

## Введение

В этом эксперименте созданы максимально правдоподобные условия. Сервер находится на удалённом сервере в Нидерландах

Технические характеристики сервера:

- CPU 1 vCPU
- RAM 2 GB
- Storage 20 GB
- Speed 1200 Mbps

Техническая характеристика моего ноутбука:

- CPU 4 ядра
- RAM 8 GB
- Speed 50 Mbps

В качестве REST сервера используется Node.js express.js, обратный прокси сервер (сжатие http) Nginx

Логика следующая:

- клиент заходит на сайт
- скачивает статические файлы
- когда выполнится js код, выполнится get запрос на получение всех видео /videos/getAll
- сервер сделает запрос в базу данных
- возвращает ответ

Тестирование проводится с помощью k6. Замеряются такие параметры, как:

- CPU usage
- Memory Usage
- Latency
- Availability

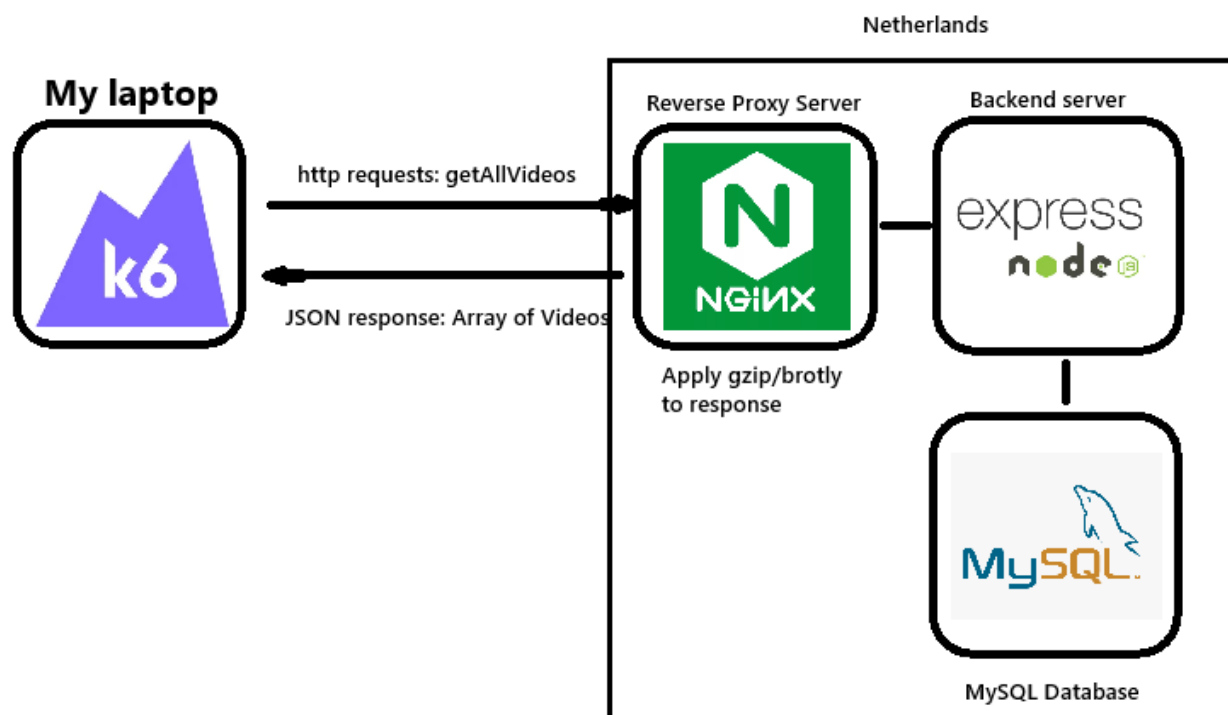
В зависимости от количества запросов в секунду (RPS)

Так выглядит ответ от сервера:

```
[
  {
    title: "Steel Horizon",
    description:
      '"Steel Horizon" is a captivating cinematic journey that explores the boundaries of imagination and reality. With stunning visuals and a compelling narrative, it draws viewers into a richly woven tale full of emotion, suspense, and intrigue. As the characters navigate through complex challenges, deep personal struggles, and unexpected twists, the story unfolds with intensity and grace. Crafted by visionary creators, the film blends elements of classic storytelling with modern cinematic techniques to create an unforgettable experience. Whether you\'re drawn to heartfelt drama, thrilling action, or thought-provoking ideas,
```

```
this film offers a powerful reflection on humanity, resilience, and discovery.',
  number: 19,
  src_url: "https://cdn.example.com/videos/video_19.mp4",
  preview_url: "/previews/bearwolf.mp4",
  image_url: "/images/bearwolf2.jpg",
  studios: ["MegaPix"],
  tags: ["documentary", "action", "comedy"],
},
]
```

## Схема проекта



## Эксперимент первый

Нагрузка:

```
stages: [
  { duration: "5s", target: 10 },
  { duration: "10s", target: 20 },
  { duration: "10s", target: 30 },
  { duration: "10s", target: 40 },
  { duration: "10s", target: 50 },
  { duration: "10s", target: 60 },
  { duration: "10s", target: 70 },
  { duration: "10s", target: 80 },
  { duration: "10s", target: 90 },
  { duration: "10s", target: 100 },
  { duration: "10s", target: 110 },
  { duration: "10s", target: 120 },
```

```
[
  { duration: "10s", target: 130 },
  { duration: "10s", target: 140 },
  { duration: "10s", target: 150 },
  { duration: "10s", target: 170 },
  { duration: "10s", target: 190 },
  { duration: "10s", target: 210 },
  { duration: "5s", target: 5 },
]
```

duration - время каждой стадии, target - кол-во одновременно активных пользователей. Каждый пользователь отправляет запрос на сервер, ждёт ответа, далее ждёт 1 секунду.

Результаты без сжатия:

```

TOTAL RESULTS
checks_total.....: 4662    23.364188/s
checks_succeeded.....: 100.00% 4662 out of 4662
checks_failed.....: 0.00% 0 out of 4662

✓ status is 200

CUSTOM
receiving_time.....: avg=2660.438207 min=187.6492 med=2388.9946 max=45987.4441 p(90)=5330.85794 p(95)=6379.181005
total_duration.....: avg=2767.870548 min=254.0539 med=2492.9023 max=46186.5341 p(90)=5476.16703 p(95)=6528.734735
waiting_time.....: avg=106.595954 min=59.3073 med=85.0042 max=661.0271 p(90)=186.68654 p(95)=218.930525

HTTP
http_req_duration.....: avg=2.76s min=254.05ms med=2.49s max=46.18s p(90)=5.47s p(95)=6.52s
{ expected_response:true }.....: avg=2.76s min=254.05ms med=2.49s max=46.18s p(90)=5.47s p(95)=6.52s
http_req_failed.....: 0.00% 0 out of 4662
http_reqs.....: 4662    23.364188/s

EXECUTION
iteration_duration.....: avg=3.77s min=1.25s med=3.49s max=47.18s p(90)=6.48s p(95)=7.53s
iterations.....: 4662    23.364188/s
vus.....: 1 min=1 max=210
vus_max.....: 210 min=210 max=210

NETWORK
data_received.....: 983 MB 4.9 MB/s
data_sent.....: 574 kB 2.9 kB/s

http_req_failed.....: 0.00% 0 out of 4662
http_reqs.....: 4662    23.364188/s

```

Результаты с сжатием (gzip 9):

```

TOTAL RESULTS
checks_total.....: 10324    57.009845/s
checks_succeeded.....: 100.00% 10324 out of 10324
checks_failed.....: 0.00% 0 out of 10324

✓ status is 200

CUSTOM
receiving_time.....: avg=1.456395 min=0 med=1.0023 max=592.0935 p(90)=2.00117 p(95)=2.69787
total_duration.....: avg=671.715013 min=64.0334 med=609.7224 max=1958.2931 p(90)=1511.34345 p(95)=1721.26277
waiting_time.....: avg=670.232343 min=63.0417 med=607.90245 max=1957.7853 p(90)=1509.78521 p(95)=1719.59562

HTTP
http_req_duration.....: avg=671.71ms min=64.03ms med=609.72ms max=1.95s p(90)=1.51s p(95)=1.72s
{ expected_response:true }.....: avg=671.71ms min=64.03ms med=609.72ms max=1.95s p(90)=1.51s p(95)=1.72s
http_req_failed.....: 0.00% 0 out of 10324
http_reqs.....: 10324    57.009845/s

EXECUTION
iteration_duration.....: avg=1.67s min=1.06s med=1.61s max=2.97s p(90)=2.51s p(95)=2.72s
iterations.....: 10324    57.009845/s
vus.....: 1 min=1 max=209
vus_max.....: 210 min=210 max=210

NETWORK
data_received.....: 61 MB 339 kB/s
data_sent.....: 1.3 MB 7.0 kB/s

```

## Эксперимент второй

Нагрузка:

```
stages: [
  { duration: "10s", target: 10 },
  { duration: "20s", target: 10 },
  { duration: "20s", target: 15 },
  { duration: "20s", target: 20 },
  { duration: "20s", target: 25 },
  { duration: "20s", target: 30 },
  { duration: "10s", target: 5 },
]
```

Результаты без сжатия:

```

TOTAL RESULTS
checks_total.....: 1454    12.003994/s
checks_succeeded.....: 100.00% 1454 out of 1454
checks_failed.....: 0.00% 0 out of 1454

✓ status is 200

CUSTOM
receiving_time.....: avg=309.471141 min=173.9618 med=266.4019 max=1099.2501 p(90)=448.872 p(95)=554.0139
total_duration.....: avg=380.082968 min=235.1732 med=338.61115 max=1177.2031 p(90)=518.70804 p(95)=638.697285
waiting_time.....: avg=70.559768 min=60.2059 med=69.4442 max=214.2495 p(90)=77.0396 p(95)=80.372955

HTTP
http_req_duration.....: avg=380.08ms min=235.17ms med=338.61ms max=1.17s p(90)=518.7ms p(95)=638.69ms
{ expected_response:true }.....: avg=380.08ms min=235.17ms med=338.61ms max=1.17s p(90)=518.7ms p(95)=638.69ms
http_req_failed.....: 0.00% 0 out of 1454
http_reqs.....: 1454    12.003994/s

EXECUTION
iteration_duration.....: avg=1.38s min=1.23s med=1.34s max=2.17s p(90)=1.52s p(95)=1.63s
iterations.....: 1454    12.003994/s
vus.....: 1 min=1 max=30
vus_max.....: 30 min=30 max=30

NETWORK
data_received.....: 306 MB 2.5 MB/s
data_sent.....: 179 kB 1.5 kB/s
```

Результаты сжатием (gzip 9):

```

TOTAL RESULTS
checks_total.....: 1861    15.405309/s
checks_succeeded.....: 100.00% 1861 out of 1861
checks_failed.....: 0.00% 0 out of 1861

✓ status is 200

CUSTOM
receiving_time.....: avg=1.775705 min=0 med=1.7262 max=65.965 p(90)=2.7154 p(95)=3.0436
total_duration.....: avg=75.302597 min=63.0433 med=74.4137 max=139.9544 p(90)=81.0822 p(95)=87.0959
waiting_time.....: avg=73.489627 min=61.9236 med=72.6174 max=123.3855 p(90)=79.0806 p(95)=85.0538

HTTP
http_req_duration.....: avg=75.3ms min=63.04ms med=74.41ms max=139.95ms p(90)=81.08ms p(95)=87.09ms
{ expected_response:true }.....: avg=75.3ms min=63.04ms med=74.41ms max=139.95ms p(90)=81.08ms p(95)=87.09ms
http_req_failed.....: 0.00% 0 out of 1861
http_reqs.....: 1861    15.405309/s

EXECUTION
iteration_duration.....: avg=1.07s min=1.06s med=1.07s max=1.15s p(90)=1.08s p(95)=1.08s
iterations.....: 1861    15.405309/s
vus.....: 6 min=1 max=30
vus_max.....: 30 min=30 max=30

NETWORK
data_received.....: 11 MB 92 kB/s
data_sent.....: 229 kB 1.9 kB/s
```

Зависимость CPU load, Network load от уровня сжатия (Без switcher)

Технические характеристики сервера:

- CPU 1 vCPU
- RAM 2 GB
- Storage 20 GB
- Speed 1200 Mbps

Техническая характеристика моего ноутбука:

- CPU 4 ядра
- RAM 8 GB
- Speed 100 Mbps

Ограничение на скорость создано искусственно с помощью настройки роутера QoS, максимальная скорость около 200 Mb/s. Ограничение скорости гарантирует постоянную полосу пропускания. На сервере отключены все посторонние процессы. CPU используется только для:

- MySQL Database
- Backend server (Express.js)
- Обратный прокси сервер (Nginx)

Нагрузка создана с помощью k6:

```
stages: [  
  { duration: "5s", target: 10 },  
  { duration: "10s", target: 20 },  
  { duration: "10s", target: 30 },  
  { duration: "10s", target: 40 },  
  { duration: "10s", target: 50 },  
  { duration: "10s", target: 60 },  
  { duration: "10s", target: 70 },  
  { duration: "10s", target: 80 },  
  { duration: "10s", target: 90 },  
  { duration: "10s", target: 100 },  
  { duration: "10s", target: 110 },  
  { duration: "10s", target: 120 },  
  { duration: "10s", target: 130 },  
  { duration: "10s", target: 140 },  
  { duration: "10s", target: 150 },  
  { duration: "10s", target: 170 },  
  { duration: "10s", target: 190 },  
  { duration: "10s", target: 210 },  
  { duration: "5s", target: 5 },  
]
```

В базе данных находится 400 видео, для тестирования используется Get запрос:

[/video/getRecommendations/1](#), который возвращает 100 случайных видео из базы данных

Будет варьироваться степень сжатия:

- no compress

- gzip 1
- gzip 5
- gzip 9

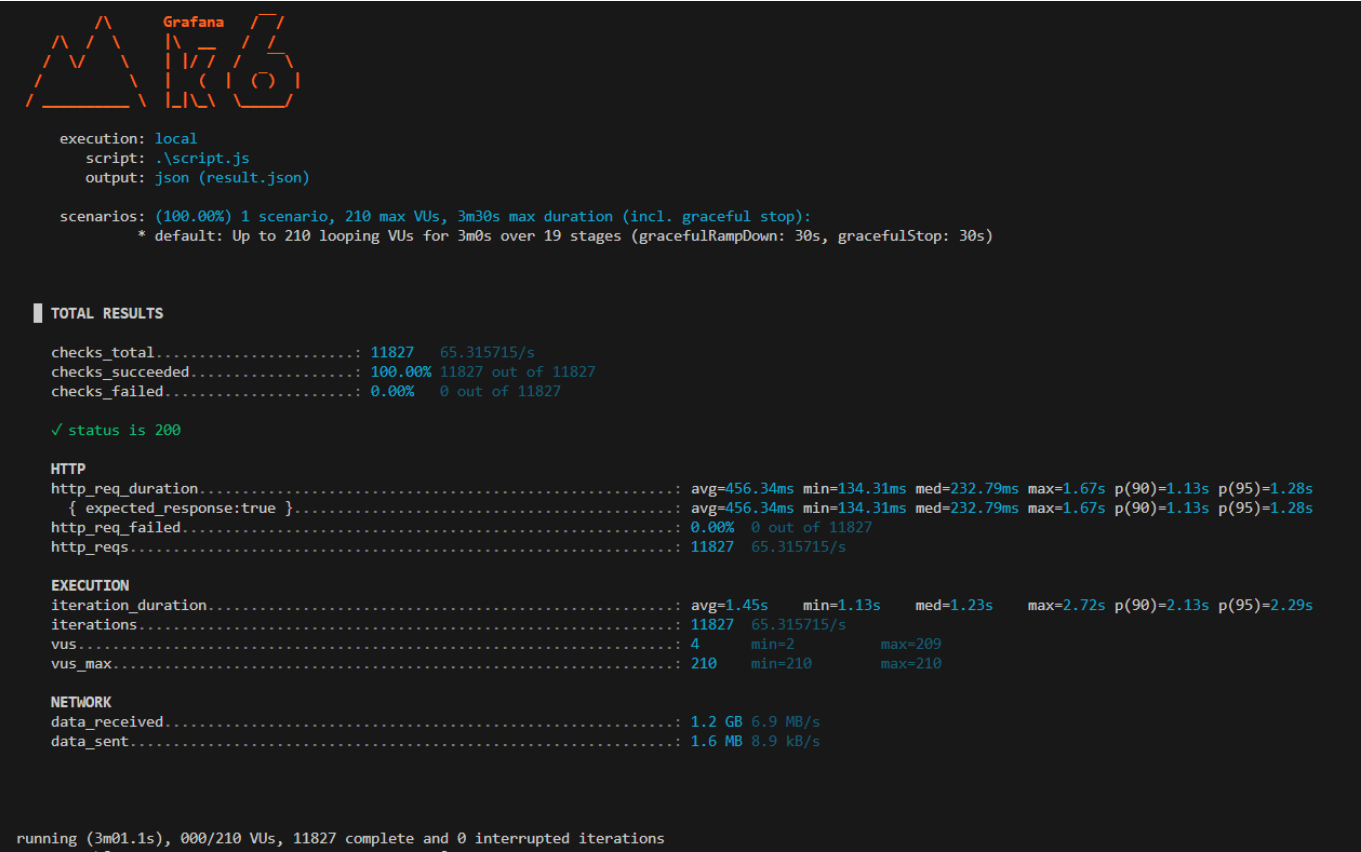
При построении графиков использовалось сглаживание по трём точкам

```
smoothing_x[i] = (x[i-1] + x[i] + x[i+1])/3
```

Построены графики от времени

Результаты измерений


No compress



Gzip level 1

```

PS D:\programming\diploma\Работа\Нагрузочное тестирование\stress_testing> k6 run .\script.js --out json=result.json



execution: local
script: .\script.js
output: json (result.json)

scenarios: (100.00%) 1 scenario, 210 max VUs, 3m30s max duration (incl. graceful stop):
    * default: Up to 210 looping VUs for 3m0s over 19 stages (gracefulRampDown: 30s, gracefulStop: 30s)

TOTAL RESULTS

checks_total.....: 12745    70.397576/s
checks_succeeded.....: 100.00% 12745 out of 12745
checks_failed.....: 0.00%    0 out of 12745

✓ status is 200

HTTP
http_req_duration.....: avg=350.55ms min=47.59ms med=146.16ms max=1.27s p(90)=994.46ms p(95)=1.07s
{ expected_response:true }.....: avg=350.55ms min=47.59ms med=146.16ms max=1.27s p(90)=994.46ms p(95)=1.07s
http_req_failed.....: 0.00%    0 out of 12745
http_reqs.....: 12745    70.397576/s

EXECUTION
iteration_duration.....: avg=1.35s    min=1.04s    med=1.15s    max=2.31s p(90)=1.99s    p(95)=2.07s
iterations.....: 12745    70.397576/s
vus.....: 1    min=1    max=209
vus_max.....: 210    min=210    max=210

NETWORK
data_received.....: 57 MB    316 kB/s
data_sent.....: 1.7 MB    9.6 kB/s

```

## Gzip level 5

```

● PS D:\programming\diplo\Работа\Нагрузочное тестирование\stress_testing> k6 run .\script.js --out json=result.json

      / \
     /   \
    /_____\
    |       |
    |  Grafana  |
    |_____|___| |
    |   |   |   |
    |   |   |   |
    |___|___|___|

execution: local
script: .\script.js
output: json (result.json)

scenarios: (100.00%) 1 scenario, 210 max VUs, 3m30s max duration (incl. graceful stop):
  * default: Up to 210 looping VUs for 3m0s over 19 stages (gracefulRampDown: 30s, gracefulStop: 30s)


TOTAL RESULTS

checks_total.....: 12338    68.171548/s
checks_succeeded.....: 100.00% 12338 out of 12338
checks_failed.....: 0.00%    0 out of 12338

✓ status is 200

HTTP
http_req_duration.....: avg=396.7ms min=47.79ms med=157ms max=1.57s p(90)=1.05s p(95)=1.21s
  { expected_response:true }.....: avg=396.7ms min=47.79ms med=157ms max=1.57s p(90)=1.05s p(95)=1.21s
http_req_failed.....: 0.00%    0 out of 12338
http_reqs.....: 12338    68.171548/s

EXECUTION
iteration_duration.....: avg=1.39s min=1.04s med=1.17s max=2.57s p(90)=2.06s p(95)=2.21s
iterations.....: 12338    68.171548/s
vus.....: 2 min=2 max=209
vus_max.....: 210 min=210 max=210

NETWORK
data_received.....: 48 MB 266 kB/s
data_sent.....: 1.7 MB 9.3 kB/s

```

## Gzip level 9

```
PS D:\programing\diploma\Работа\Нагрузочное тестирование\stress_testing> k6 run .\script.js --out json=result.json

Grafana

execution: local
script: .\script.js
output: json (result.json)

scenarios: (100.00%) 1 scenario, 210 max VUs, 3m30s max duration (incl. graceful stop):
  * default: Up to 210 looping VUs for 3m0s over 19 stages (gracefulRampDown: 30s, gracefulStop: 30s)

TOTAL RESULTS

checks_total.....: 12214 67.489938/s
checks_succeeded.....: 100.00% 12214 out of 12214
checks_failed.....: 0.00% 0 out of 12214

✓ status is 200

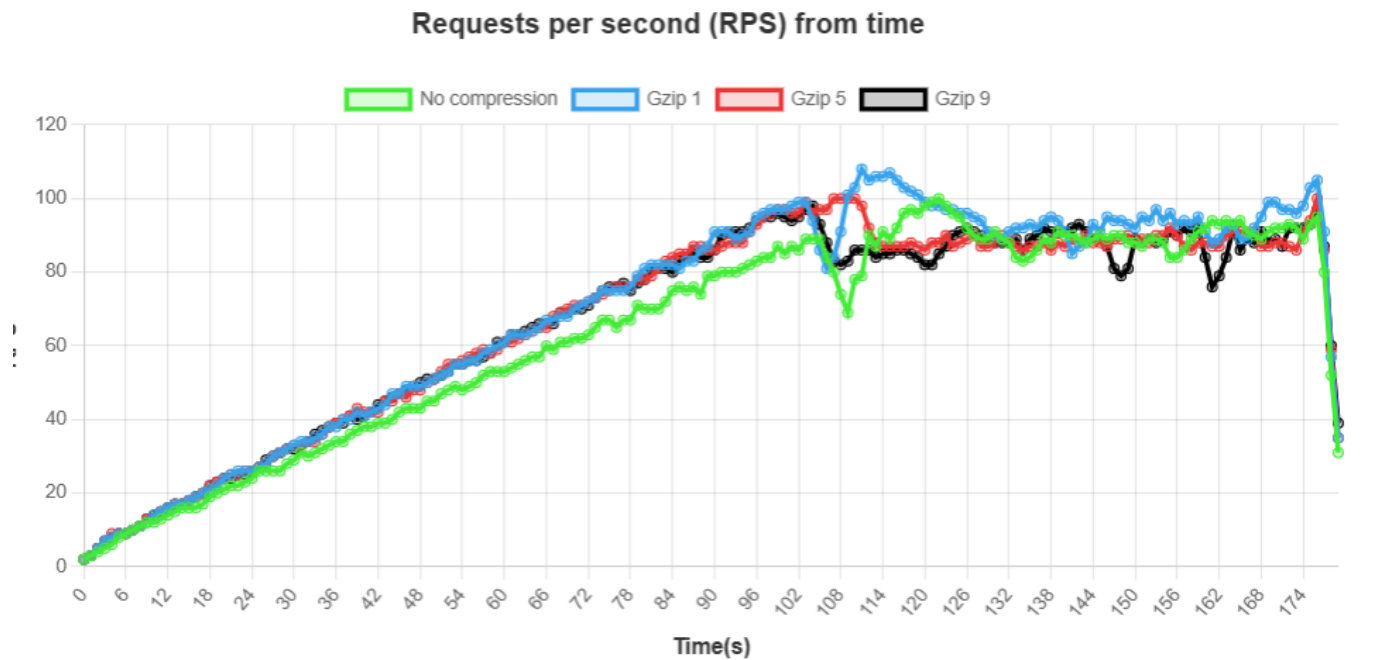
HTTP
http_req_duration.....: avg=411.31ms min=44.38ms med=338.54ms max=1.52s p(90)=1.12s p(95)=1.23s
  { expected_response:true }.....: avg=411.31ms min=44.38ms med=338.54ms max=1.52s p(90)=1.12s p(95)=1.23s
http_req_failed.....: 0.00% 0 out of 12214
http_reqs.....: 12214 67.489938/s

EXECUTION
iteration_duration.....: avg=1.41s min=1.04s med=1.33s max=2.56s p(90)=2.12s p(95)=2.23s
iterations.....: 12214 67.489938/s
vus.....: 1 min=1 max=209
vus_max.....: 210 min=210 max=210

NETWORK
data_received.....: 45 MB 248 kB/s
data_sent.....: 1.7 MB 9.2 kB/s

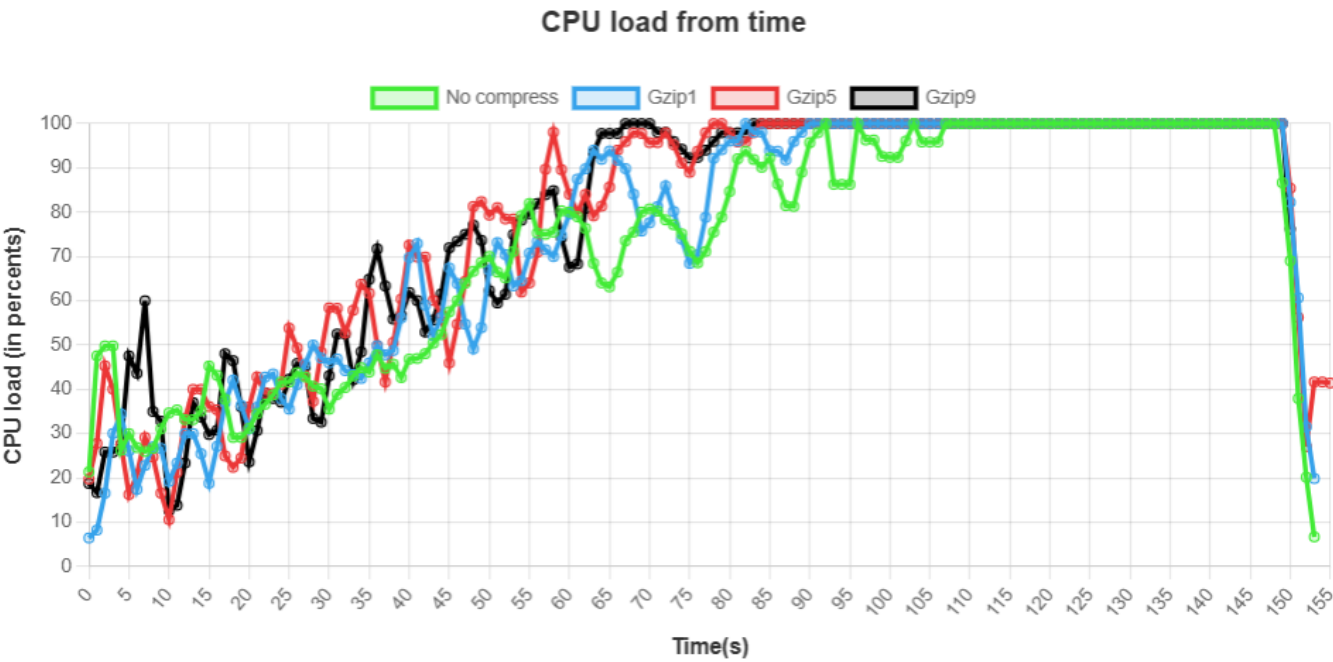
running (3m01.0s), 000/210 VUs, 12214 complete and 0 interrupted iterations
```

Requests per seconds

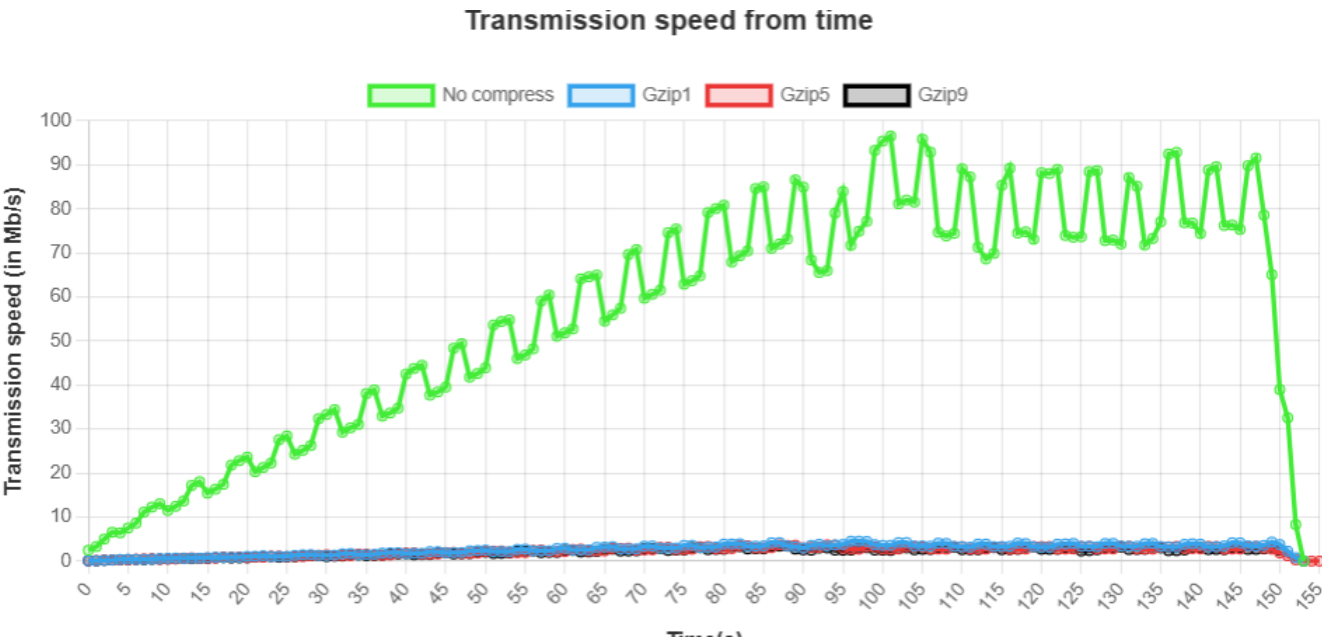


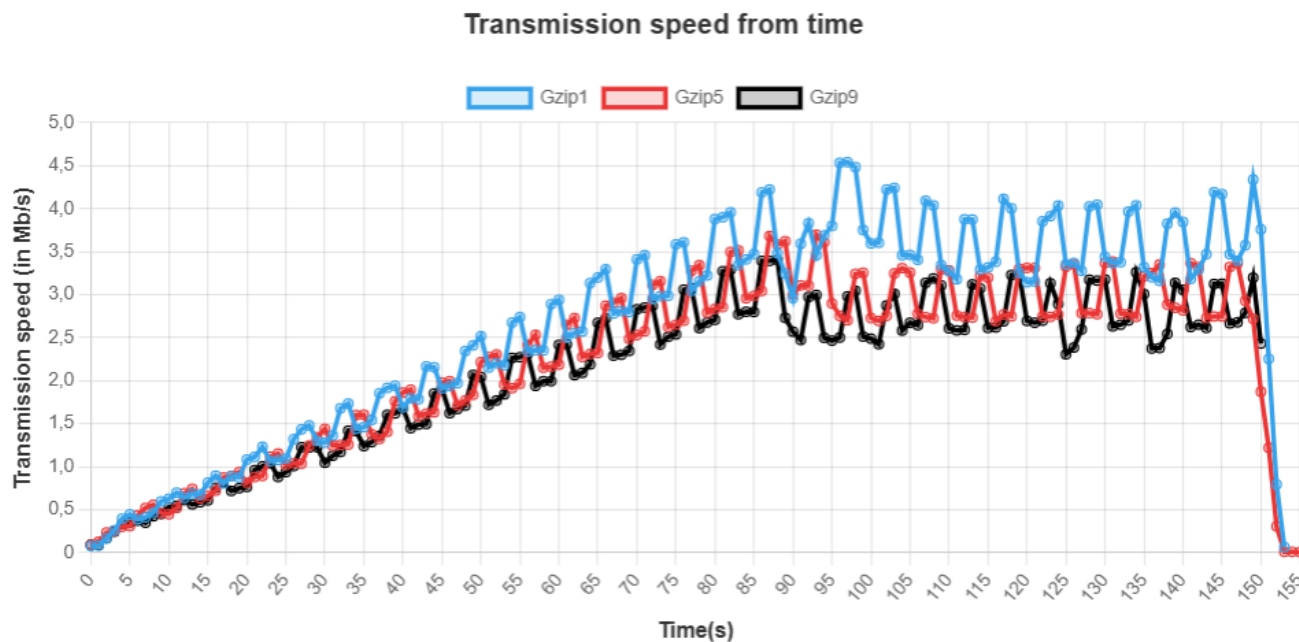
Cpu load





Transmission speed





## Анализ полученных результатов

Как видно, использование gzip заметно снизило нагрузку на сеть примерно в 20-30 раз, при этом даже при 80 МБ/с несжатого трафика, нагрузка на CPU из-за сжатия оказалась крайне малой, что при загрузке CPU на 100%, показатель RPS остался на том же уровне.