

Тестирование производительности сервера

- [Подготовка сервера](#)
- [Проведение теста](#)
 - [Тест с транскодированием потока](#)
 - [Тест без транскодирования потока](#)
- [Результаты тестирования](#)
 - [Физический сервер](#)
 - [Виртуальный сервер](#)
 - [Сервер в облаке \(на примере AWS\)](#)
- [Рекомендации](#)

Работа с видеоконтентом относится к высоконагруженной задаче, требующей от сервера, на котором установлен WCS, соответствие определенным требованиям.

Минимальные требования к серверу указаны в [документации](#), но определить, достаточно ли производительности оборудования для вашего проекта, можно только выполнив ряд нагрузочных тестов по условиям вашего типового использования WCS.

Для выполнения теста вам потребуется:

- виртуальный (VPS) или физический сервер
- источник видео трансляции (потокковое видео с программ OBS studio или ManyCam, трансляция с вашей web или IP-камеры)
- сервер WCS для эмуляции зрителей вашего контента

Проверим возможности Linux x86_64 сервера с производительностью, соответствующей минимальным требованиям для WCS:

- 2 гигабайта оперативной памяти
- 10 гигабайт места на диске
- 1 ядро процессора

Подготовка сервера

1. Указываем в файле настроек запуска ядра WCS сервера `wcs-core.properties` [размер](#) Java memory heap 1 GB:

```
-Xmx1024M
```

Затем перезапускаем WCS.

2. Все современные серверные процессоры многоядерные. Проведем нагрузочные тесты, используя только одно core CPU, отключив все остальные. Для этого смотрим текущий статус используемых ядер CPU:

```
[root@demo ~]# lscpu | grep list
On-line CPU(s) list: 0-3
```

По выводу видим, что на данном сервере 4 core CPU (0,1,2,3). Отключаем все ядра, кроме четвертого:

```
echo 0 | sudo tee /sys/devices/system/cpu/cpu0/online
echo 0 | sudo tee /sys/devices/system/cpu/cpu1/online
echo 0 | sudo tee /sys/devices/system/cpu/cpu2/online
```

Текущий статус после выполнения команд:

```
[root@demo ~]# lscpu | grep list
On-line CPU(s) list: 3
Off-line CPU(s) list: 0-2
```

Для теста используем 1 CPU Intel Xeon E3-1240 v5@3.50GHz.

Проведение теста

Тест с транскодированием потока

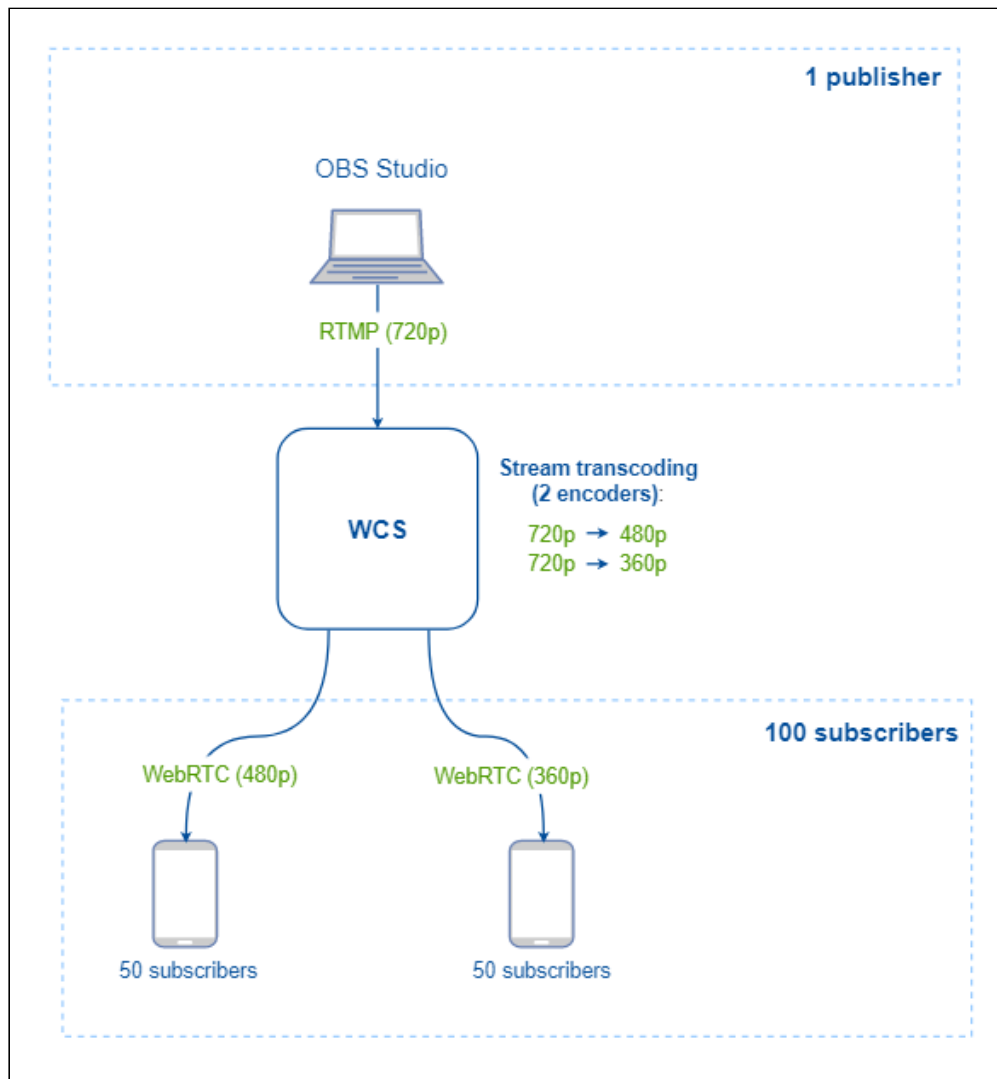
1. Публикуем RTMP поток с определенными параметрами качества из программы OBS Studio на WCS сервер (пример указан в [документации](#)).

Разрешение	Битрейт, кбит/с
1280x720 (720p)	1500

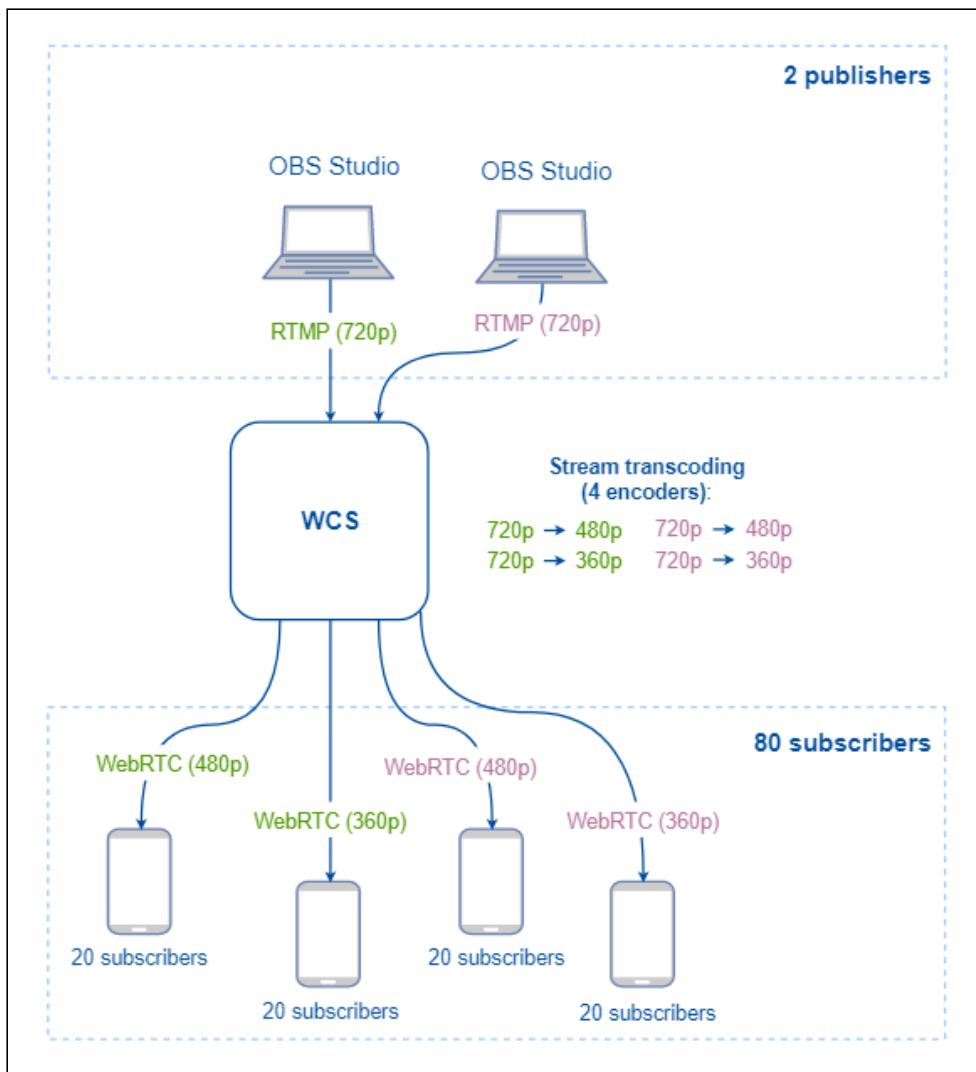
2. Транскодируем этот поток с помощью [REST запросов](#) в несколько популярных разрешений (480p и 360p).

Разрешение	Битрейт, кбит/с
854x480 (480p)	1000
640x360 (360p)	500

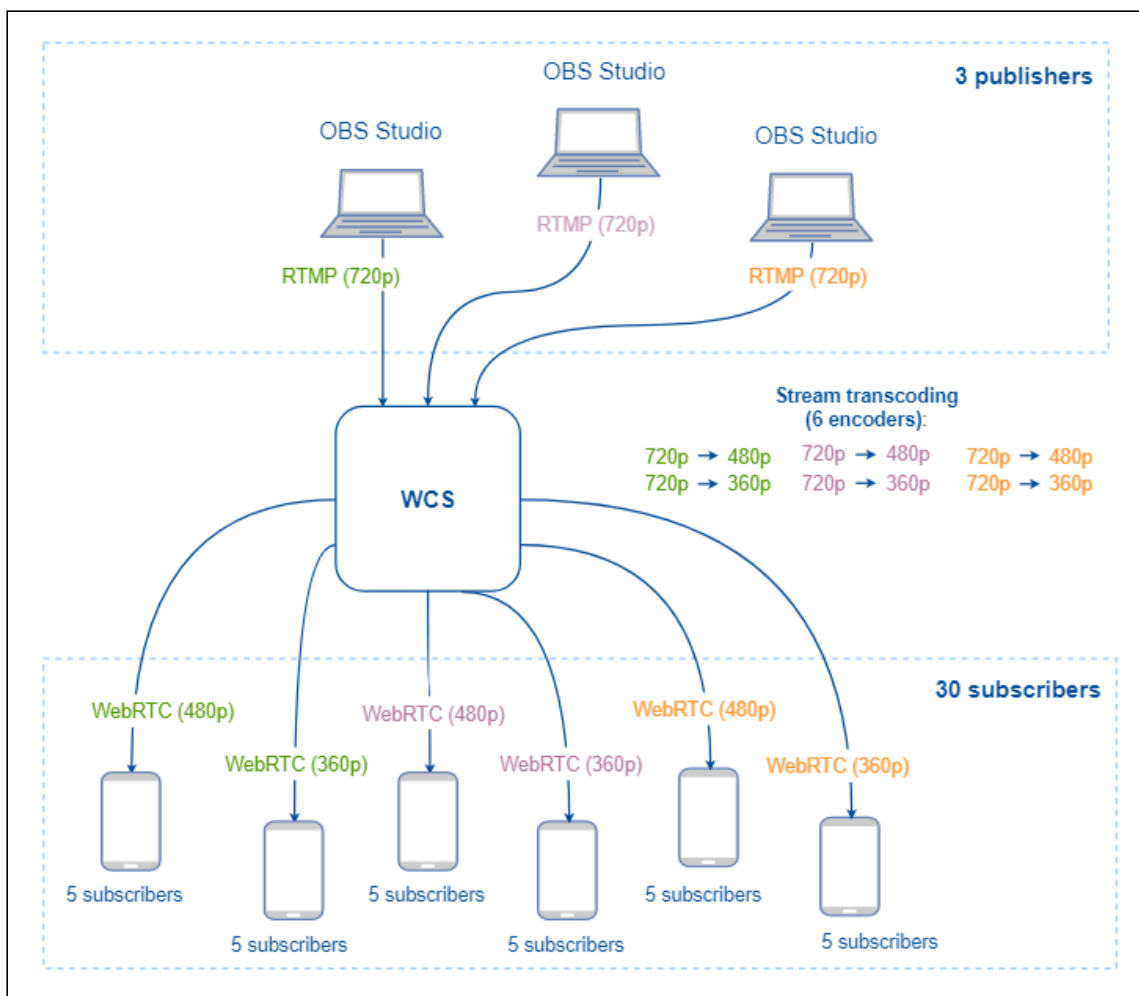
3. Используем [пример нагрузочного тестирования](#) с захватом потоков по WebRTC на другом сервере WCS. Этим примером мы эмулируем зрителей (подписчиков) трансляции, воспроизводящих поток в разных разрешениях (720p, 480p, 360p). При заданном числе зрителей (около 100) нагрузка процессора на WCS сервере приближается к 80%, это рекомендуемая максимальная нагрузка на CPU, при которой сервер выполняет свой функционал корректно.



4. Публикуем два потока 720p, транскодируем их в разрешения 480p и 360p, добавляем в нагрузочное тестирование и по загрузке процессора на сервере с WCS (как описано выше, допустимо до 80%) определяем максимальное количество подписчиков на эти трансляции (около 80).



5. Аналогично выполнив тест с тремя потоками 720p, получаем возможное число зрителей около 30.

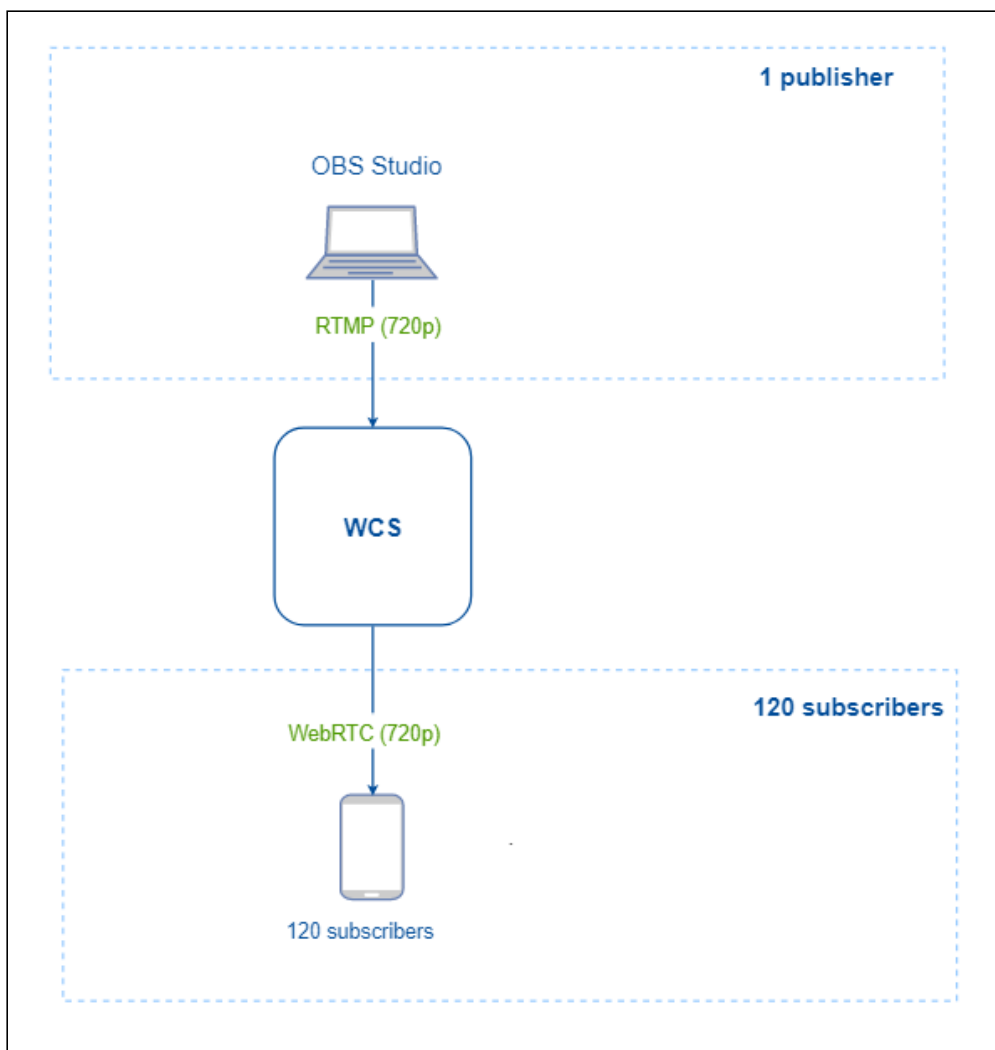


Тест без транскодирования потока

1. В этом тесте проверим, сколько зрителей получит возможность просмотра трансляции без транскодирования потока на сервере, т.е. публикуем RTMP поток с определенными параметрами качества из программы OBS Studio на WCS сервер (пример указан в [документации](#)) и просматриваем его же подписчиками.

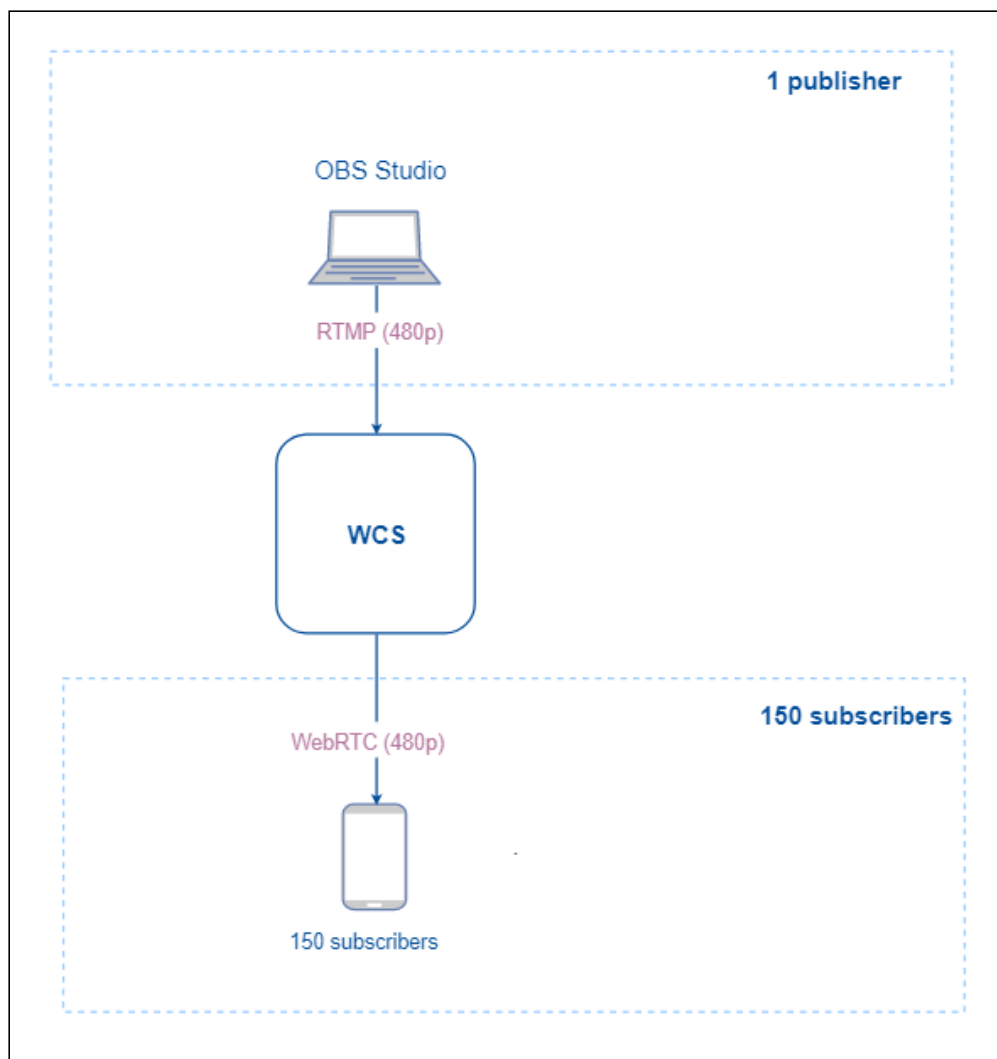
Разрешение	Битрейт, кбит/с
1280x720 (720p)	1500

2. Используем [пример нагрузочного тестирования](#) с захватом потоков по WebRTC на другом сервере WCS. Этим примером мы эмулируем зрителей (подписчиков) трансляции, воспроизводящих поток с WCS сервера. Увеличиваем число зрителей до достижения на WCS сервере параметров загрузки процессора до 80%. При просмотре трансляции 720р мы получили возможное число зрителей потока - 120 подписчиков.



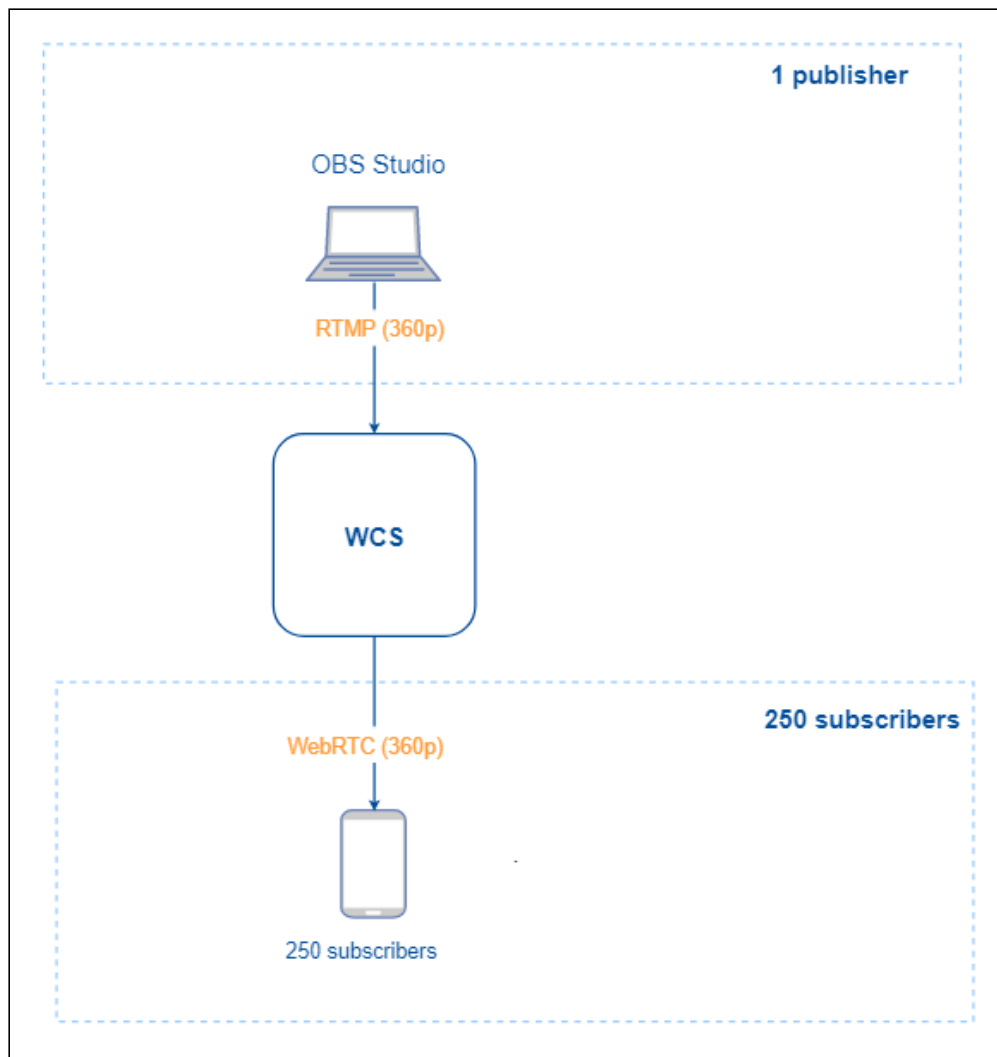
3. Повторяем тест с трансляцией 480р и нагрузочным тестом с захватом потоков на другом сервере WCS.

Разрешение	Битрейт, кбит/с
854x480 (480p)	1000



4. Проверим возможности сервера при трансляции 360p и нагрузочным тестом.

Разрешение	Битрейт, кбит/с
640x360 (360p)	500



Результаты тестирования

Физический сервер

На основании теста с минимально рекомендуемой конфигурацией (1 CPU, 2 GB RAM, 1 GB RAM для Java heap) на выделенном (физическом) сервере, определили примерные возможности WCS по работе с потоковым видео на таком сервере:

Без транскодинга

Тест

Публикации

Зрители

Количество

Разрешение

Битрейт, кбит/с

Количество

1

1

1280x720 (720p)

1500

120

2

1

854x480 (480p)

1000

150

3

1

640x360 (360p)

500

250

С транскодингом

Тест

Публикации

Зрители

Количество

Разрешение

Битрейт, кбит/с

Разрешение

Битрейт, кбит/с

Количество

1

1

1280x720 (720p)

1500

854x480 (480p)

1000

50

640x360 (360p)

500

50

2

1

1280x720 (720p)

1500

854x480 (480p)

1000

20

640x360 (360p)

500

20

1

1280x720 (720p)

1500

854x480 (480p)

1000

20

640x360 (360p)

500

20

3

1

1280x720 (720p)

1500

854x480 (480p)

1000

5

640x360 (360p)

500

5

1

1280x720 (720p)

1500

854x480 (480p)

1000

5

640x360 (360p)

500

5

1

1280x720 (720p)

1500

854x480 (480p)

1000

5

640x360 (360p)

500

5

Виртуальный сервер

Виртуальный сервер при схожих параметрах предлагает меньшую производительность, чем физический сервер из-за ряда причин (например, особенности системы виртуализации), но позволяет масштабировать вычислительные мощности при необходимости в кратчайшие сроки.

Результаты тестов с минимально рекомендуемой конфигурацией (1 CPU, 2 Gb RAM, 1 GB RAM для Java heap) на виртуальном сервере от Digital Ocean (digitalocean.com):

Без транскодинга

Тест

Публикации

Зрители

Количество

Разрешение

Битрейт, кбит/с

Количество

1

1

1280x720 (720p)

3000

50

2

1

854x480 (480p)

1800

70

3

1

640x360 (360p)

1300

70

С транскодингом

Тест

Публикации

Количество

Разрешение

Битрейт, кбит/с

1

2

1280x720 (720p)

3000

2

3

854x480 (480p)

1800

3

5

640x360 (360p)

1300

Сервер в облаке (на примере AWS)

Тесты на серверах в облаке Amazon (виртуальных и физических) проводились с применением следующих настроек

1. Включено [аппаратное ускорение шифрования WebRTC трафика](#)

2. Включена [оптимизация доставки потока подписчикам](#)

На тестируемом сервере был опубликован поток 1080p с битрейтом 2,2 Мбит/с, без значительных скачков битрейта.

Были получены следующие результаты:

Тип инстанса	CPUs	RAM, Gb	Bandwidth, Gbps	Количество подписчиков
c5.4xlarge	16	32	до 10	1500
c5.9xlarge	36	72	10	2000 (предел пропускной способности сети)
c5n.9xlarge	36	96	50	3000

Рекомендации

По проведенным тестам можно сделать вывод, что физический сервер при схожих параметрах оборудования показывает большую производительность по сравнению с виртуальным сервером. Разнообразие устройств для просмотра и работы с потоковым видео (это как и мобильные платформы, так и web-интеграции контента), ограничения по емкости сетевых каналов до зрителей в свою очередь требуют значительных ресурсов для транскодирования потоков на WCS сервере. Примерные требования по производительности сервера для WCS при типовых задачах указаны ниже:

Количество подписчиков	CPUs	RAM, Gb	Трафик, Tb	Пример использования
до 200	4	8	5	Система видеонаблюдения
до 500	8	16	6	Вебинары
до 1000	16	64	9	Видеочат
до 2000	20	96	10	Стриминг HD видео

При большем количестве потоков и зрителей, усложнению бизнес модели проекта, наращивание производительности одного WCS сервера нецелесообразно и ведет к появлению единой точки отказа. Масштабирование, географическое и логическое разделение (с выделением в зависимости от производительности и [роли отдельных](#)

серверов в CDN функций транскодинга и доставки контента) позволяет на основании предложенных и выполненных нами тестов гибко определить необходимый уровень производительности для каждого из WCS серверов.

Attachments:

- perf-1.png (image/png)
- perf-1.png (image/png)
- perf-2.png (image/png)
- perf-3.png (image/png)
- perf-1.png (image/png)
- perf-2.png (image/png)
- perf-3.png (image/png)
- perf-2-1.png (image/png)
- perf-2-2.png (image/png)
- perf-2-1.png (image/png)
- perf-2-2.png (image/png)
- perf-2-3.png (image/png)
- perf-1.png (image/png)
- perf-1.png (image/png)
- perf-2.png (image/png)
- perf-3.png (image/png)