального количества публикаций проводится при помощи скрипта

```
/usr/local/FlashphonerWebCallServer/tools/pushStreams.sh
```

1. Для теста используются:

- `test1.flashphoner.com` - тестируемый WCS сервер
- `test2.flashphoner.com` - тестирующий WCS сервер
- пример Media Devices для публикации исходного потока

2. Публикация исходного потока на сервер `test2.flashphoner.com`



3. Запуск скрипта `pushStreams` на сервере `test2.flashphoner.com`

```
/usr/local/FlashphonerWebCallServer/tools/pushStreams.sh ws://test1.flashphoner.com:8080 http://test1.
flashphoner.com:8081 test loadtest 10
```

Здесь:

- `ws://test1.flashphoner.com:8080` - websocket URL тестируемого сервера

- `http://test1.flashphoner.com:8081` - HTTP URL тестируемого сервера для использования REST API
- `test` - имя исходного потока, который будет публиковаться на тестируемый сервер
- `loadtest` - префикс имени, с которым будут публиковаться потоки на тестируемый сервер
- `10` - количество публикаций

При успешном запуске, скрипт выводит на консоль список публикаций

```
Publishing stream test to ws://test1.flashphoner.com:8080 as loadtest1
Publishing stream test to ws://test1.flashphoner.com:8080 as loadtest2
Publishing stream test to ws://test1.flashphoner.com:8080 as loadtest3
Publishing stream test to ws://test1.flashphoner.com:8080 as loadtest4
Publishing stream test to ws://test1.flashphoner.com:8080 as loadtest5
Publishing stream test to ws://test1.flashphoner.com:8080 as loadtest6
Publishing stream test to ws://test1.flashphoner.com:8080 as loadtest7
Publishing stream test to ws://test1.flashphoner.com:8080 as loadtest8
Publishing stream test to ws://test1.flashphoner.com:8080 as loadtest9
Publishing stream test to ws://test1.flashphoner.com:8080 as loadtest10
Stream test is publishing successfully as loadtest1 to ws://test1.flashphoner.com:8080
Stream test is publishing successfully as loadtest2 to ws://test1.flashphoner.com:8080
Stream test is publishing successfully as loadtest3 to ws://test1.flashphoner.com:8080
Stream test is publishing successfully as loadtest4 to ws://test1.flashphoner.com:8080
Stream test is publishing successfully as loadtest5 to ws://test1.flashphoner.com:8080
Stream test is publishing successfully as loadtest6 to ws://test1.flashphoner.com:8080
Stream test is publishing successfully as loadtest7 to ws://test1.flashphoner.com:8080
Stream test is publishing successfully as loadtest8 to ws://test1.flashphoner.com:8080
Stream test is publishing successfully as loadtest9 to ws://test1.flashphoner.com:8080
Stream test is publishing successfully as loadtest10 to ws://test1.flashphoner.com:8080
```

4. Контроль публикаций на странице статистики тестируемого сервера `http://test1.flashphoner.com:8081/?action=stat`

5.Запуск скрипта `pushStreams` на сервере `test2.flashphoner.com` для остановки теста

```
/usr/local/FlashphonerWebCallServer/tools/pushStreams.sh ws://test1.flashphoner.com:8080 http://test1.
flashphoner.com:8081 test loadtest stop
```

Здесь:

- `ws://test1.flashphoner.com:8080` - websocket URL тестируемого сервера
- `http://test1.flashphoner.com:8081` - HTTP URL тестируемого сервера для использования REST API
- `test` - имя исходного потока, который публикуется на тестируемый сервер
- `loadtest` - префикс имени, с которым публикуются потоки на тестируемый сервер
- `stop` - ключевое слово для остановки публикаций с указанными параметрами

При успешной остановке публикаций, скрипт выводит на консоль сведения об остановке

```
Stream test is publishing as loadtest1 to ws://test1.flashphoner.com:8080, stopping
Stream test is publishing as loadtest2 to ws://test1.flashphoner.com:8080, stopping
Stream test is publishing as loadtest3 to ws://test1.flashphoner.com:8080, stopping
Stream test is publishing as loadtest4 to ws://test1.flashphoner.com:8080, stopping
Stream test is publishing as loadtest5 to ws://test1.flashphoner.com:8080, stopping
Stream test is publishing as loadtest6 to ws://test1.flashphoner.com:8080, stopping
Stream test is publishing as loadtest7 to ws://test1.flashphoner.com:8080, stopping
Stream test is publishing as loadtest8 to ws://test1.flashphoner.com:8080, stopping
Stream test is publishing as loadtest9 to ws://test1.flashphoner.com:8080, stopping
Stream test is publishing as loadtest10 to ws://test1.flashphoner.com:8080, stopping
```

Тестирование максимального количества WebRTC подписчиков

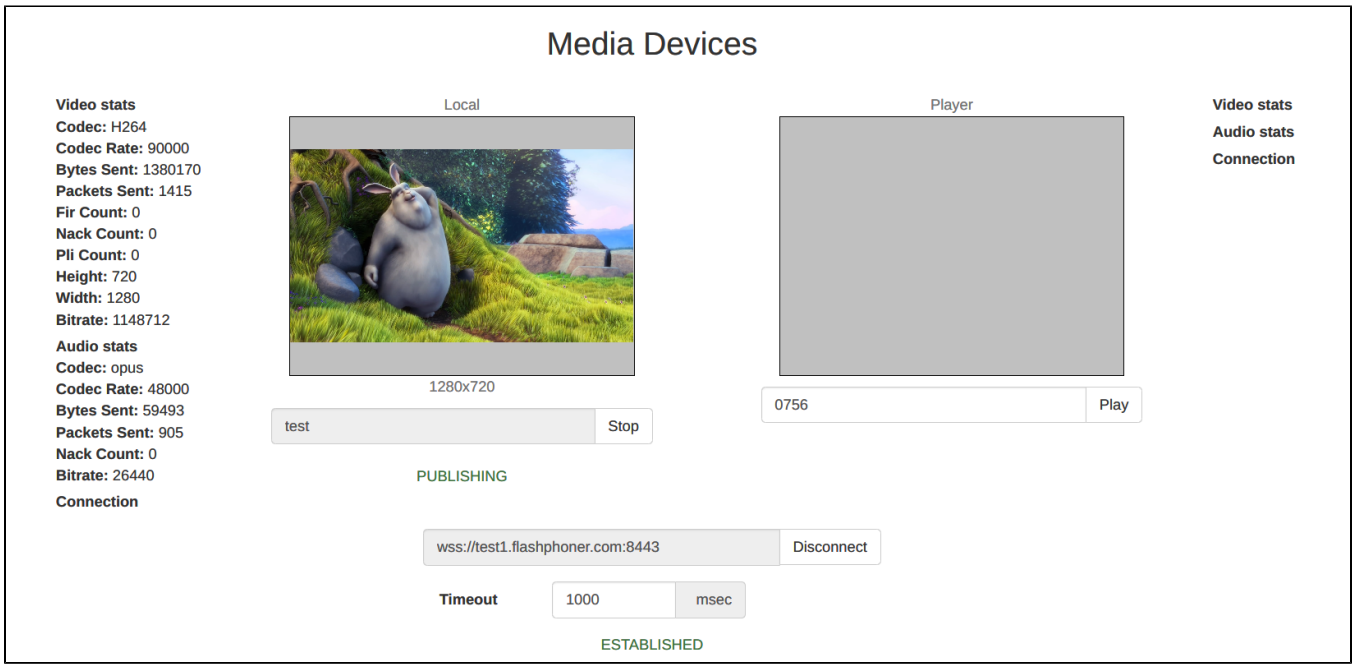
Тестирование максимального количества подписчиков проводится при помощи скрипта

```
/usr/local/FlashphonerWebCallServer/tools/pullStreams.sh
```

1. Для теста используются:

- `test1.flashphoner.com` - тестируемый WCS сервер
- `test2.flashphoner.com` - тестирующий WCS сервер
- пример Media Devices для публикации исходного потока

2. Публикация исходного потока на сервер `test1.flashphoner.com`



3. Запуск скрипта `pullStreams` на сервере `test2.flashphoner.com`

```
/usr/local/FlashphonerWebCallServer/tools/pullStreams.sh ws://test1.flashphoner.com:8080 http://test1.
flashphoner.com:8081 loadtest 10
```

Здесь:

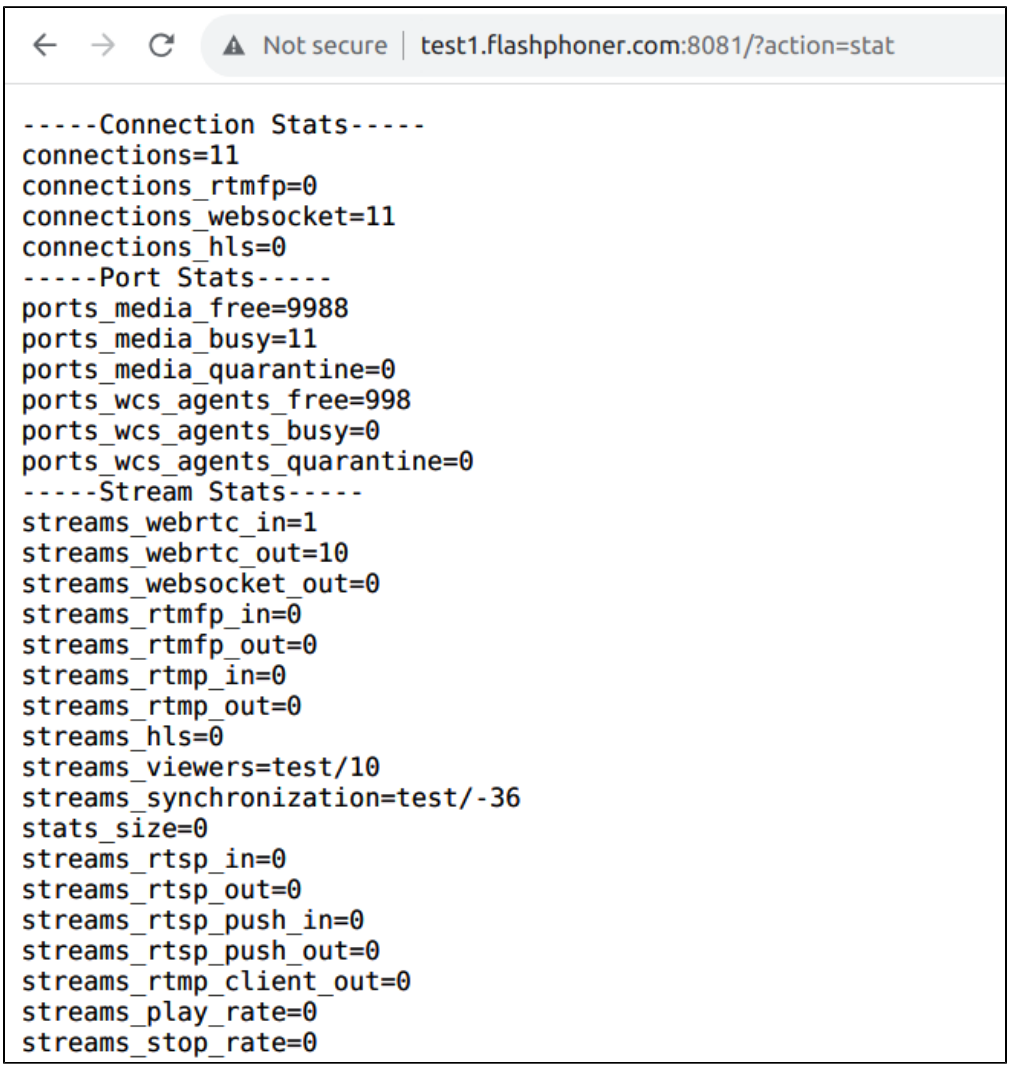
- `ws://test1.flashphoner.com:8080` - websocket URL тестируемого сервера
- `http://test1.flashphoner.com:8081` - HTTP URL тестируемого сервера для использования REST API
- `loadtest` - префикс имени, с которым будут захватываться потоки с тестируемого сервера
- `10` - количество подписчиков

Обратите внимание: скрипт захватит все опубликованные на тестируемом сервере потоки, каждый заданным количеством подписчиков.

При успешном запуске, скрипт выводит на консоль список подписчиков

```
Playing stream test from ws://test1.flashphoner.com:8080 as loadtest1-test
Playing stream test from ws://test1.flashphoner.com:8080 as loadtest2-test
Playing stream test from ws://test1.flashphoner.com:8080 as loadtest3-test
Playing stream test from ws://test1.flashphoner.com:8080 as loadtest4-test
Playing stream test from ws://test1.flashphoner.com:8080 as loadtest5-test
Playing stream test from ws://test1.flashphoner.com:8080 as loadtest6-test
Playing stream test from ws://test1.flashphoner.com:8080 as loadtest7-test
Playing stream test from ws://test1.flashphoner.com:8080 as loadtest8-test
Playing stream test from ws://test1.flashphoner.com:8080 as loadtest9-test
Playing stream test from ws://test1.flashphoner.com:8080 as loadtest10-test
Stream test is playing successfully as loadtest1-test from ws://test1.flashphoner.com:8080
Stream test is playing successfully as loadtest2-test from ws://test1.flashphoner.com:8080
Stream test is playing successfully as loadtest3-test from ws://test1.flashphoner.com:8080
Stream test is playing successfully as loadtest4-test from ws://test1.flashphoner.com:8080
Stream test is playing successfully as loadtest5-test from ws://test1.flashphoner.com:8080
Stream test is playing successfully as loadtest6-test from ws://test1.flashphoner.com:8080
Stream test is playing successfully as loadtest7-test from ws://test1.flashphoner.com:8080
Stream test is playing successfully as loadtest8-test from ws://test1.flashphoner.com:8080
Stream test is playing successfully as loadtest9-test from ws://test1.flashphoner.com:8080
Stream test is playing successfully as loadtest10-test from ws://test1.flashphoner.com:8080
```

4. Контроль подписчиков на странице статистики тестируемого сервера `http://test1.flashphoner.com:8081/?action=stat`



5.Запуск скрипта `pullStreams` на сервере `test2.flashphoner.com` для остановки теста

```
/usr/local/FlashphonerWebCallServer/tools/pullStreams.sh ws://test1.flashphoner.com:8080 http://test1.
flashphoner.com:8081 loadtest stop
```

Здесь:

- `ws://test1.flashphoner.com:8080` - websocket URL тестируемого сервера
- `http://test1.flashphoner.com:8081` - HTTP URL тестируемого сервера для использования REST API
- `loadtest` - префикс имени, с которым захватываются потоки с тестируемого сервера
- `stop` - ключевое слово для остановки подписчиков с указанными параметрами

При успешной остановке подписок, скрипт выводит на консоль сведения об остановке

```
Stream test is playing from ws://test1.flashphoner.com:8080 as loadtest1-test, stopping
Stream test is playing from ws://test1.flashphoner.com:8080 as loadtest2-test, stopping
Stream test is playing from ws://test1.flashphoner.com:8080 as loadtest3-test, stopping
Stream test is playing from ws://test1.flashphoner.com:8080 as loadtest4-test, stopping
Stream test is playing from ws://test1.flashphoner.com:8080 as loadtest5-test, stopping
Stream test is playing from ws://test1.flashphoner.com:8080 as loadtest6-test, stopping
Stream test is playing from ws://test1.flashphoner.com:8080 as loadtest7-test, stopping
Stream test is playing from ws://test1.flashphoner.com:8080 as loadtest8-test, stopping
Stream test is playing from ws://test1.flashphoner.com:8080 as loadtest9-test, stopping
Stream test is playing from ws://test1.flashphoner.com:8080 as loadtest10-test, stopping
```