



API 서버 성능 개선기



Outsider
2017.11.09 @ play.node

얘기할 내용

- ✓ 성능 테스트를 위한 준비
- ✓ 성능 테스트의 도구
- ✓ 성능 테스트의 결과 분석

얘기안할 내용

- ✓ 코드의 성능 개선
- ✓ 코드 작성 팁
- ✓ 잘못된 코드

인증 API 서버



인증 API 서버

- ✓ 회원 가입
- ✓ 로그인
- ✓ 권한부여
- ✓ 회원 정보 관리

성능 테스트 #237

ⓘ Open

outsider opened this issue on 26 Apr · 3 comments



outsider commented on 26 Apr

Owner



API에 대한 전체 성능 테스트를 하고 느린 API와 병목구간을 찾아야 합니다.



Why performance testing?

- ✓ 서버 하나의 한계 파악
- ✓ 병목구간 확인
- ✓ 코드 개선 후 비교

Unit test에서도 확인 가능하지만
실제 트래픽과 유사한
성능이 궁금했다.

성능 테스트 도구



- ✓ 사용자 시나리오로 작성
- ✓ 대량 트래픽 조절 가능
- ✓ 가능하면 Node.js로 사용

ARTILLERY.IO

<https://artillery.io/>

Node.js 작성된 부하 테스트 도구

YML

```
config:
  target: 'http://localhost:3000'
  http:
    timeout: 20
phases:
  -
    duration: 240
    arrivalCount: 8
    name: "Warm-up"
  -
    duration: 240
    arrivalCount: 24
  -
    duration: 240
    arrivalCount: 48
  -
    duration: 600
    arrivalCount: 150
  -
    duration: 240
    arrivalCount: 48
  -
    duration: 240
    arrivalCount: 24
processor: "./processor.js"
payload:
  path: './payload.csv'
  fields:
    - 'email'
    - 'password'
order: 'sequence'
```

```
duration: 240  
arrivalCount: 24
```

240초동안 24 유저(시나리오)를 생성한다.

-> 4분동안 10초에 한명씩 새로운 유저를 생성한다.

```
scenarios:
  - name: '사용자 흐름'
    flow:
      # 회원 가입
      - post:
          url: '/signup'
          json:
            email: '{{ email }}'
            password: '{{ password }}'
      # 로그인
      - post:
          url: '/login'
          json:
            email: '{{ email }}'
            password: '{{ password }}'
          capture:
            - json: '$.data.token'
              as: 'token'
      # 정보조회
      - get:
          url: '/user/info'
          headers:
            Authorization: 'Bearer {{ token }}'
```

테스트 데이터 생성

```
#!/usr/bin/env node
const fs = require('fs');

const ROWS = process.env.PAYLOAD || 10000;
const FILE_NAME = './test/payload.csv'

fs.writeFileSync(FILE_NAME, '');

for(let i = 0; i < ROWS; i++) {
  const u = generateUser();

  fs.appendFileSync(FILE_NAME, `${u.email},${u.password}\n`);
}
```

package.json

```
{  
  "scripts": {  
    "preloadtest": "./test/generate-payloads",  
    "loadtest": "artillery run test/config.yml"  
  }  
}
```

package.json

```
{  
  "scripts": {  
    "preloadtest": "./test/generate-payloads",  
    "loadtest": "artillery run test/config.yml"  
  }  
}
```

```
$ npm run loadtest
```



Test run on 03 Nov 2017 18:45:95

Summary

Test duration	2300 sec
Scenarios created	322
Scenarios completed	142

Scenario counts:

사용자 흐름	322 (100%)
--------	------------

Codes

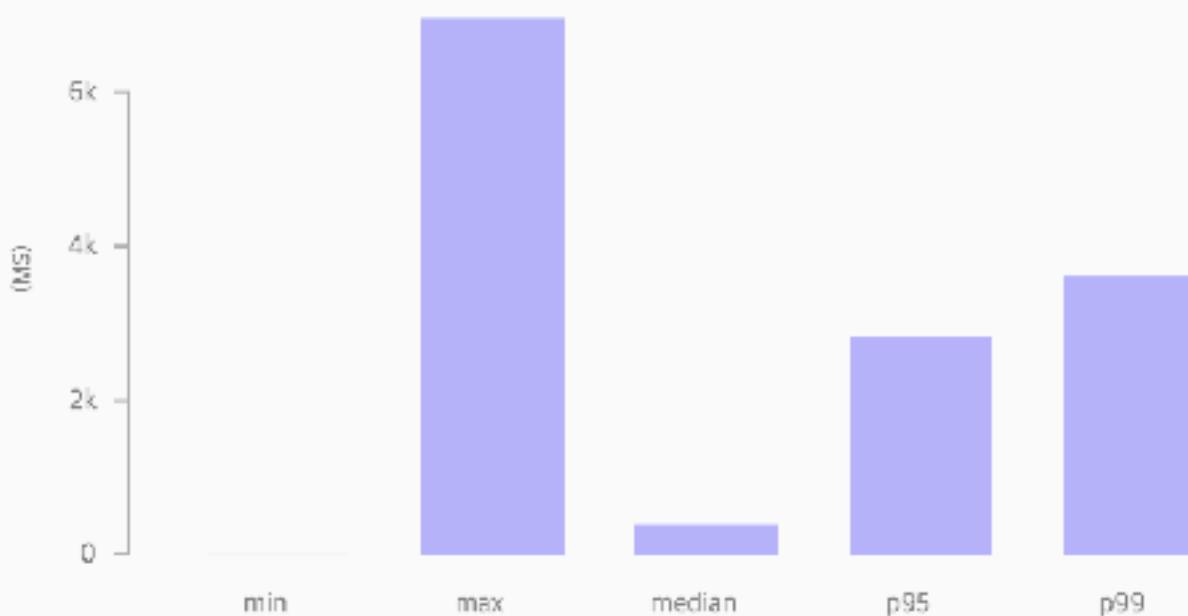
200	113333
201	3082
202	3020
401	275
404	21
409	1
422	2

Errors

ESOCKETTIMEDOUT	360
-----------------	-----

Charts

Overall Latency Distribution ?





Test run on 03 Nov 2017 18:45:95

Summary

Test duration	2300 sec
Scenarios created	322
Scenarios completed	142

Scenario counts:

사용자 흐름	322 (100%)
--------	------------

Codes

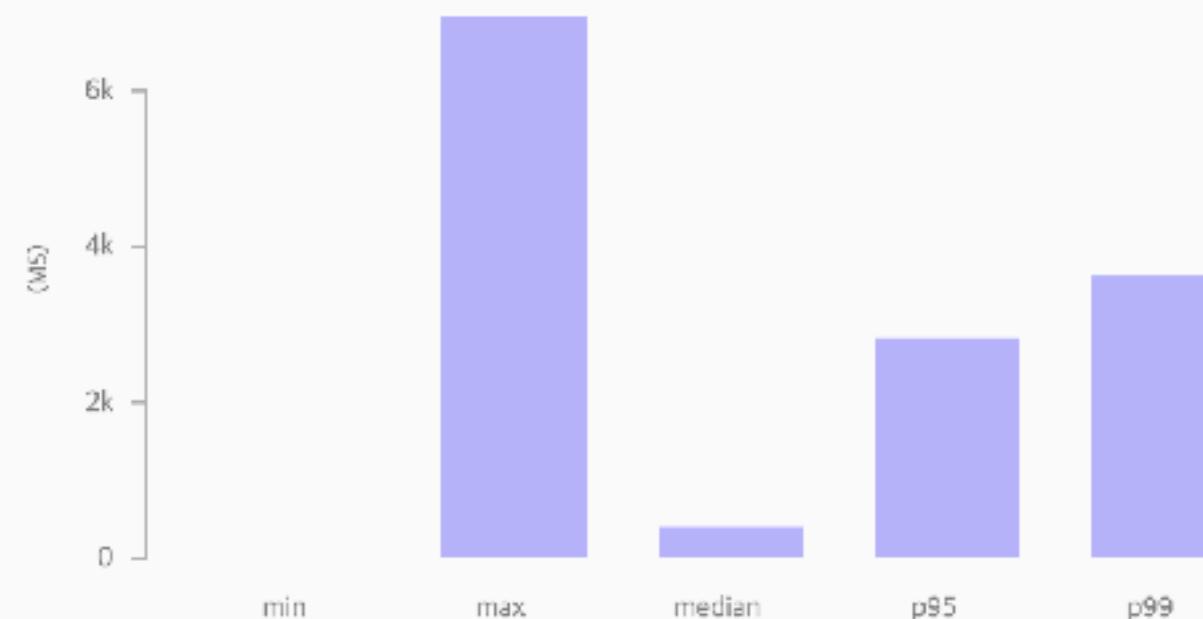
200	113333
201	3082
202	3020
401	275
404	21

Errors

ESOCKETTIMEDOUT	360
-----------------	-----

```
$ artillery report \
  artillery_report_20171103_185907.json
```

Overall Latency Distribution ?





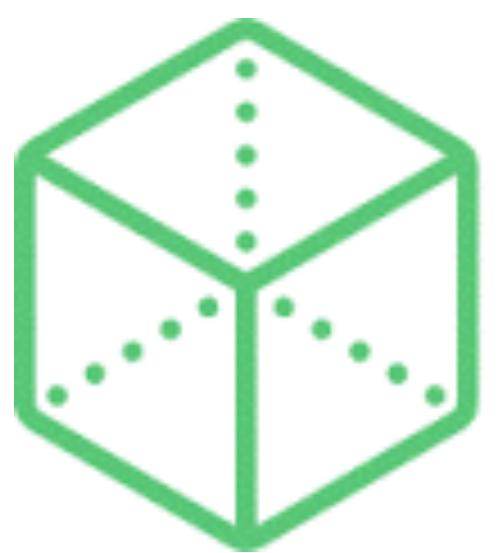
WHAT ARE
YOU
LOOKING AT?

APM

: Application Performance Management



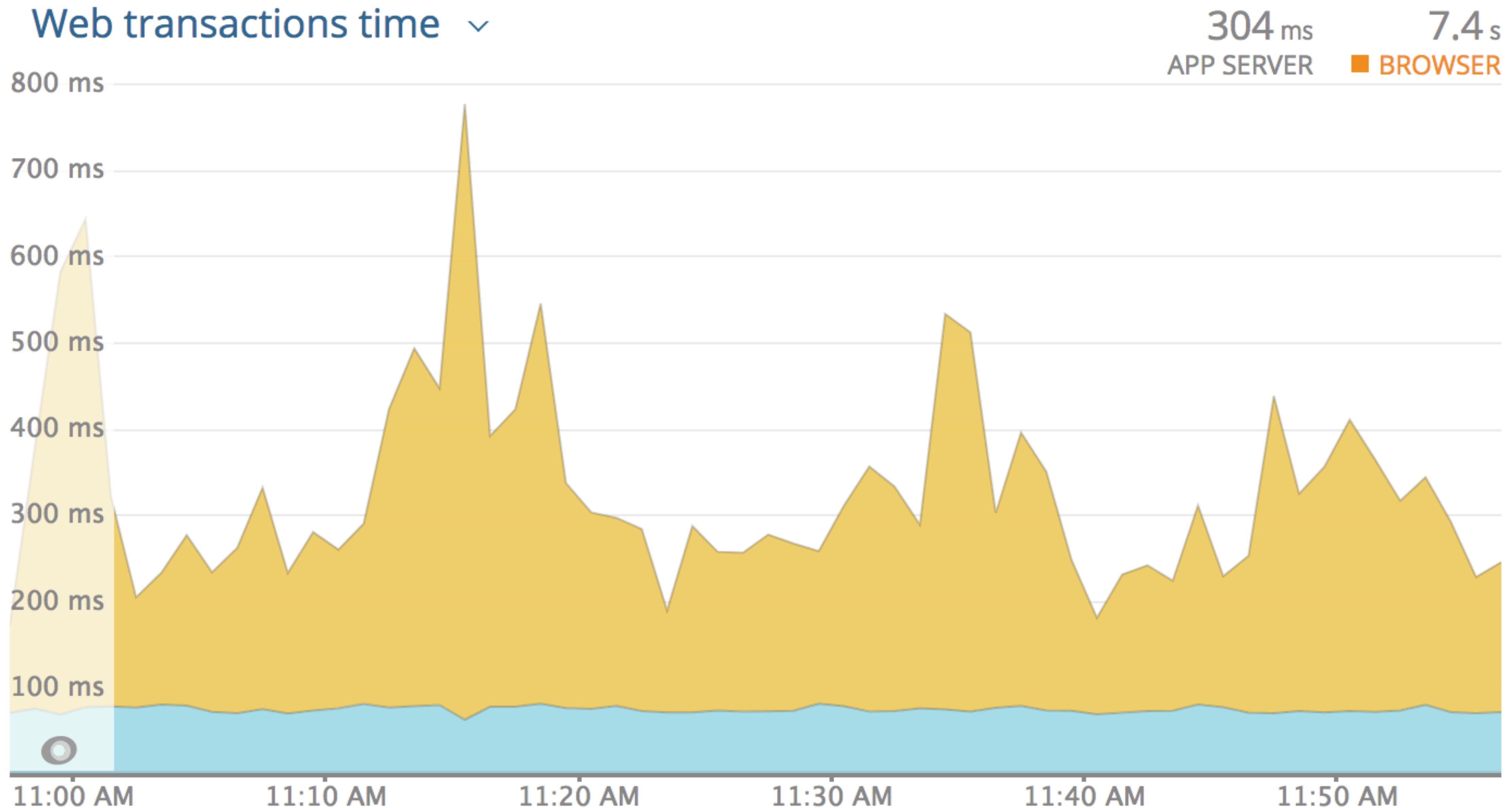
RisingStack



N | Solid™

대부분 유료

Web transactions time ▾



New relic 배포 플래그

```
#!/usr/bin/env bash

curl -X POST \
  'https://api.newrelic.com/v2/applications/APPID/deployments.json' \
-H 'X-Api-Key:YOUR_API_KEY' -i \
-H 'Content-Type: application/json' \
-d \
'{
  "deployment": {
    "revision": "1",
    "changelog": "'${MACHINE_ID} start'",
    "description": "'${MACHINE_ID} start'",
    "user": "YOUR_EMAIL"
  }
}'
```

Web transactions time

2250 ms

2000 ms

1750 ms

1500 ms

1250 ms

1000 ms

750 ms

500 ms

250 ms



오후 10:30

오후 10:40

오후 10:50

오후 11:00

오후 11:10

오후 11:20

node

Postgres

Response time

886 ms

APP SERVER

Deployment at Today, 22:35
Deployed by: outsideris@gmail.com
Revision: 1
start

A

1

0

C

T

61

40

20

S

Add to an Insights dashboard

Add to note

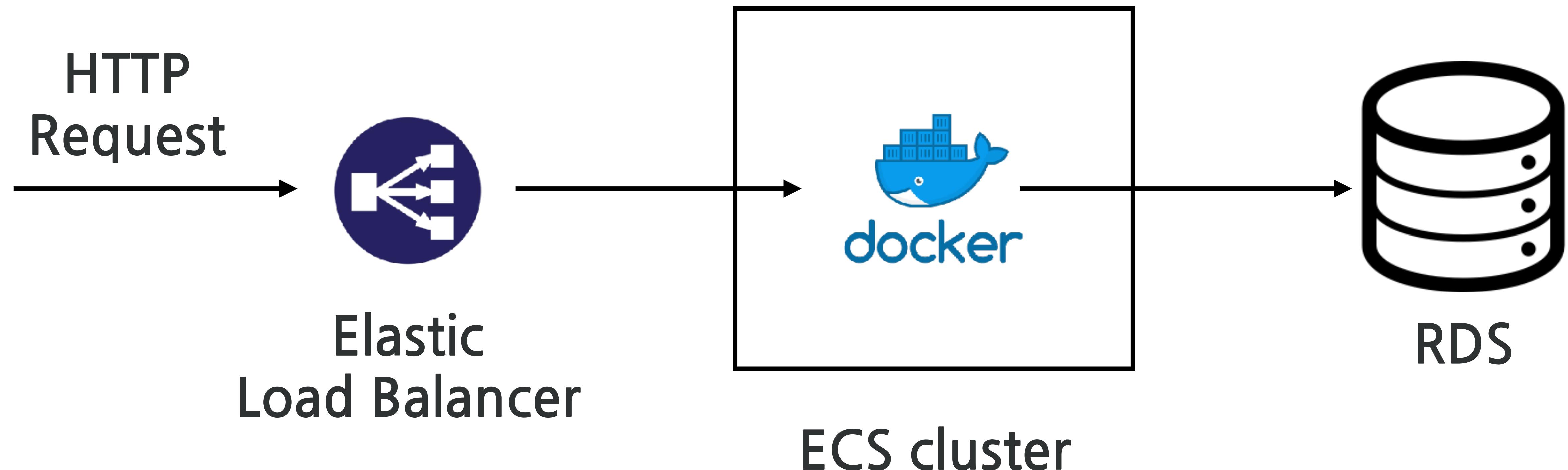
package.json

```
{  
  "scripts": {  
    "preloadtest": "./test/generate-payloads && ./test/mark-ended.sh",  
    "loadtest": "artillery run test/config.yml",  
    "portloadtest": "./test/mark-ended.sh"  
  }  
}
```

테스트 서버 구성



타겟 서버





HashiCorp

Vagrant



Vagrantfile

```
#!/bin/env ruby

Vagrant.configure(2) do |config|
  config.vm.box = 'dummy'

  TEST_MACHINE_COUNT = 1
  1.upto(TEST_MACHINE_COUNT) do |i|
    config.vm.define "test-machine-#{i}" do |machine|

      machine.vm.provider :aws do |aws, override|
        aws.tags = { 'Name' => "test-machine-#{i}" }

        aws.access_key_id = ENV['AWS_ACCESS_KEY_ID']
        aws.secret_access_key = ENV['AWS_SECRET_ACCESS_KEY']
        aws.keypair_name = 'test'
        aws.subnet_id = 'subnet-xxxxxx'
        aws.instance_type = 'c4.xlarge'
        aws.region = 'ap-northeast-1'
        aws.ami = 'ami-ea4eae8c' # Ubuntu 16.04
        aws.security_groups = ['sg-xxxxxx', 'sg-xxxxxx']

        override.ssh.username = 'ubuntu'
        override.ssh.private_key_path = '~/.ssh/test.pem'
        override.ssh.pty = false
        override.ssh.shell = "bash -c 'BASH_ENV=/etc/profile exec bash'"
      end

      machine.nfs.functional = false
      machine.vm.synced_folder '../', '/www', type: "rsync", rsync__exclude: "node_modules/"

      machine.vm.provision "shell" do |s|
        s.inline = "export MACHINE_ID=$1 && sudo apt-get update && sudo apt-get install -y python"
        s.args = ["test-machine-#{i}"]
      end

      machine.vm.provision "ansible" do |ansible|
        ansible.groups = {
          "testnode" => ["test-machine-[0:#{$TEST_MACHINE_COUNT}]"]
        }
        ansible.playbook = './playbook.yml'
      end
    end
  end
end
```

```
machine.vm.provider :aws do |aws, override|
  aws.tags = { 'Name' => "test-machine-#{i}" }

  aws.access_key_id = ENV['AWS_ACCESS_KEY_ID']
  aws.secret_access_key = ENV['AWS_SECRET_ACCESS_KEY']
  aws.keypair_name = 'test'
  aws.subnet_id = 'subnet-xxxxxx'
  aws.instance_type = 'c4.xlarge'
  aws.region = 'ap-northeast-1'
  aws.ami = 'ami-ea4eae8c' # Ubuntu 16.04
  aws.security_groups = ['sg-xxxxxx', 'sg-xxxxxx']

  override.ssh.username = 'ubuntu'
  override.ssh.private_key_path = '~/.ssh/test.pem'
  override.ssh.pty = false
  override.ssh.shell = "bash -c 'BASH_ENV=/etc/profile exec bash'"
end
```

```
machine.vm.synced_folder '../..', '/www',
  type: "rsync", rsync__exclude: "node_modules/"

machine.vm.provision "shell" do |s|
  s.inline = "export MACHINE_ID=$1 && sudo apt-get update && sudo
apt-get install -y python"

  s.args = ["test-machine-#{i}"]
end
```

```
machine.vm.provision "ansible" do |ansible|
  ansible.groups = {
    "testnode" => ["test-machine-[0:#{$TEST_MACHINE_COUNT}]"]
  }
  ansible.playbook = './playbook.yml'
end
```

ansible: playbook.yml

```
- name: Install prerequisites
  become: yes
  hosts:
    - all
  tasks:
    - name: install node.js
      shell: curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -
    - name: install apt packages
      apt:
        name: "{{ item }}"
        update_cache: yes
      with_items:
        - nodejs
    - name: install "artillery" nodejs package
      npm:
        name: artillery
        version: '1.5.8-3'
        path: /www
```

플러그인 설치

```
$ vagrant plugin install vagrant-aws
```

서버 실행

```
$ vagrant up
```

부하 테스트 실행

```
$ ansible all -m shell -a "cd /www && npm run loadtest"
```

서버 종료

```
$ vagrant destroy
```

A photograph of a man sitting on a dark-colored couch. He is wearing a dark t-shirt and plaid pajama bottoms. He has his head in his hands, appearing to be in distress or exhaustion. The background shows a wall with a subtle floral or paisley pattern.

생각보다 서버 성능이...

- ✓ 클러스터 인스턴스 변경
- ✓ 컨테이너 CPU, 메모리 조정
- ✓ DB Pool 사이즈 조정

테스트 결과

```
scenarios:
  - name: '사용자 흐름'
    flow:
      # 회원 가입
      - post:
          url: '/signup'
          json:
            email: '{{ email }}'
            password: '{{ password }}'
        # 로그인
        - post:
            url: '/login'
            json:
              email: '{{ email }}'
              password: '{{ password }}'
            capture:
              - json['data.token']
              as: 'token'
        # 정보 조회
        - get:
            url: '/user/info'
            headers:
              Authorization: 'Bearer {{ token }}'
        # 정보 갱신
        - patch:
            url: '/user/info'
            json:
              field1: 'blah blah'
              field2: 'blah blah'
```

1 시나리오

총 890 요청

200 응답: 818

201 응답: 32

202 응답: 40

시나리오당 20초 정도

```
scenarios:
```

```
- name: '사용자 흐름'
```

- 4분간 60초마다 1 유저

```
# 회원 가입
```

```
- post:
```

```
    url: '/signup'
```

```
    json:
```

```
        email: '{{email}}'
```

```
        password: '{{password}}'
```

```
# 로그인
```

```
- post:
```

```
    url: '/login'
```

```
    json:
```

```
        email: '{{email}}'
```

```
        password: '{{password}}'
```

```
capture:
```

```
- json: '$.data.token'
```

```
as: 'token'
```

- 4분간 20초마다 1 유저

- 4분간 10초마다 1 유저

- 6분간 6초마다 1 유저

```
capture:
```

```
- json: '$.data.token'
```

```
as: 'token'
```

- 10분간 4초마다 1 유저

```
# 상조회
```

```
- get:
```

```
    url: '/user/info'
```

```
headers:
```

```
    Authorization: Bearer {{token}}
```

```
# 정보 갱신
```

- 4분간 10초마다 1 유저

```
- patch:
```

```
    url: '/user/info'
```

```
headers:
```

```
    Authorization: Bearer {{token}}
```

```
json:
```

```
    field1: 'blah blah'
```

```
    field2: 'blah blah'
```

- 4분간 20초마다 1 유저

node.js 버전 업 #121

Closed

outsider opened this issue on 9 Feb · 4 comments



outsider commented on 9 Feb

Owner



현재 Node.js v6.9.x를 쓰고 있는데 4월에 v8.x가 나올 예정이고 v8이 10월에 LTS가 될 예정입니다.
오픈 시기를 생각하면 v7.x로 작업을 미리 해두고 v8.x로 바꿀 준비를 해놓으면 어떨까 합니다.
소스에서 충돌날 건 없어보이는데 사용하는 모듈의 지원여부는 확인해 봐야 합니다.
코드가 좀 더 올라간 시점에서 v7.x로의 전환을 테스트할 예정입니다.



outsider modified the milestone: API 1차 개발 on 9 Feb



outsider added this to the 개발후 QA milestone on 3 Mar



outsider modified the milestone: 2차 QA, 1차 QA on 12 Apr



outsider commented on 31 May

Owner



8.0.0이 나왔습니다. 오픈전에는 8.0.0으로 넘어갔으면 하지만....
한편으로는 6.10에서 성능 테스트 돌려보고 8.0.0으로 올려서 비교해 보고픈 (개인적) 욕심도 있습니다.



raccoony commented on 31 May

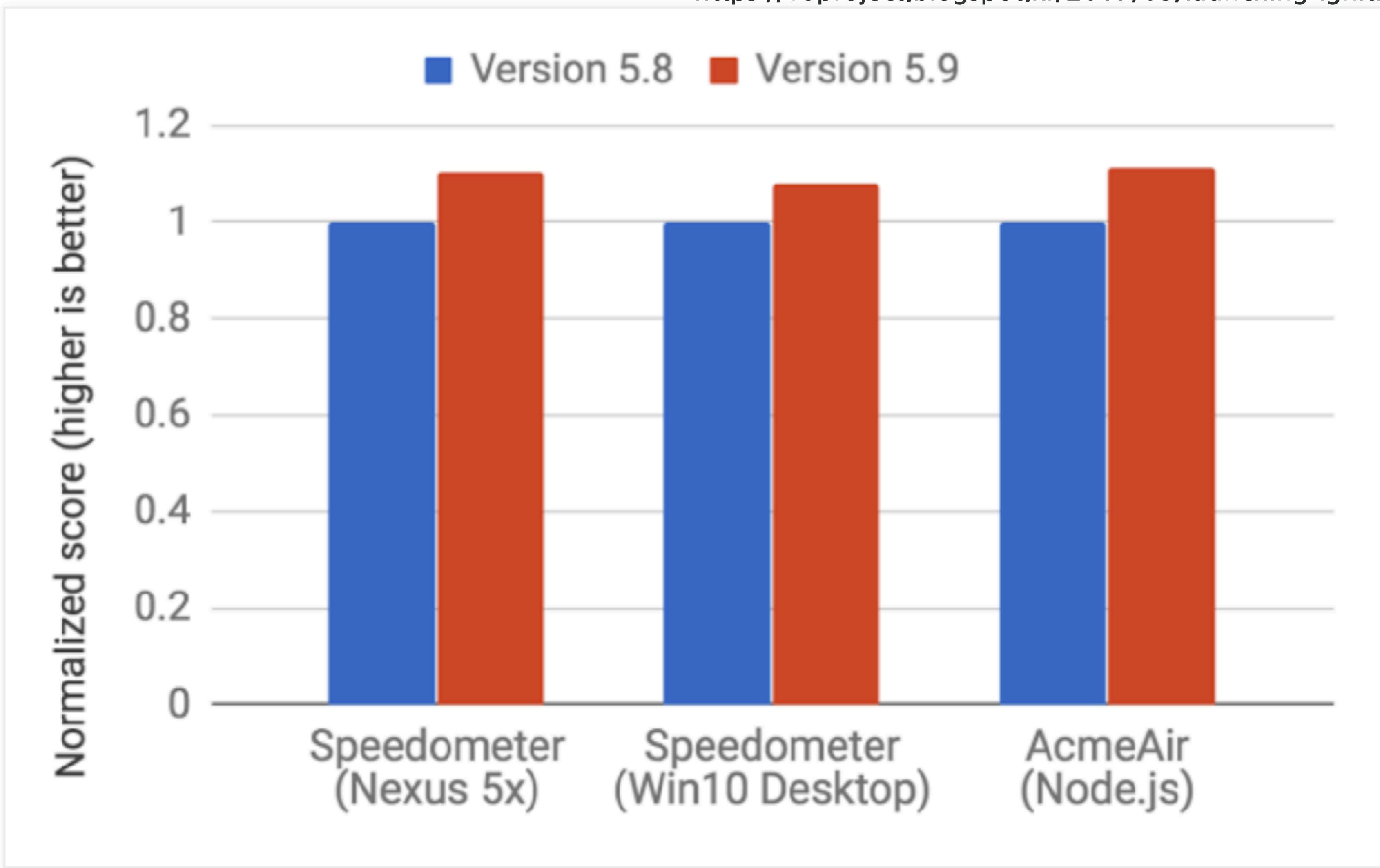
Owner



욕심은 채워야죠!

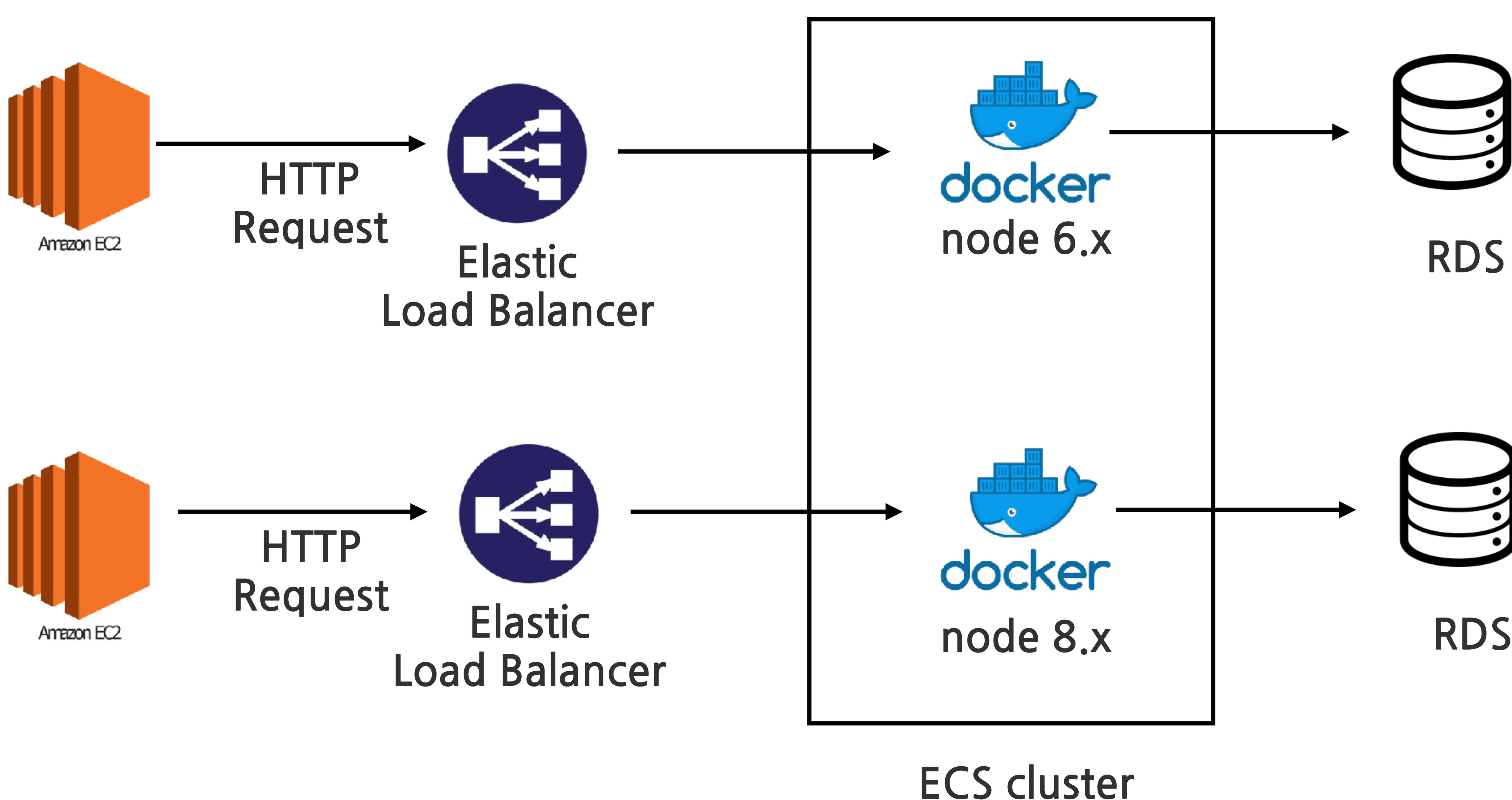
node.js	v8
6.11.5	5.1.281.108
8.9.0	6.1.534.46

V8의 컴파일러는 Crankshaft
5.9부터는 Ignition + Turbofan

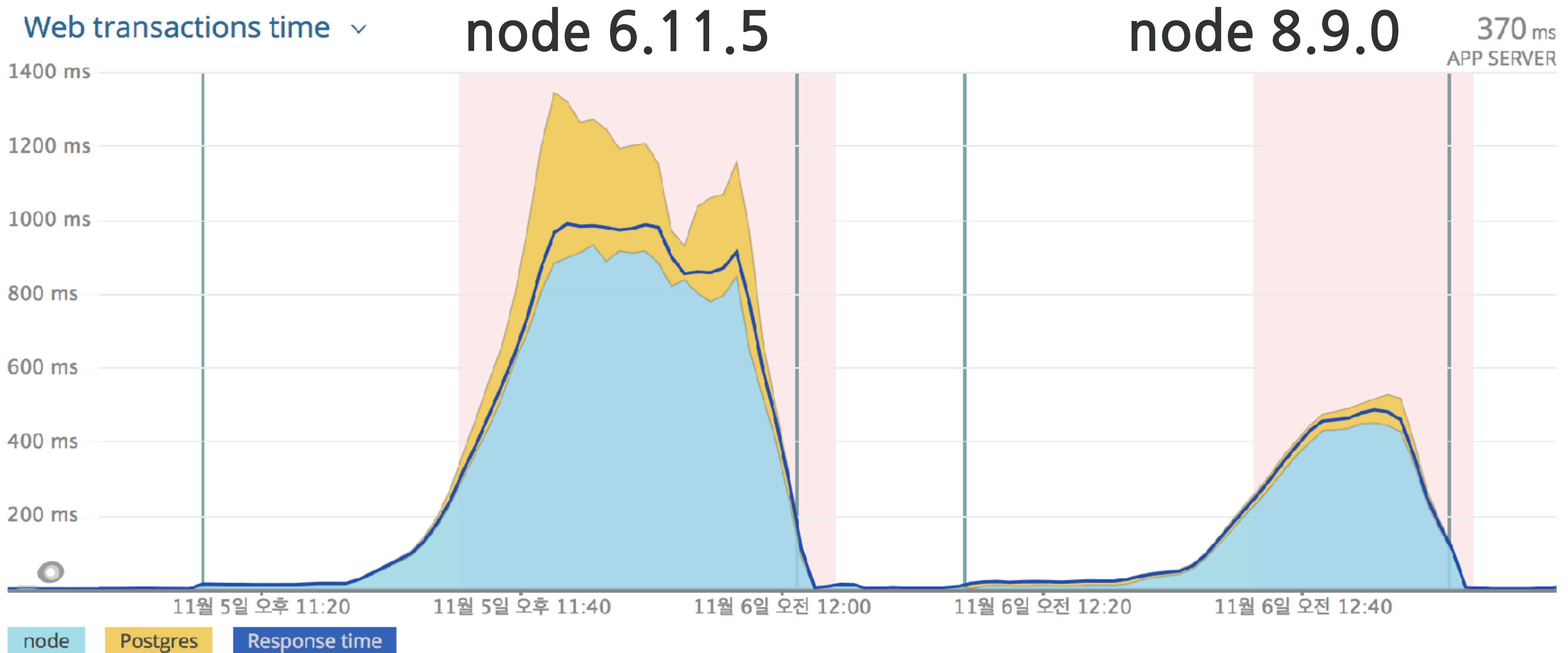


Improvements on Web and Node.js benchmarks

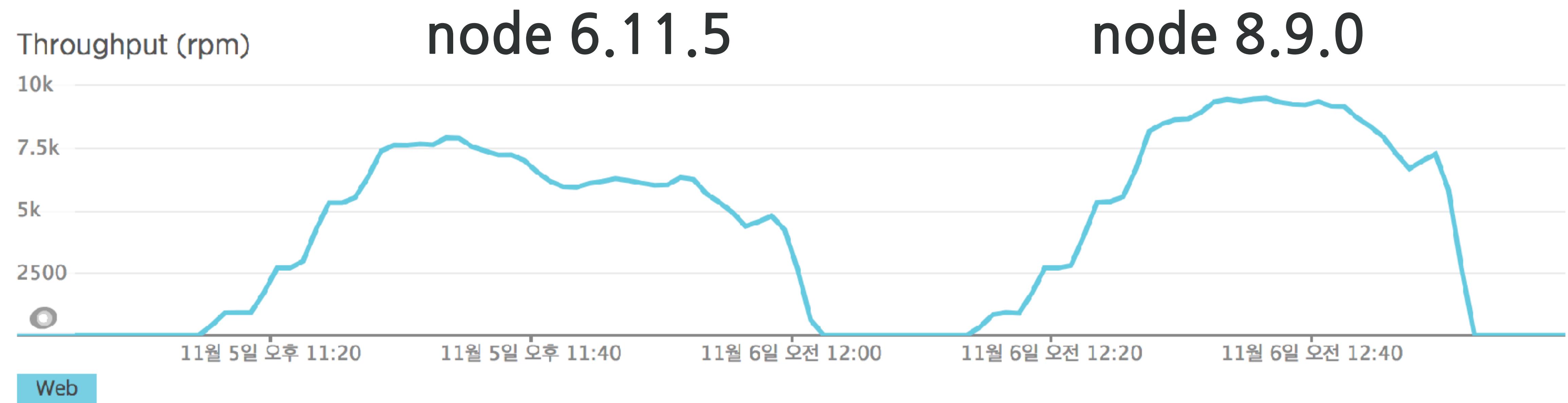
10% 이상의 성능 향상



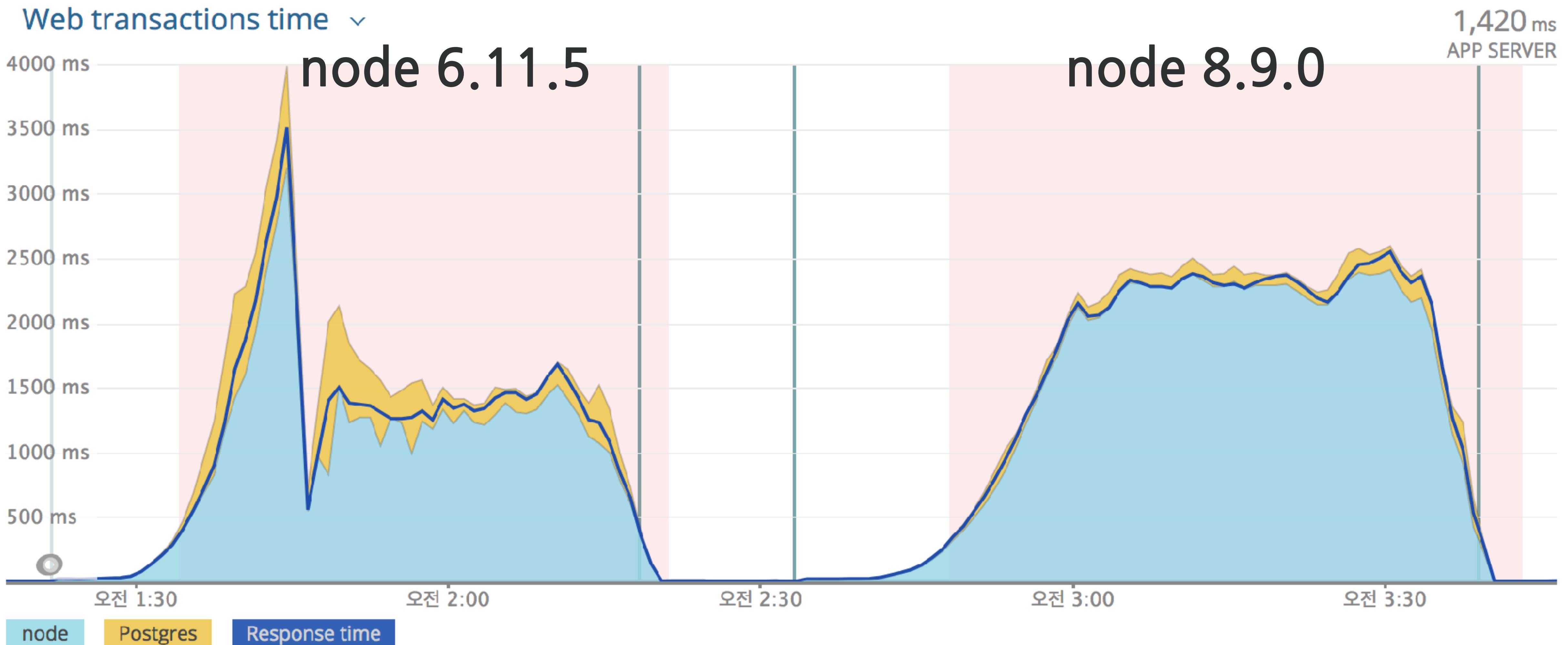
테스트 서버 1대



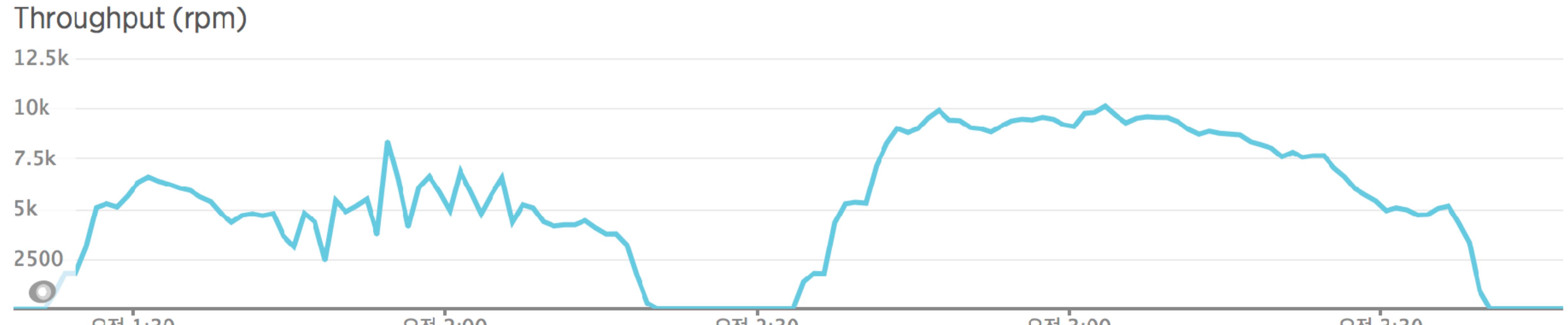
테스트 서버 1대



테스트 서버 2대



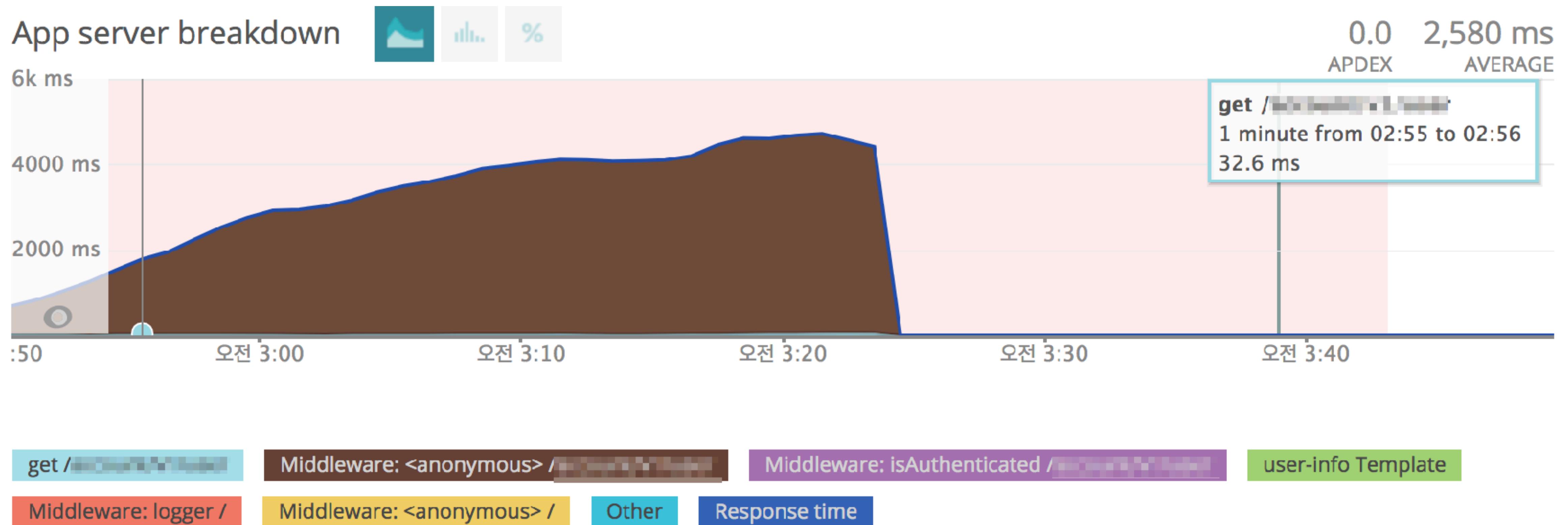
테스트 서버 1대



node 6.11.5

node 8.9.0

성능 병목 구간



유료 기능.....

Recent Slow Transactions

Resp. Time

Available only in New Relic Pro, transaction traces provide an at-a-glance, sortable list of slow transactions that lets you quickly zero in on trouble spots. [More about transaction traces »](#) [Upgrade to New Relic Pro now »](#)

프로파일링

node --inspect YOUR_APP.js

```
$ node --inspect ./bin/www
Debugger listening on ws://127.0.0.1:9229/77128b6e-fe32-4da9-a1db
For help see https://nodejs.org/en/docs/inspector
Debugger attached.
```

chrome://inspect

← → ⌂ Chrome | chrome://inspect/#devices

DevTools

- Devices
- Pages
- Extensions
- Apps
- Shared workers
- Service workers
- Other

Devices

Discover USB devices [Port forwarding...](#)

Discover network targets [Configure...](#)

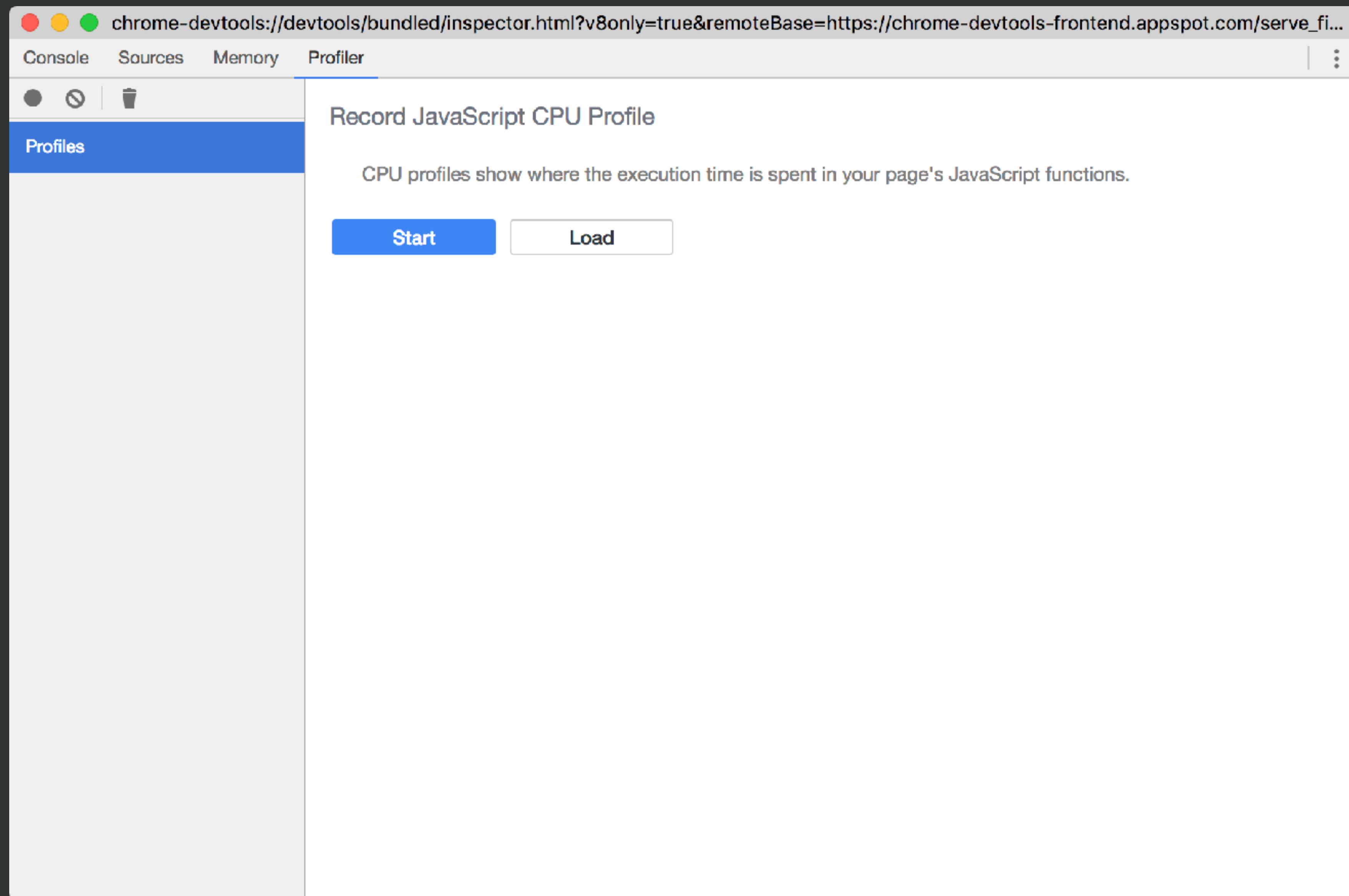
[Open dedicated DevTools for Node](#)

Remote Target #LOCALHOST

Target (v8.9.0)

 [./bin/api](#) file:///Users/outsider/smartsstudy/inspect

chrome-devtools



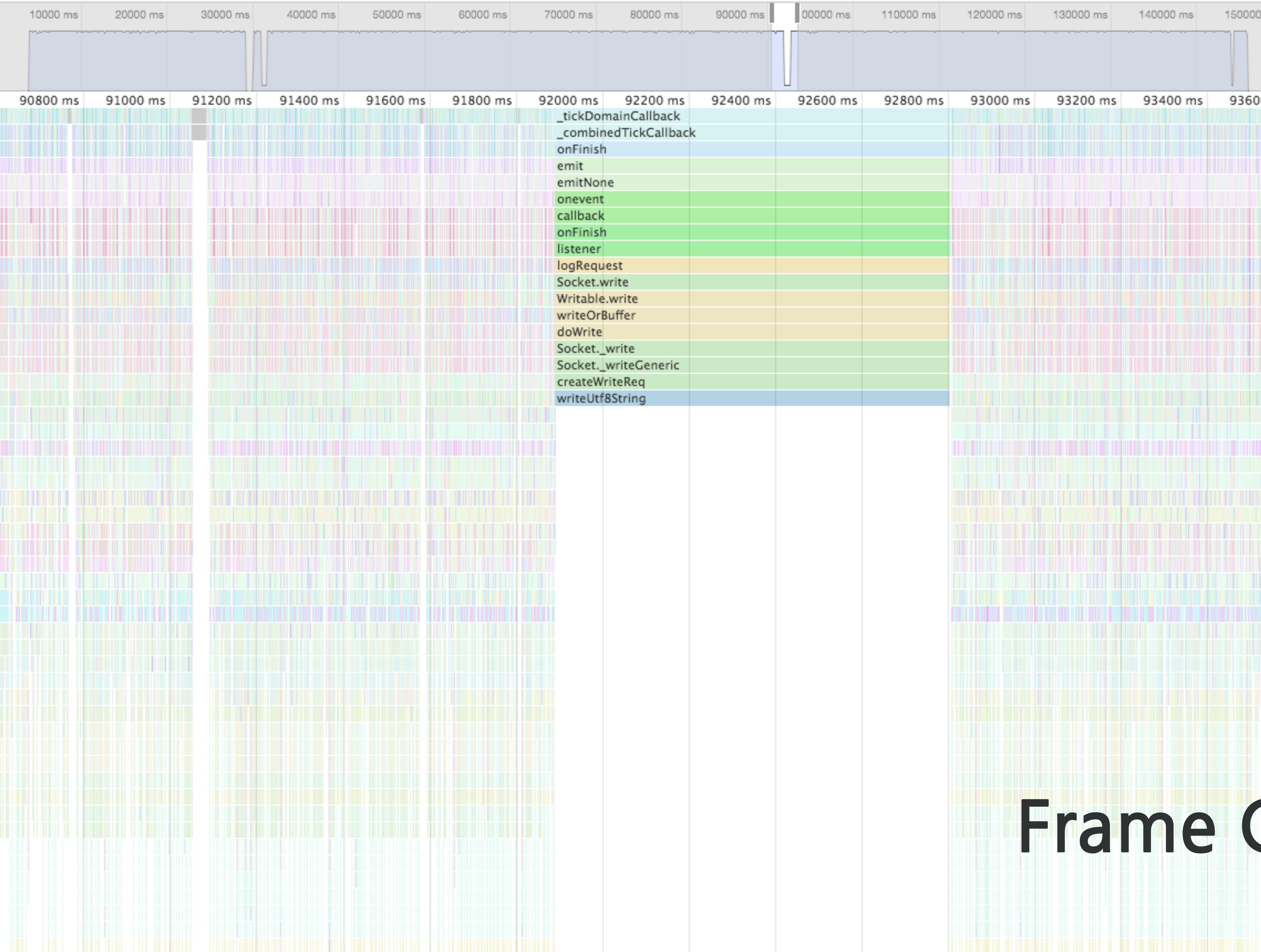
Self Time	Total Time	Function	
38307.6 ms 25.40 %	38307.6 ms 25.40 %	(program)	
6144.1 ms 4.07 %	6144.1 ms 4.07 %	► writeBuffer	
4638.4 ms 3.08 %	4638.4 ms 3.08 %	(garbage collector)	
3735.4 ms 2.48 %	3735.4 ms 2.48 %	► verify	
3301.5 ms 2.19 %	4599.0 ms 3.05 %	► (anonymous)	
3296.6 ms 2.19 %	3555.9 ms 2.36 %	► replace	
3088.6 ms 2.05 %	4166.3 ms 2.76 %	► Result.addFields	/Users/outsider...b/result.js:90
2830.8 ms 1.88 %	2830.8 ms 1.88 %	► writeUtf8String	
1814.2 ms 1.20 %	1863.3 ms 1.24 %	► split	
1659.5 ms 1.10 %	41892.8 ms 27.78 %	► emit	events.js:156
1275.2 ms 0.85 %	1275.2 ms 0.85 %	► writev	
1220.1 ms 0.81 %	1220.1 ms 0.81 %	► randomBytes	
1165.8 ms 0.77 %	12350.0 ms 8.19 %	► Socket._writeGeneric	net.js:708
1163.9 ms 0.77 %	1879.3 ms 1.25 %	► ModelBase.set	/Users/outsider...e/model.js:181
1146.1 ms 0.76 %	1743.4 ms 1.16 %	► (anonymous)	
928.0 ms 0.62 %	7188.9 ms 4.77 %	► Result.parseRow	/Users/outsider...b/result.js:69
778.7 ms 0.52 %	888.7 ms 0.59 %	► DoJoin	native array.js:95
761.6 ms 0.50 %	786.2 ms 0.52 %	► baseAssignValue	/Users/outsider...lodash.js:2571
756.0 ms 0.50 %	756.0 ms 0.50 %	► stat	
748.7 ms 0.50 %	57186.7 ms 37.92 %	► tryCatcher	/Users/outsider...ase/util.js:12
741.4 ms 0.49 %	14294.3 ms 9.48 %	► next	/Users/outsider...r/index.js:176
708.2 ms 0.47 %	1512.1 ms 1.00 %	► Buffer.write	buffer.js:850

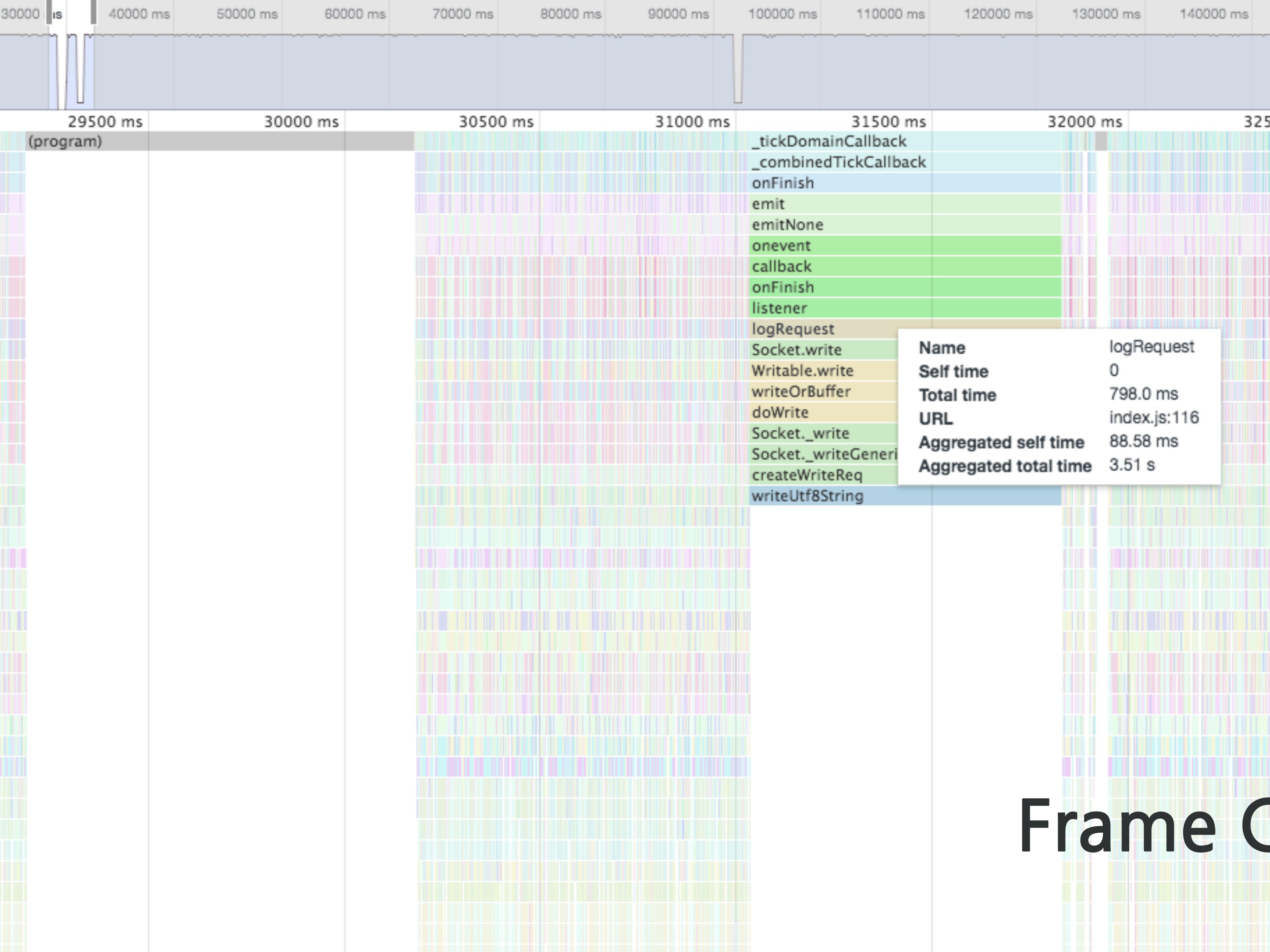
Heavy (Bottom Up)

Chart ▾



Frame Chart





Frame Chart

Sources

Content scripts

>>

:



index.js x

```
112     file:///Users/outsider/smartsstudy/          /node_modules/morgan/index.js
113     recordStartTime.call(req)
114
115     function logRequest () {
116       if (skip !== false && skip(req, res)) {
117         debug('skip request')
118         return
119       }
120
121       var line = formatLine(morgan, req, res)
122
123       if (line == null) {
124         debug('skip line')
125         return
126       }
127
128       debug('log request')
129       stream.write(line + '\n')
130     };
131
132     if (immediate) {
133       // immediate log
134       logRequest()
135     } else {
136       // record response start
137       onHeaders(res, recordStartTime)
138
139       // log when response finished
140       onFinished(res, logRequest)
141     }
142
143     next()
144   }
145 }
146 }
```

V8 Tick Profiler

```
$ node --prof ./bin/api
```

isolate-0x10380000-v8.log

V8 Tick Processor

```
$ node --prof-process isolate-0x10380000-v8.log > processed.txt
dropping: overflow
Code move event for unknown code: 0x2a2f8121cac0
Code move event for unknown code: 0x2a2f8124f9a0
Code move event for unknown code: 0x2a2f81254ae0
```

processed.txt

[Shared libraries]:

ticks	total	nonlib	name
783	0.5%		/usr/lib/system/libsystem_pthread.dylib
735	0.5%		/usr/lib/system/libsystem_platform.dylib

[JavaScript]:

ticks	total	nonlib	name
3358	2.3%	2.3%	LoadIC: A load IC from the snapshot
2979	2.0%	2.0%	Builtin: KeyedLoadIC_Megamorphic
1966	1.3%	1.3%	Builtin: CallFunction_ReceiverIsAny

[C++]:

ticks	total	nonlib	name
2253	1.5%	1.5%	T __channel_get_opt
1779	1.2%	1.2%	T __mprotect
1545	1.0%	1.1%	t v8::internal::LookupIterator::State v8::internal::LookupIterator::LookupInRegularHolder<false>(v8::internal::Map*, v8::internal::Object*, v8::internal::Object**)

[Summary]:

ticks	total	nonlib	name
60762	41.1%	41.5%	JavaScript
53997	36.5%	36.9%	C++
4560	3.1%	3.1%	GC
1589	1.1%		Shared libraries
31582	21.3%		Unaccounted

[C++ entry points]:

ticks	cpp	total	name
4305	10.3%	2.9%	T v8::internal::Runtime_StoreIC_Miss(int, v8::internal::Object**, v8::internal::Isolate*)
3155	7.5%	2.1%	T v8::internal::Builtin_FunctionConstructor(int, v8::internal::Object**, v8::internal::Isolate*)

[Bottom up (heavy) profile]:

Note: percentage shows a share of a particular caller in the total amount of its parent calls.

Callers occupying less than 1.0% are not shown.

ticks	parent	name
31582	21.3%	UNKNOWN
13500	42.7%	LazyCompile: *Socket._writeGeneric net.js:708:42
11724	86.8%	LazyCompile: *Writable.write _stream_writable.js:264:36
11662	99.5%	LazyCompile: *Socket.write net.js:699:34
5453	46.8%	Function: ~logRequest /Users/outsider/smartsstudy/node_modules/morgan/index.js:116:25

[JavaScript]:

ticks	total	nonlib	name
3358	2.3%	2.3%	LoadIC: A load IC from the snapshot
2979	2.0%	2.0%	Builtin: KeyedLoadIC_Megamorphic
1966	1.3%	1.3%	Builtin: CallFunction_ReceiverIsAny
1866	1.3%	1.3%	StoreIC: A store IC from the snapshot
1507	1.0%	1.0%	Builtin: KeyedStoreIC_Megamorphic
1489	1.0%	1.0%	Builtin: InterpreterEntryTrampoline
1341	0.9%	0.9%	Builtin: FastNewClosure
1074	0.7%	0.7%	Stub: GetPropertyStub
898	0.6%	0.6%	Stub: StringAddStub
890	0.6%	0.6%	KeyedLoadIC: A keyed load IC from the snapshot
717	0.5%	0.5%	Builtin: ArgumentsAdaptorTrampoline
706	0.5%	0.5%	LazyCompile: *Promise._settlePromises /Users/outsider/smartsstudy/node_modules/bluebird/js/r
560	0.4%	0.4%	LazyCompile: *emit events.js:156:44
516	0.3%	0.4%	Builtin: CompileLazy
502	0.3%	0.3%	Builtin: FunctionPrototypeHasInstance
492	0.3%	0.3%	Builtin: ObjectHasOwnProperty
486	0.3%	0.3%	StoreIC: A store IC from the snapshot {1}
472	0.3%	0.3%	RegExp: (.*?)(\\[[0-9]\\])
450	0.3%	0.3%	LazyCompile: *<anonymous> :1:10
442	0.3%	0.3%	Builtin: JConstructStubGenericUnrestrictedReturn
425	0.3%	0.3%	Builtin: RegExpPrototypeExec
398	0.3%	0.3%	Builtin: KeyedStoreIC_Megamorphic_Strict
395	0.3%	0.3%	Builtin: FastCloneRegExp
389	0.3%	0.3%	LazyCompile: *Promise._then /Users/outsider/smartsstudy/node_modules/bluebird/js/r
377	0.3%	0.3%	Builtin: FastArrayPush
372	0.3%	0.3%	LazyCompile: *<anonymous> /Users/outsider/smartsstudy/node_modules/pg/lib/connection
352	0.2%	0.2%	Builtin: RegExpPrototypeTest
347	0.2%	0.2%	Builtin: StrictEqual
345	0.2%	0.2%	Builtin: StringEqual

[Bottom up (heavy) profile]:

Note: percentage shows a share of a particular caller in the total amount of its parent calls.
Callers occupying less than 1.0% are not shown.

ticks	parent	name
31582	21.3%	UNKNOWN
13500	42.7%	LazyCompile: *Socket._writeGeneric net.js:708:42
11724	86.8%	LazyCompile: *Writable.write _stream_writable.js:264:36
11662	99.5%	LazyCompile: *Socket.write net.js:699:34
5453	46.8%	Function: ~logRequest /Users/outsider/smartstudy/node_modules/morgan/index.js:116:25
5296	97.1%	Function: ~listener /Users/outsider/smartstudy/node_modules/on-finished/index.js:16
157	2.9%	LazyCompile: *listener /Users/outsider/smartstudy/node_modules/on-finished/index.js
3419	29.3%	LazyCompile: *Client._pulseQueryQueue /Users/outsider/smartstudy/node_modules/pg/lib/
3405	99.6%	LazyCompile: *<anonymous> /Users/outsider/smartstudy/node_modules/knex/lib/dialects
2744	23.5%	LazyCompile: *<anonymous> /Users/outsider/smartstudy/node_modules/pg/lib/client.js:13
2363	86.1%	LazyCompile: *emit events.js:156:44
380	13.8%	LazyCompile: *<anonymous> /Users/outsider/smartstudy/node_modules/bluebird/js/relea
925	6.9%	LazyCompile: *clearBuffer _stream_writable.js:469:21
839	90.7%	LazyCompile: *end _http_outgoing.js:726:45
839	100.0%	LazyCompile: *send /Users/outsider/smartstudy/node_modules/express/lib/response.js:10
734	87.5%	LazyCompile: *json /Users/outsider/smartstudy/node_modules/express/lib/response.js:
105	12.5%	Function: ~json /Users/outsider/smartstudy/node_modules/express/lib/response.js:229
56	6.1%	Function: ~Writable.uncork _stream_writable.js:302:37
56	100.0%	Function: ~end _http_outgoing.js:726:45
51	91.1%	Function: ~send /Users/outsider/smartstudy/node_modules/express/lib/response.js:106
5	8.9%	LazyCompile: *send /Users/outsider/smartstudy/node_modules/express/lib/response.js:
30	3.2%	LazyCompile: *<anonymous> /Users/outsider/smartstudy/node_modules/knex/lib/dialects/pos
30	100.0%	Function: ~end _http_outgoing.js:726:45
19	63.3%	LazyCompile: *send /Users/outsider/smartstudy/node_modules/express/lib/response.js:
11	36.7%	Function: ~send /Users/outsider/smartstudy/node_modules/express/lib/response.js:106
805	6.0%	Function: ~Socket.write net.js:785:35

3358 2.3% LoadIC: A load IC from the snapshot
333 9.9% LazyCompile: *emit events.js:156:44
43 12.9% LazyCompile: *<anonymous> /Users/outsider/smartstudy/node_modules/knex/lib/runner.js:1
43 100.0% LazyCompile: *tryCatcher /Users/outsider/smartstudy/node_modules/bluebird/js/release
39 90.7% LazyCompile: *<anonymous> /Users/outsider/smartstudy/node_modules/bluebird/js/release
39 100.0% Function: ~<anonymous> /Users/outsider/smartstudy/node_modules/knex/lib/runner.js:1
3 7.0% LazyCompile: *<anonymous> /Users/outsider/smartstudy/node_modules/knex/lib/runner.js:1
3 100.0% LazyCompile: *tryCatcher /Users/outsider/smartstudy/node_modules/bluebird/js/release
1 2.3% LazyCompile: *tryCatcher /Users/outsider/smartstudy/node_modules/bluebird/js/release
1 100.0% Function: ~<anonymous> /Users/outsider/smartstudy/node_modules/bluebird/js/release
30 9.0% Function: ~<anonymous> /Users/outsider/smartstudy/node_modules/knex/lib/runner.js:51:7
30 100.0% LazyCompile: *tryCatcher /Users/outsider/smartstudy/node_modules/bluebird/js/release
24 80.0% Function: ~<anonymous> /Users/outsider/smartstudy/node_modules/bluebird/js/release
24 100.0% LazyCompile: *tryCatcher /Users/outsider/smartstudy/node_modules/bluebird/js/release
6 20.0% LazyCompile: *<anonymous> /Users/outsider/smartstudy/node_modules/bluebird/js/release
6 100.0% LazyCompile: *tryCatcher /Users/outsider/smartstudy/node_modules/bluebird/js/release
29 8.7% LazyCompile: *<anonymous> /Users/outsider/smartstudy/node_modules/pg/lib/connection.js:1
29 100.0% LazyCompile: *emit events.js:156:44
29 100.0% LazyCompile: *addChunk _stream_readable.js:261:18
29 100.0% LazyCompile: *onread net.js:576:16
22 6.6% LazyCompile: *<anonymous> /Users/outsider/smartstudy/node_modules/knex/lib/util/make-kebab-case.js:1
22 100.0% LazyCompile: *emit events.js:156:44
22 100.0% LazyCompile: *<anonymous> /Users/outsider/smartstudy/node_modules/knex/lib/runner.js:1
22 100.0% LazyCompile: *tryCatcher /Users/outsider/smartstudy/node_modules/bluebird/js/release
21 6.3% Function: ~<anonymous> /Users/outsider/smartstudy/node_modules/knex/lib/runner.js:152:1
21 100.0% LazyCompile: *tryCatcher /Users/outsider/smartstudy/node_modules/bluebird/js/release
20 95.2% LazyCompile: *Promise._settlePromises /Users/outsider/smartstudy/node_modules/bluebird/js/release
20 100.0% LazyCompile: *processImmediateTimers timers.js:697:26
1 4.8% LazyCompile: *Promise._settlePromiseFromHandler /Users/outsider/smartstudy/node_modules/bluebird/js/release
1 100.0% LazyCompile: *Async.drainQueue /Users/outsider/smartstudy/node_modules/bluebird/js/release
496:56

mogan 제거

```
const morgan = require('morgan');
app.use(morgan('dev'));
```

Auto cannon 으로 비교

```
$ npx autocannon -c 100 -H "Authorization: token" \
http://localhost:3000/api
```

Stat	Avg	Stdve	Max
Latency (ms)	248.716	44.386	437.6
Req/Sec	398.72	40.022	472.4

Stat	Avg	Stdve	Max
Latency (ms)	235.728	38.748	404.4
Req/Sec	420.92	36.002	477.4

node-postgres 업데이트

6.4.2 -> 7.4.0

Auto cannon 으로 비교

```
$ npx autocannon -c 100 -H "Authorization: token" \
http://localhost:3000/api
```

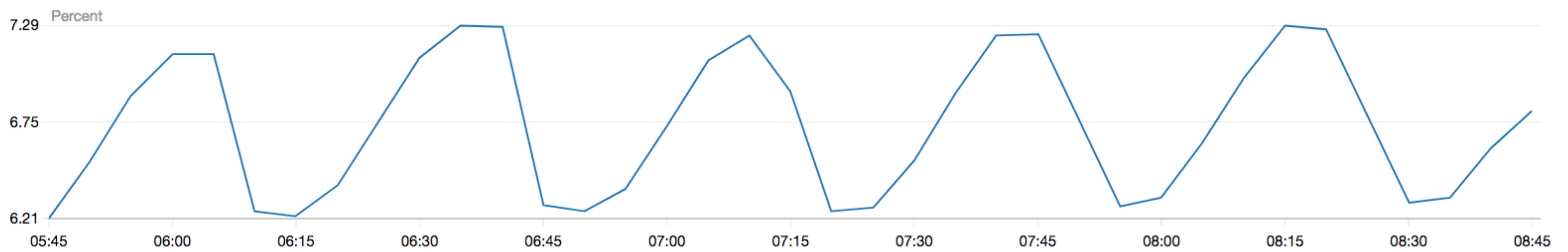
Stat	Avg	Stdve	Max
Latency (ms)	235.728	38.748	404.4
Req/Sec	420.92	36.002	477.4

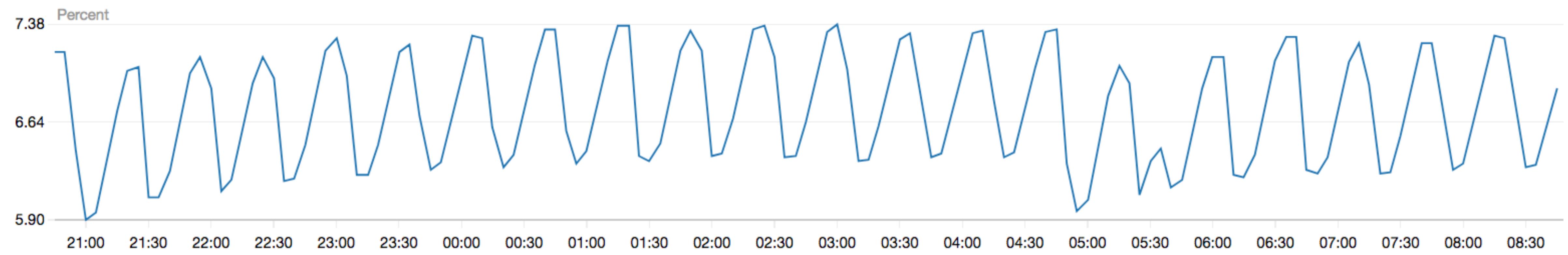
Stat	Avg	Stdve	Max
Latency (ms)	200.933	36.118	358.6
Req/Sec	495.78	45.058	594.8

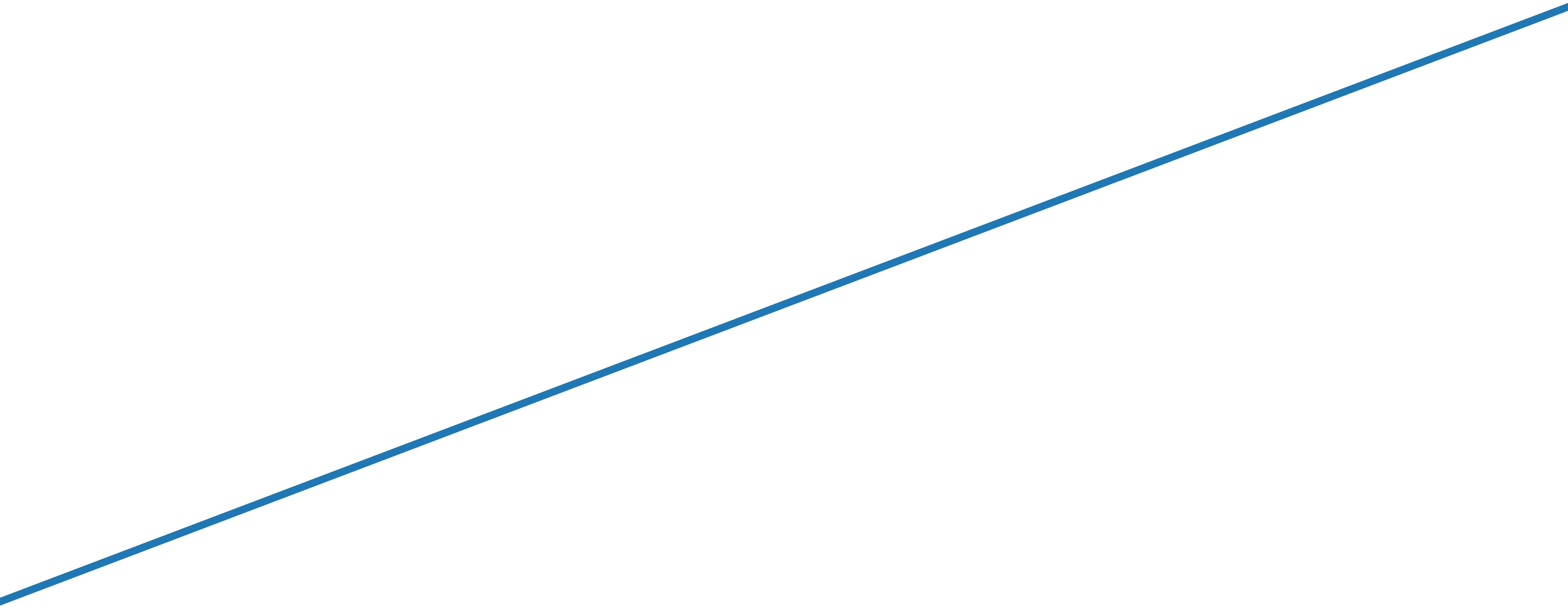
Heap Dump

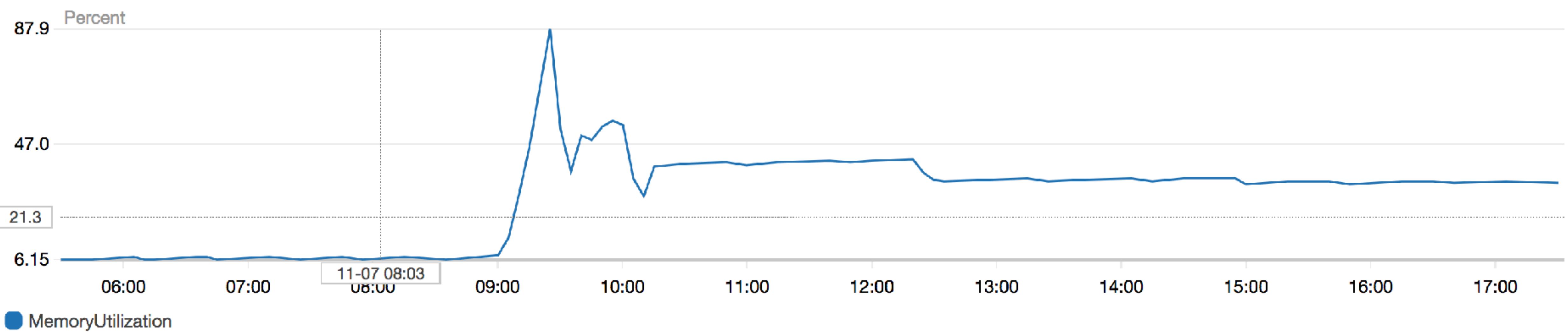
메모리 누수는 추적이 어렵다.

일주일마다 서버 재시작!!











Profiles

Select profiling type

Take heap snapshot

Heap snapshot profiles show memory distribution among your page's JavaScript objects and related DOM nodes.

Record allocation profile

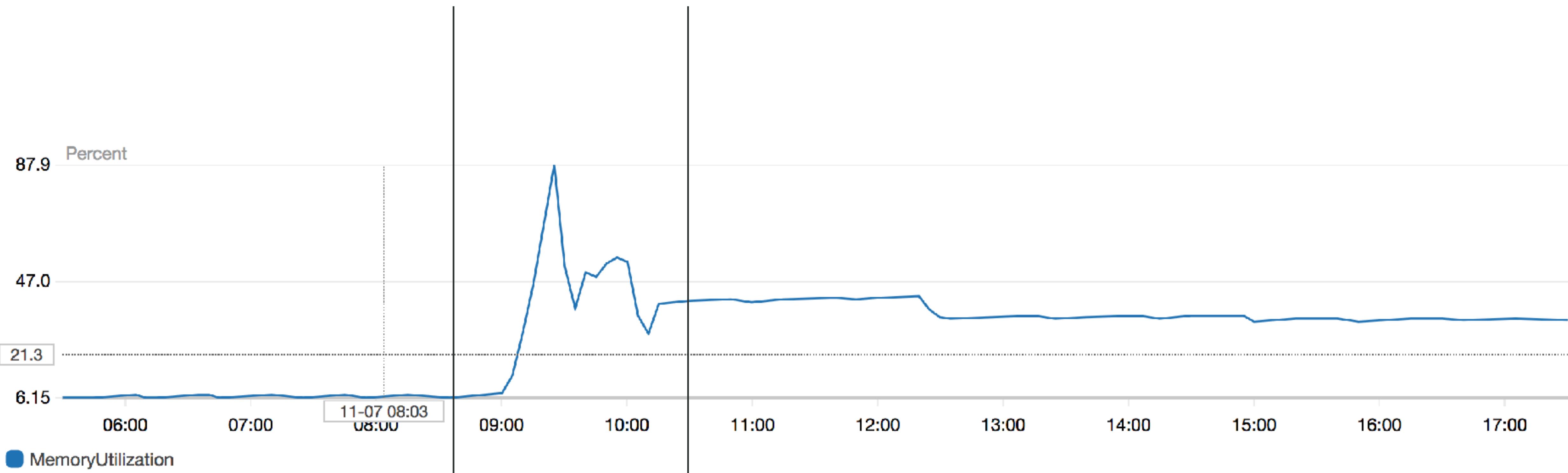
Allocation profiles show memory allocations from your JavaScript functions.

Record allocation timeline

Allocation timelines show memory allocations from your heap over time. Use this profile type to isolate memory leaks.

Take snapshot

Load



Heap dump

Heap dump

[Connection](#)[Console](#)[Sources](#)[Memory](#)[Profiler](#)

Profiles

HEAP SNAPSHOTS



before-loadtest
44.6 MB



after-loadtest
170 MB

Select profiling type



Take heap snapshot

Heap snapshot profiles show memory distribution among your page's .



Record allocation profile

Allocation profiles show memory allocations from your JavaScript functi



Record allocation timeline

Allocation timelines show memory allocations from your heap over time

[Take snapshot](#)[Load](#)



Profiles	HEAP SNAPSHTOS	before-loadtest 44.6 MB	after-loadtest 170 MB	Summary	Class filter	All objects					
				Constructor	Distance	Objects Count	Shallow Size	Retained Size			
				▶ global /	1	1 0 %	40 0 %	44 100 942 94 %			
				▶ (compiled code)	3	20 216 5 %	4 356 184 9 %	24 471 888 52 %			
				▶ (array)	2	78 611 20 %	13 106 720 28 %	16 601 512 36 %			
				▶ (string)	2	68 840 17 %	16 112 480 34 %	16 112 480 34 %			
				▶ (closure)	2	23 582 6 %	1 690 616 4 %	13 743 884 29 %			
				▶ (system)	-	125 067 31 %	5 602 960 12 %	11 397 592 24 %			
				▶ Object	2	10 700 3 %	597 368 1 %	9 921 234 21 %			
				▶ system / Context	3	6 480 2 %	475 480 1 %	9 470 448 20 %			
				▶ Array	3	26 724 7 %	855 168 2 %	4 093 416 9 %			
				▶ (concatenated string)	4	20 791 5 %	831 640 2 %	1 982 824 4 %			
				▶ Module	3	1 760 0 %	140 688 0 %	1 919 192 4 %			
				▶ (regexp)	6	903 0 %	50 568 0 %	420 800 1 %			
				▶ Layer	9	67 0 %	5 360 0 %	276 176 1 %			
				▶ Mime	8	2 0 %	208 0 %	225 288 0 %			
				▶ EventEmitter	3	51 0 %	2 880 0 %	212 080 0 %			
				▶ Route	11	20 0 %	1 856 0 %	209 128 0 %			
				▶ Map	3	18 0 %	584 0 %	177 176 0 %			
				▶ Service	9	106 0 %	11 024 0 %	168 800 0 %			
				▶ AST_Node	10	85 0 %	9 160 0 %	116 176 0 %			
				▶ Pool	9	1 0 %	160 0 %	106 836 0 %			
				▶ Operation	16	60 0 %	1 480 0 %	102 736 0 %			
				▶ NativeModule	7	87 0 %	5 568 0 %	83 208 0 %			
				▶ Type	8	16 0 %	1 536 0 %	73 760 0 %			



Profiles	Constructor	Distance	Objects Count		Shallow Size		Retained Size	
			Count	Percentage	Size	Percentage	Size	Percentage
HEAP SNAPSHOTs	▶ global /	1	1	0 %	40	0 %	44 100	94 %
	▶ (compiled code)	3	20 216	5 %	4 356	184	24 471	52 %
	▶ (array)	2	78 611	20 %	13 106	720	16 601	36 %
	▶ (string)	2	68 840	17 %	16 112	480	16 112	34 %
	▶ (closure)	2	23 582	6 %	1 690	616	13 743	29 %
	▶ (system)	-	125 067	31 %	5 602	960	11 397	24 %
	▶ Object	2	10 700	3 %	597	368	9 921	21 %
	▶ system / Context	3	6 480	2 %	475	480	9 470	20 %
	▶ Array	3	26 724	7 %	855	168	4 093	9 %
	▶ concatenated string	4	20 791	5 %	831	640	1 982	4 %
	▶ Module	3	1 750	0 %	1 189	0	1 590	192
	▶ RegExp	1	9 033	0 %	68	0	420	300
	▶ Layer	9	67	0 %	5 360	0 %	276	176
	▶ Mime	8	2	0 %	208	0 %	225	288
	▶ EventEmitter	3	51	0 %	2 880	0 %	212	080
	▶ Route	11	20	0 %	1 856	0 %	209	128
	▶ Map	3	18	0 %	584	0 %	177	176
	▶ Service	9	106	0 %	11 024	0 %	168	800
	▶ AST_Node	10	85	0 %	9 160	0 %	116	176
	▶ Pool	9	1	0 %	160	0 %	106	836
	▶ Operation	16	60	0 %	1 480	0 %	102	736
	▶ NativeModule	7	87	0 %	5 568	0 %	83	208
	▶ Type	8	16	0 %	1 536	0 %	73	760

단위는 byte

Heap dump의 사이즈 = 메모리 용량

Shallow Size = 객체의 실제 크기

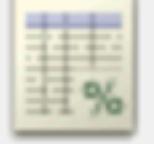
Retained Size = 객체의 레퍼런스가 있으면 이를 포함한 크기



● ✖ trash

Comparison ▼ Class filter before-loadtest ▼

Profiles	Constructor	# New	# Deleted	# Delta	Alloc. Size	Freed Size	Size Delta
		260 978	71 656	+189 322	42 630 360	11 839 904	+30 790 456
HEAP SNAPSHOTs	►(array)	216 531	63 304	+153 227	27 554 968	14 886 312	+12 668 656
before-loadtest 44.6 MB	►(string)	436 706	104 454	+332 252	20 943 264	4 682 448	+16 260 816
after-loadtest 170 MB	►(closure)	252 022	20 349	+231 673	17 486 720	1 458 680	+16 028 040
	►system / Context	186 540	5 974	+180 566	13 294 576	439 448	+12 855 128
	►Object	160 619	9 820	+150 799	8 595 144	548 768	+8 046 376
	►Date	57 942	1	+57 941	5 562 432	96	+5 562 336
	►(compiled code)	18 885	17 747	+1 138	5 301 120	3 590 816	+1 710 304
	►Promise	80 124	14	+80 110	5 128 336	1 296	+5 127 040
	►Array	150 830	24 470	+126 360	4 826 560	783 040	+4 043 520
	►(concatenated string)	111 525	19 907	+91 618	4 461 000	796 280	+3 664 720
	►Field	45 226	0	+45 226	3 618 080	0	+3 618 080
	►TraceSegment	23 382	0	+23 382	3 367 008	0	+3 367 008
	►Timer	23 563	10	+23 553	1 696 136	320	+1 695 816
	►(number)	87 316	110	+87 206	1 397 056	1 760	+1 395 296
	►Child	7 685	13	+7 672	717 304	728	+716 576
	►HTTPPARSER	407	3	+404	713 064	5 256	+707 808
	►Builder	5 334	0	+5 334	597 408	0	+597 408
	►Result	5 333	0	+5 333	511 968	0	+511 968

	Comparison ▾	Class filter	before-loadtest ▾							
Profiles	Constructor	# New	# Deleted	# Delta	Alloc. Size	Freed Size	Size Delta			
HEAP SNAPSHOTS  before-loadtest 44.6 MB	▼JWT	1	1	0	144	144	0			
	▶JWT @4342121	.	.	.	144	144	0			
	▶JWT @511657	.	.	.	144	144	0			
	▶Reservoir	3	3	0	144	144	0			
	▶Set	4	4	0	144	144	0			
	▶goog.proto2.Descriptor	1	1	0	128	128	0			
	▶FreeList	1	1	0	120	120	0			
	▶HTTPSTransport	1	1	0	120	120	0			
after-loadtest  after-loadtest 170 MB	▶_DluralDecoder	1	1	0	120	120	0			
	Retainers									
	Object			Distance	Shallow Size	Retained Size				
	▼jwtClient in system / Context @3987897			12	112	0 %	4 872	0 %		
	▼context in () @4268483			11	72	0 %	72	0 %		
	▼fn in system / Context @4267991			10	56	0 %	128	0 %		
	▼context in () @4267989			9	72	0 %	200	0 %		
	▼checkSubscriptionReceipt in @3671819			8	40	0 %	14 872	0 %		
before-loadtest 170 MB	▼exports in Module @3671765			7	80	0 %	888	0 %		
	▼/www/service/src/core/lib/google-store.js in @2139041			6	24	0 %	99 720	0 %		
	▼_cache in Module() @1243723			5	72	0 %	568 304	0 %		
	▼constructor in @640225			4	56	0 %	480	0 %		
	▼__proto__ in Module @2139037			3	80	0 %	320	0 %		
	▼mainModule in process @7503			2	24	0 %	28 056	0 %		
	▶process in global / @1279			1	40	0 %	175 529 450	98 %		
	▶73 in @3429243			3	7 936	0 %	17 472	0 %		
before-loadtest 170 MB	▶value in system / PropertyCell @377693			3	40	0 %	848	0 %		

Thank you