Pocetna ideja optimizacije performansi

Prilikom pisanja simulatora odlučili smo se da koristimo jednu aplikaciju, gde će jedan uređaji činiti jedna ili više niti u zavisnosti od problema. Naveo bi nekoliko razloga za to:

1. Efikasno korišćenje resursa: Više niti omogućava efikasnije korišćenje resursa na samom računaru ili serveru. Umesto da se pokrećete više instanci iste aplikacije, koje bi možda svaka zauzela određeni deo memorije i drugih resursa, više niti unutar jedne aplikacije dele isti proces i resurse.
2. Bolja podrška za paralelizaciju: Više niti omogućava efikasnije korišćenje više procesorskih jezgara i paralelizaciju zadatka.
3. Promena niti: Kada se jedna nit uspava ili blokira pozivom Thread.sleep(), ostale niti unutar istog procesa nastavljaju izvršavanje. Procesor može jednostavno prebaciti izvršavanje na drugu nit koja nije u stanju mirovanja, a da ne mora prelaziti na drugi proces. Ovo je efikasnije jer procesor ne mora preći na potpuno drugi kontekst izvršavanja. Kada bi umesto niti imali više procesa koji se izvršavaju, promena procesa bi bila složenija operacija. Procesor bi morao da sačuva stanje trenutnog procesa, uključujući registre, memoriju i sve ostale resurse koji su uključeni u izvršavanje, pre nego što pređe na izvršavanje drugog procesa. Ovo je znatno skuplja operacija u odnosu na jednostavnu promenu niti.

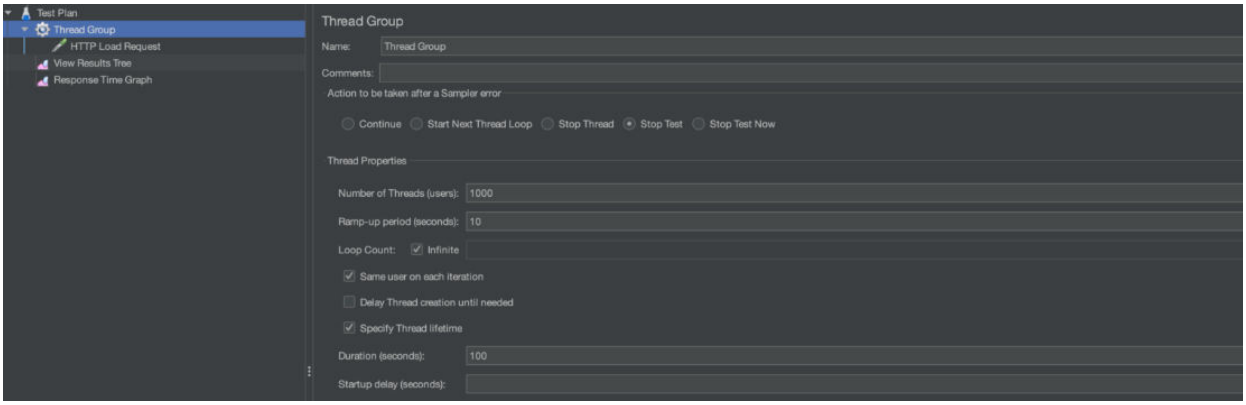
Upotreba virtuelnih niti:

Radi demonstarcije razloga korišćenja virtuelnih niti koristimo JMeter alat. Ovaj test predstavlja predstavlja polaznu tačku koju smo uzeli u razmatranje prilikom pocetnog pisanja koda. Ovu odluka je donešena još u početnim fazama razvoja projekta.

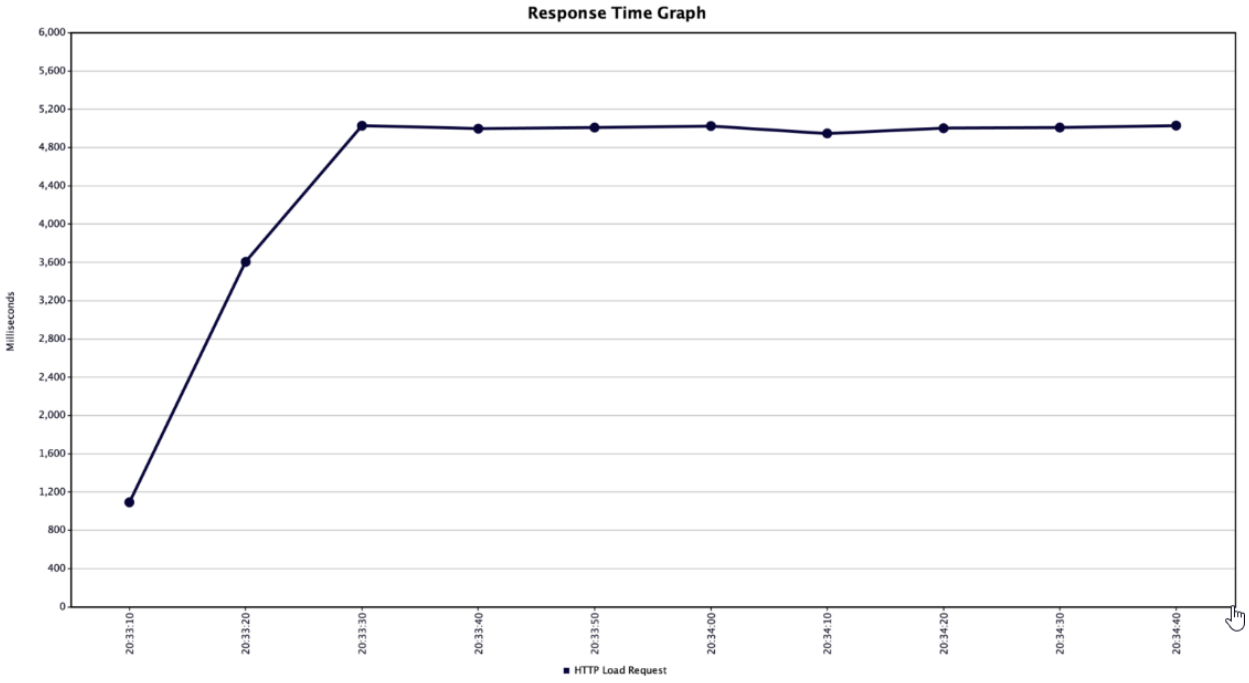
```
@RestController
@RequestMapping("/load")
public class LoadTestController {

    private static final Logger LOG = LoggerFactory.getLogger(LoadTestController.class);

    @GetMapping
    public void doSomething() throws InterruptedException {
        LOG.info("hey, I'm doing something");
        Thread.sleep(1000);
    }
}
```

U ovom specifičnom scenariju, pozvaćemo endpoint u Rest kontroleru koji će jednostavno usporiti izvršenje na jednu sekundu, simulirajući kompleksni asinhroni zadatak ili rad našeg uređaja.

JMeter test će sadržati samo jednu grupu niti, simulirajući 1000 istovremenih korisnika koji pristupaju endpointu /load tokom 100 sekundi.

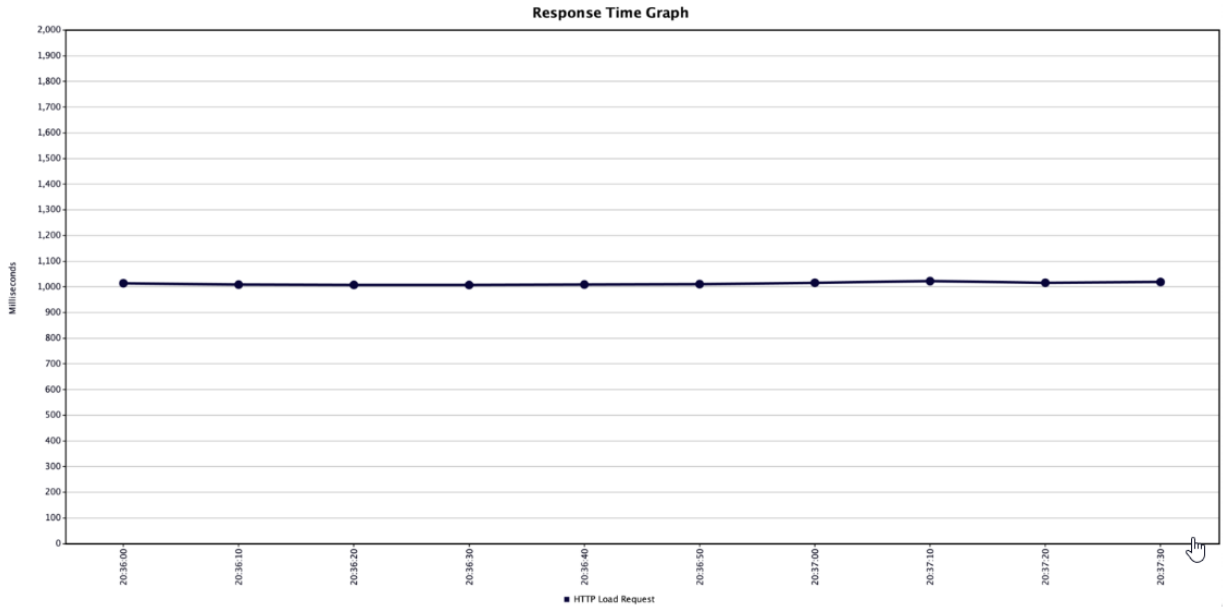


Grafikon odgovora za standardne niti jasno pokazuje vreme potrebno za završetak poziva, koje veoma brzo dostiže 5000 milisekundi.



Ovo se dešava jer su sistemske niti ograničen resurs, i kada su sve niti zauzete, ne preostaje ništa za Spring aplikaciju osim da zaustavi zahtev dok jedna nit ne postane slobodna. Isto bi bilo I u slučaju rada simulatora (kada se odradi sleep), lakše je procesoru da izvrši promenu virtuelne niti.

Korišćenje virtuelnih niti:



Grafikon virtuelnih niti pokazuje da se odgovor stabilizuje na 1000 milisekundi. Virtuelne niti se stvaraju i koriste odmah nakon zahteva jer su veoma jeftine sa aspekta resursa.

Testiranje (Ambijentalni Senzor)

Ponasanje uređaja u slucaju kada imamo vise aktivnih uređaja u sistemu.

Uređaji šalje odgovor svakih 5 sekundi

NAPOMENA: Zadnja red je rađen pre optimizacije (rezultati su sada dosta bolji)

Br. Uređaja	1	10	100	1000	10.000
Vreme odgovora paljenje (ms)	30 - 50	30 - 50	30 - 50	100 - 200	120 - 220
Vreme odgovora gašenje (ms)	30 - 50	30 - 50	30 - 50	100 - 200	120 - 230
Vreme odgovora istorija za 1 sat (ms)	120 - 150	120 - 150	120 - 150	150 - 180	150 - 180

Sistem radi, ali mqtt ne moze da postigne odradi publish kada je 10.000 uređaja aktivno i svaki salje odgovor na 5s.

```
Exception in thread "" Too many publishes in progress (32202)
    at org.eclipse.paho.mqttv5.client.internal.ClientState.send(ClientState.java:544)
    at org.eclipse.paho.mqttv5.client.internal.ClientComms.internalSend(ClientComms.java:155)
    at org.eclipse.paho.mqttv5.client.internal.ClientComms.sendNoWait(ClientComms.java:218)
    at org.eclipse.paho.mqttv5.client.MqttAsyncClient.publish(MqttAsyncClient.java:1530)
    at org.eclipse.paho.mqttv5.client.MqttClient.publish(MqttClient.java:564)
    at rs.ac.uns.ftn.nwt.simulator_server.service.AmbientSensorService.generateMeasurements(AmbientSensorService.java:107)
    at rs.ac.uns.ftn.nwt.simulator_server.service.AmbientSensorService.lambda$startSimulate$0(AmbientSensorService.java:71) <1 internal line>
    at java.base/java.lang.VirtualThread.run(VirtualThread.java:309)
Exception in thread "" Too many publishes in progress (32202)
    at org.eclipse.paho.mqttv5.client.internal.ClientState.send(ClientState.java:544)
    at org.eclipse.paho.mqttv5.client.internal.ClientComms.internalSend(ClientComms.java:155)
    at org.eclipse.paho.mqttv5.client.internal.ClientComms.sendNoWait(ClientComms.java:218)
    at org.eclipse.paho.mqttv5.client.MqttAsyncClient.publish(MqttAsyncClient.java:1530)
    at org.eclipse.paho.mqttv5.client.MqttClient.publish(MqttClient.java:564)
    at rs.ac.uns.ftn.nwt.simulator_server.service.AmbientSensorService.generateMeasurements(AmbientSensorService.java:107)
    at rs.ac.uns.ftn.nwt.simulator_server.service.AmbientSensorService.lambda$startSimulate$0(AmbientSensorService.java:71) <1 internal line>
    at java.base/java.lang.VirtualThread.run(VirtualThread.java:309)
```

Vreme publish-a se povecalo na 30 sekundi. Jedno od mogucih resenja je imati vise mqtt borkera.

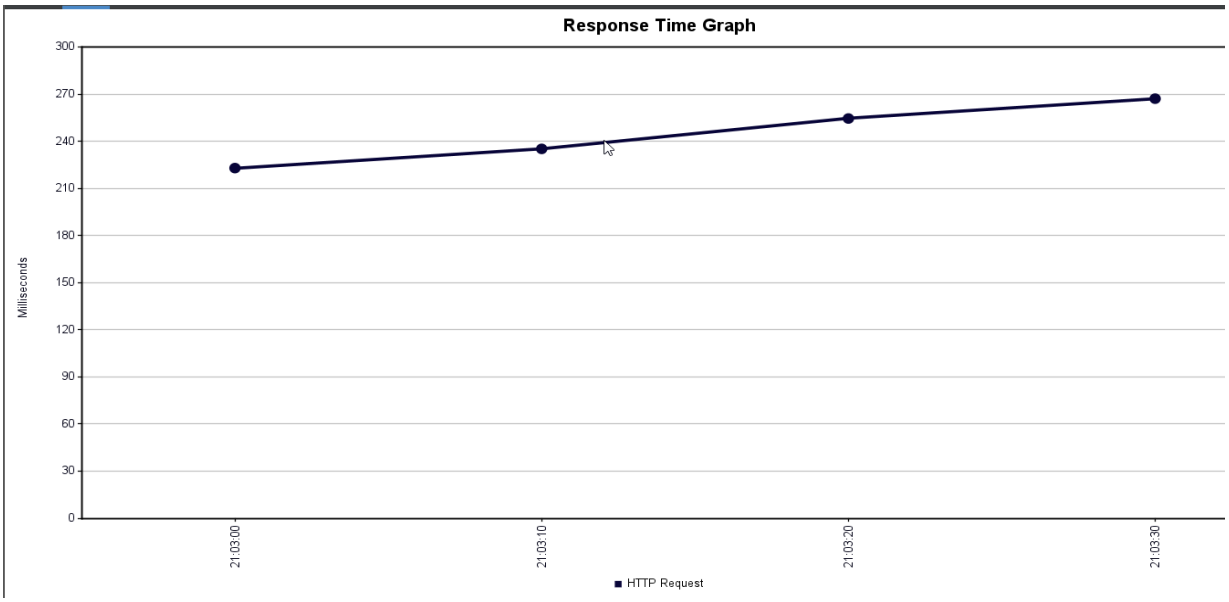

```
{
    "timestamp": "2024-01-31T00:30:11.992",
    "temperature": -0.8811414866029638,
    "humidity": 79.32885851339704
},
{
    "timestamp": "2024-01-31T00:30:56.992",
    "temperature": -0.7896508909327447,
    "humidity": 79.42034910906726
}
```

1. Dobavljanje istorije (za poslednji mesec)

Testni slučaji: Za 5 uređaja u bazi šalje se nasumičan zahtev za dobavljanje istorije u poslednjih mesec dana.

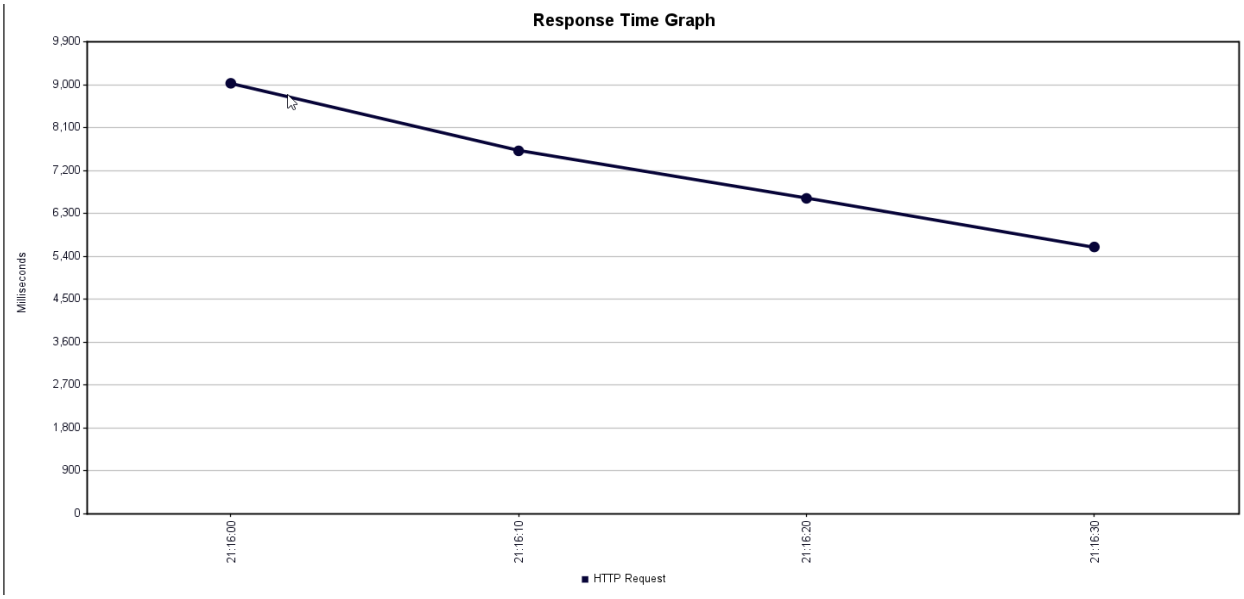
1 korisnik

Label	# Samples	Average ↑	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	181	248	0	426	33.79	0.00%	4.0/sec	1284.48	1.08	327110.8
TOTAL	181	248	0	426	33.79	0.00%	4.0/sec	1284.48	1.08	327110.8



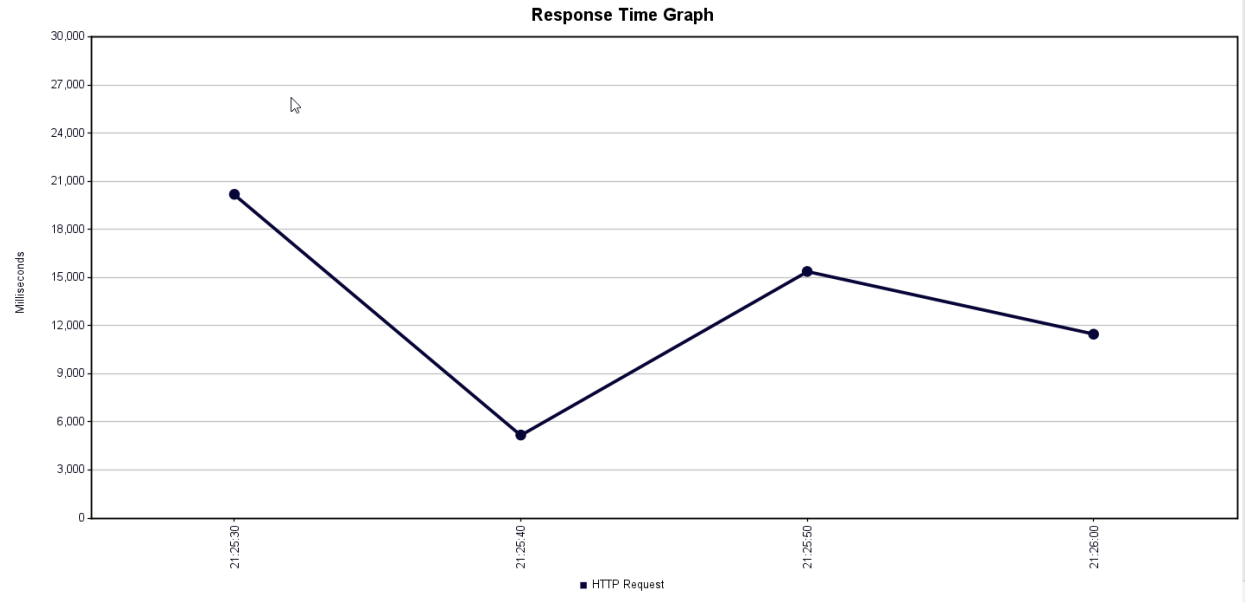
50 korisnika

Label	# Samples	Average ↑	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	349	6568	0	18644	4369.26	1.15%	7.2/sec	2277.01	1.93	323292.1
TOTAL	349	6568	0	18644	4369.26	1.15%	7.2/sec	2277.01	1.93	323292.1



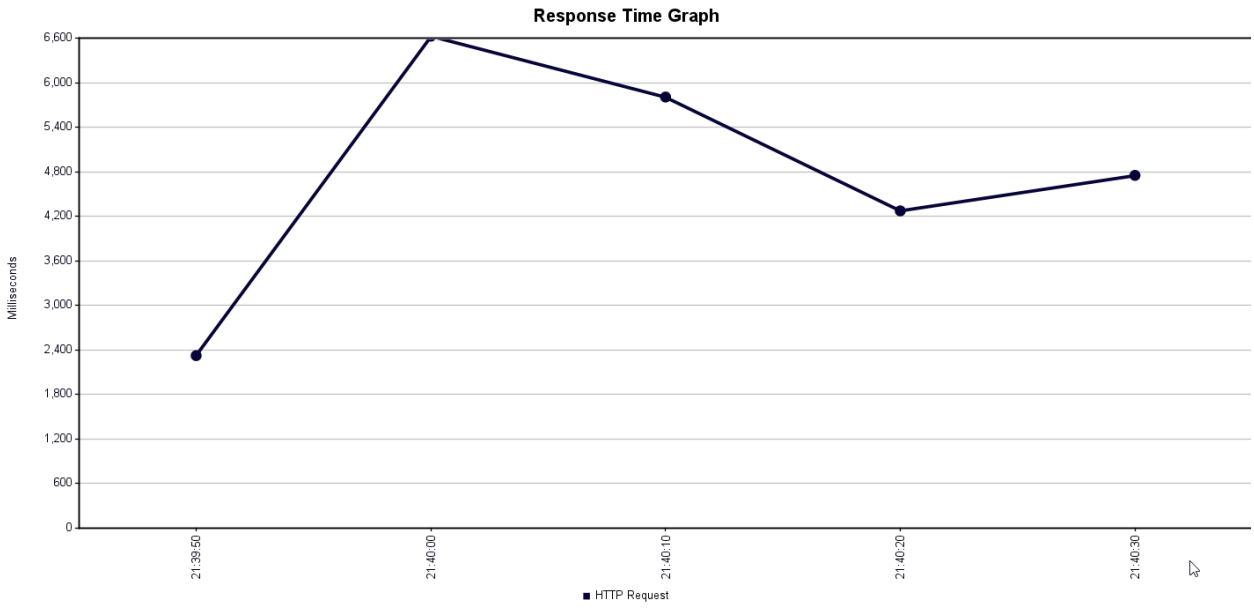
100 korisnika

Label	# Samples	Average ↑	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	334	13531	0	46781	11951.10	0.00%	7.0/sec	2229.57	1.87	326959.0
TOTAL	334	13531	0	46781	11951.10	0.00%	7.0/sec	2229.57	1.87	326959.0



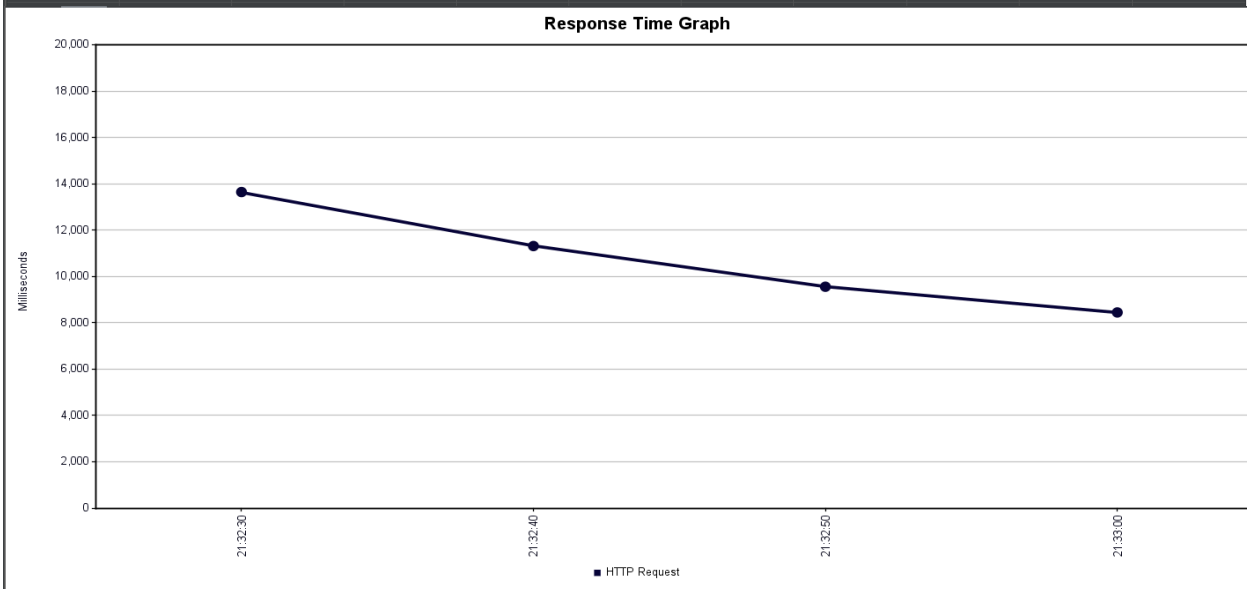
50 korisnika optimizacija

Label	# Samples	Average ↑	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	444	5214	0	31512	4091.25	0.00%	8.9/sec	2849.20	2.39	326808.0
TOTAL	444	5214	0	31512	4091.25	0.00%	8.9/sec	2849.20	2.39	326808.0



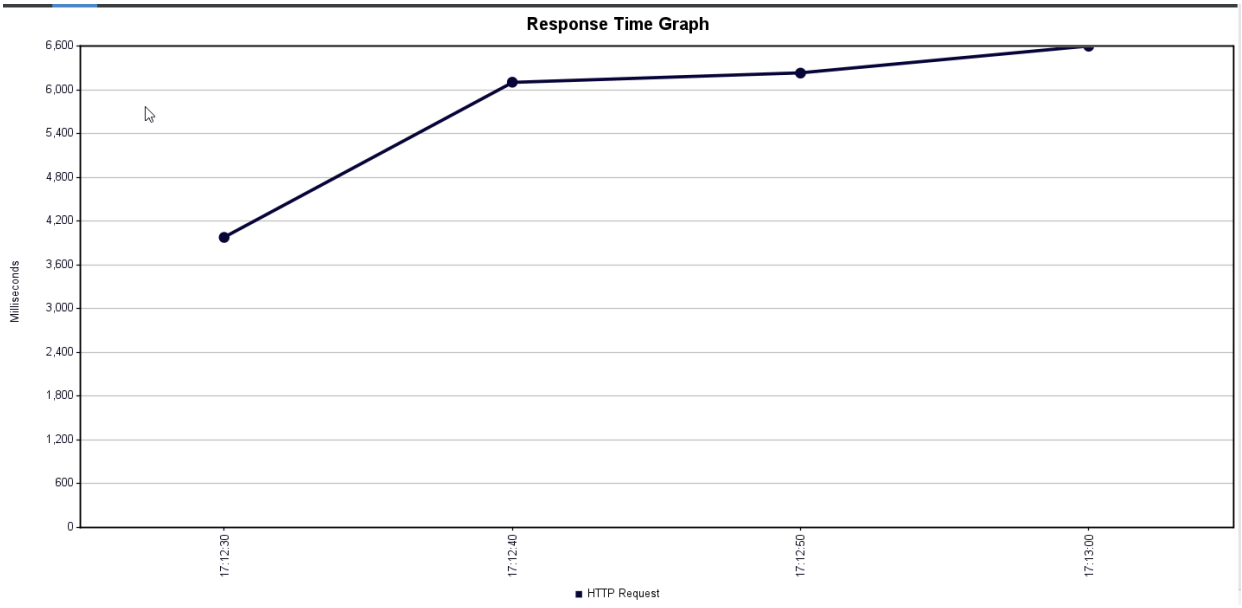
100 korisnika optimizacija

Label	# Samples	Average ↑	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	413	11574	0	44822	5297.42	1.45%	7.6/sec	2387.37	2.03	322139.6
TOTAL	413	11574	0	44822	5297.42	1.45%	7.6/sec	2387.37	2.03	322139.6



100 korisnika optimizacija, optimizacija

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	832	5718	0	13382	2490.43	0.00%	15.8/sec	5111.86	4.23	331450.0
TOTAL	832	5718	0	13382	2490.43	0.00%	15.8/sec	5111.86	4.23	331450.0



Br. korisnika	1	50	100
Propusnost(sec)	4	7.2	7.0
Propusnost (sec) optimizacija		8.9	7.6
Propusnost dodatna optimizacija (sec)		15.0	15.8
Greska (%)	0.0	0.0	0.0

Optimizacija:

Vrišimo prolaz kroz listu temperatura i vlažnosti vazduha radi njihovog mapiranja na DTO. Pre optimizacije prolazili smo prvo kroz jednu listu, pa zatim kroz drugu. Sada se jednim prolazom ovo mapira.

```
return IntStream.range(0, Math.max(temperatures.size(), humidityList.size())
 ) IntStream
        .mapToObj(i -> castToWebResponse(
                temperatures.get(i).getValue(),
                humidityList.get(i).getValue(),
                humidityList.get(i).getTimestamp())
                 ) Stream<AmbientSensorHistoryWebRespons...>
        .toList();
```

Dodatna optimizacije:

Popravljen je flux upit gde se jednim upitom dobavlja I temperatura I vlaznost vazduha za datu uređaji. Performanse su se povećale za duplo

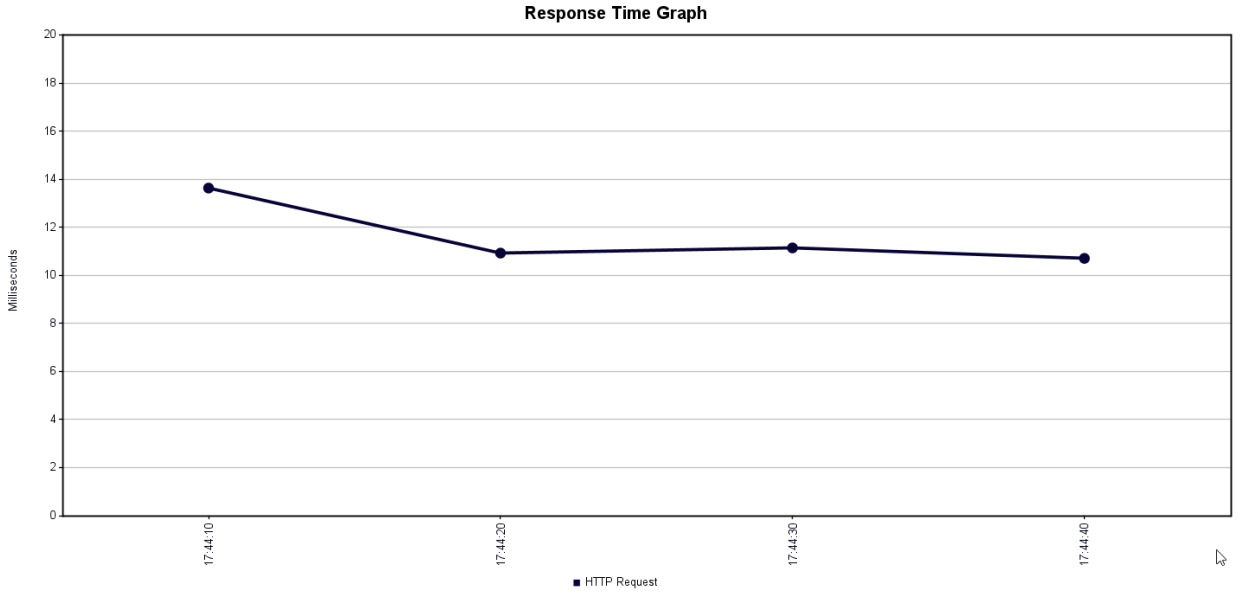
```
public List<AmbientSensorHistoryWebResponseDto> getAmbientSensorMeasurements(LocalDateTime startDateTime, LocalDateTime endDateTime,
String device, long id) {
    String startDateTimeString = String.valueOf(DateTimeUtility.convertToUnixTimestamp(startDateTime));
    String endDateTimeString = String.valueOf(DateTimeUtility.convertToUnixTimestamp(endDateTime));

    String fluxQuery = String.format("""
                from(bucket: "%s")
                    |> range(start: %s, stop: %s)
                    |> filter(fn: (r) => r["_measurement"] == "%s" and r["id"] == "%s")
                    |> filter(fn: (r) => r["_field"] == "%s" or r["_field"] == "%s")
                    |> pivot(rowKey:["_time"], columnKey: ["_field"], valueColumn: "_value")
                """, bucket, startDateTimeString, endDateTimeString, device, id,
            AMBIENT_SENSOR_FIELD_TEMPERATURE, AMBIENT_SENSOR_FIELD_HUMIDITY);

    return this.queryFlux(fluxQuery);
}
```

2. Dobavljanje istotorije (za poslednji 6 sata)

Testni slučaji: Za 5 nasumična uređaja vrši se dobavljanje podataka u poslednjih 6 sati.

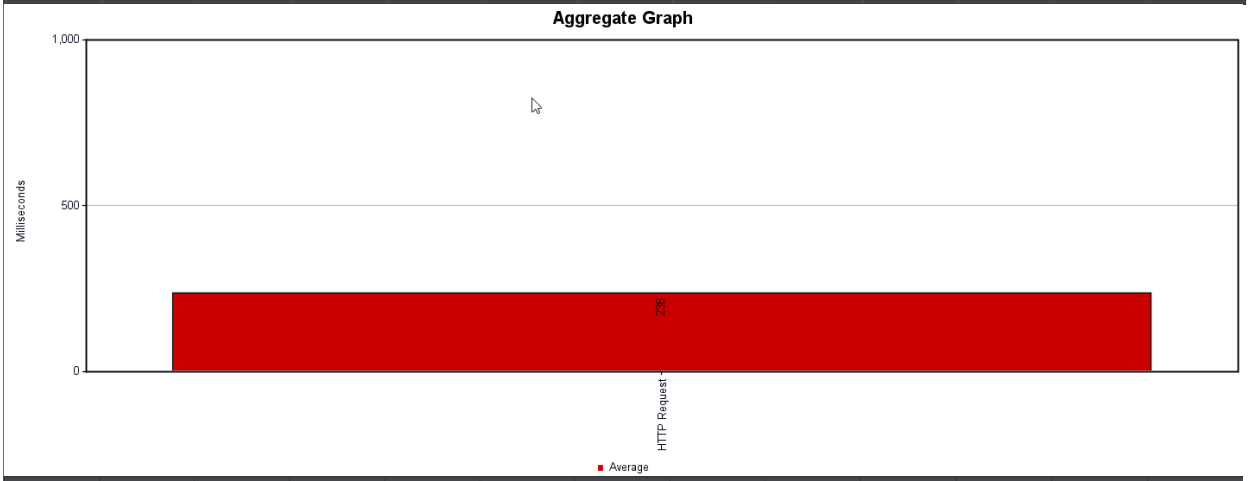
1 korisnik

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	3911	11	0	572	9.37	0.00%	86.9/sec	56.24	23.25	662.7
TOTAL	3911	11	0	572	9.37	0.00%	86.9/sec	56.24	23.25	662.7



50 korisnika

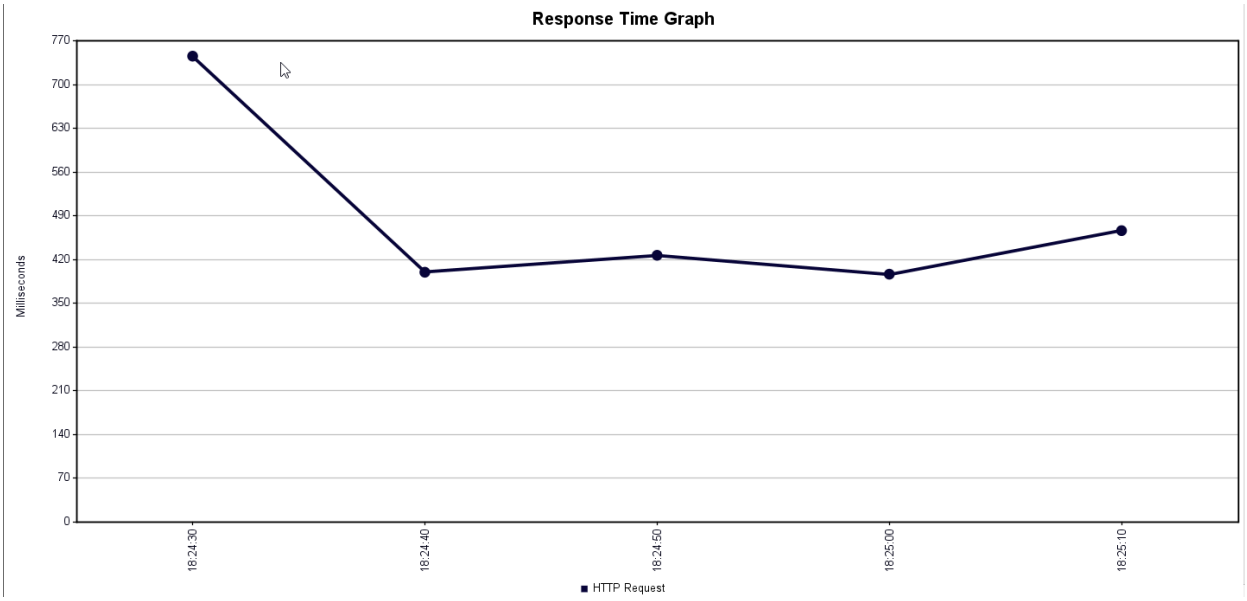
Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/...	Sent KB/sec
HTTP Request	9208	236	155	537	684	1000	22	1779	0.00%	203.6/sec	636.16	54.47
TOTAL	9208	236	155	537	684	1000	22	1779	0.00%	203.6/sec	636.16	54.47



Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/...	Sent KB/sec
HTTP Request	10052	435	198	1131	1430	2114	20	3375	0.00%	220.3/sec	672.19	58.96
TOTAL	10052	435	198	1131	1430	2114	20	3375	0.00%	220.3/sec	672.19	58.96

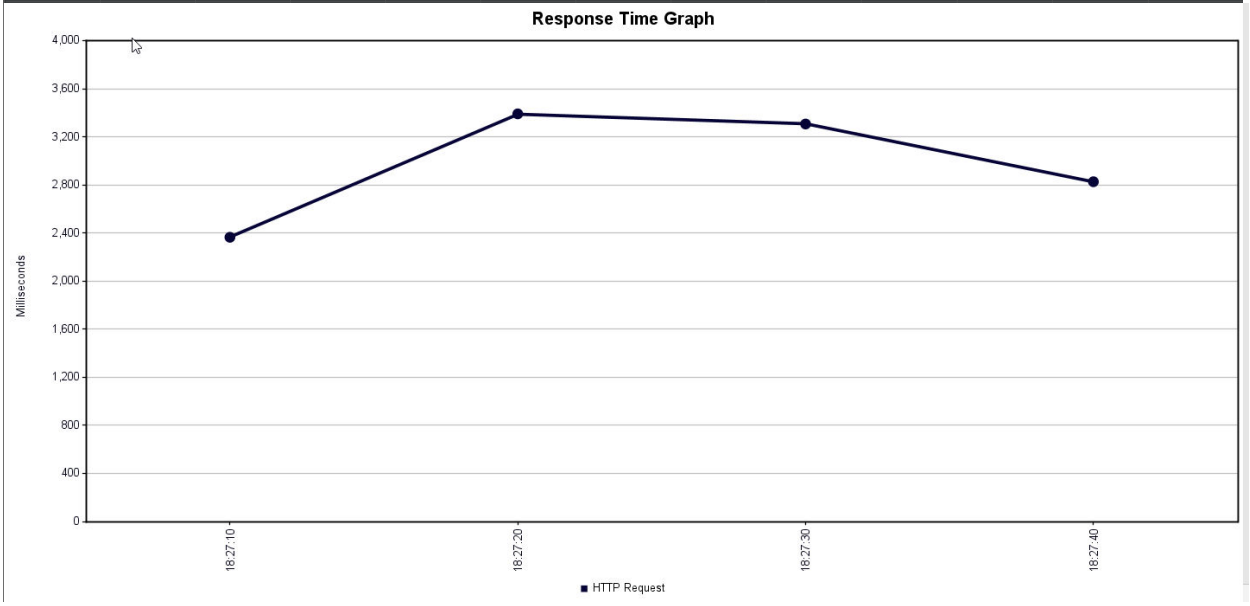
100 korisna

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/...	Sent KB/sec
HTTP Request	11878	368	193	902	1186	1794	15	3645	0.00%	261.4/sec	130.15	69.95
TOTAL	11878	368	193	902	1186	1794	15	3645	0.00%	261.4/sec	130.15	69.95



500 korisnika

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/...	Sent KB/sec
HTTP Request	8105	2771	321	9125	10074	13672	18	14625	7.77%	167.6/sec	476.29	44.84
TOTAL	8105	2771	321	9125	10074	13672	18	14625	7.77%	167.6/sec	476.29	44.84



Br. korisnika	1	50	100	500
Propusnost(sec)	88.6	203.0	261.4	167.6
Greska (%)	0.0	0.0	0.0	7.77

Testiranje Veš mašina

Uređaji ima slična ponasanja kao I abmijentalni senzor. Testiranje je izvrseno na nacin da je kreirano n ves masina sa istim zakazanim teminom. Rad veš mašine je postavljen na 30min. Odgovor je postavljen na 1 min

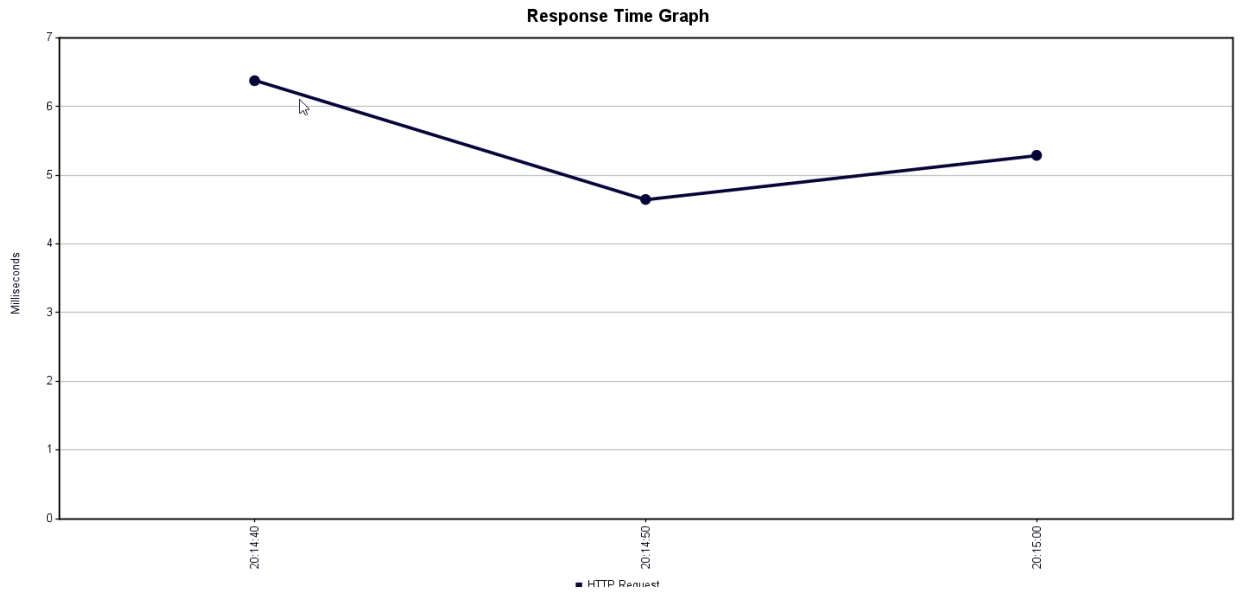
Br. Uređaja	1	10	100	1000	10.000
Vreme odgovora zakazivanje (ms)	30 - 50	30 - 50	30 - 50	80 – 150	160 - 180
Vreme odgovora otkazivanje (ms)	30 - 50	30 - 50	30 - 50	100 - 200	120 - 230
Vreme odgovora istorija događaka (ms)	30 - 50	30- 50	120 - 150	150 - 180	150 - 180

3. Dobavljanje istorije događaja

Tesni slučaji: Za 5 nasumična uređaja u bazi vrši se dobavljanje istorije događaja. U ovom delu izvršena je optimizacija upotrebom paginacije. Difoltni size je postavljen na 25

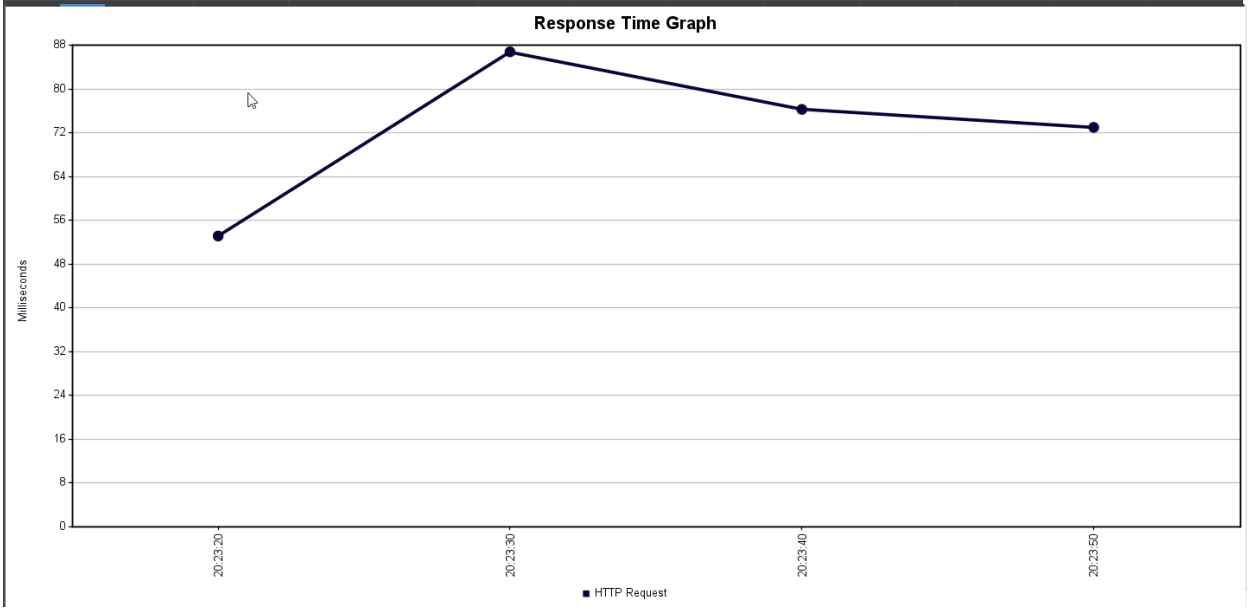
1 korisnik

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/...	Sent KB/sec
HTTP Request	7733	5	5	8	10	14	4	70	0.00%	171.8/sec	485.07	26.85
TOTAL	7733	5	5	8	10	14	4	70	0.00%	171.8/sec	485.07	26.85



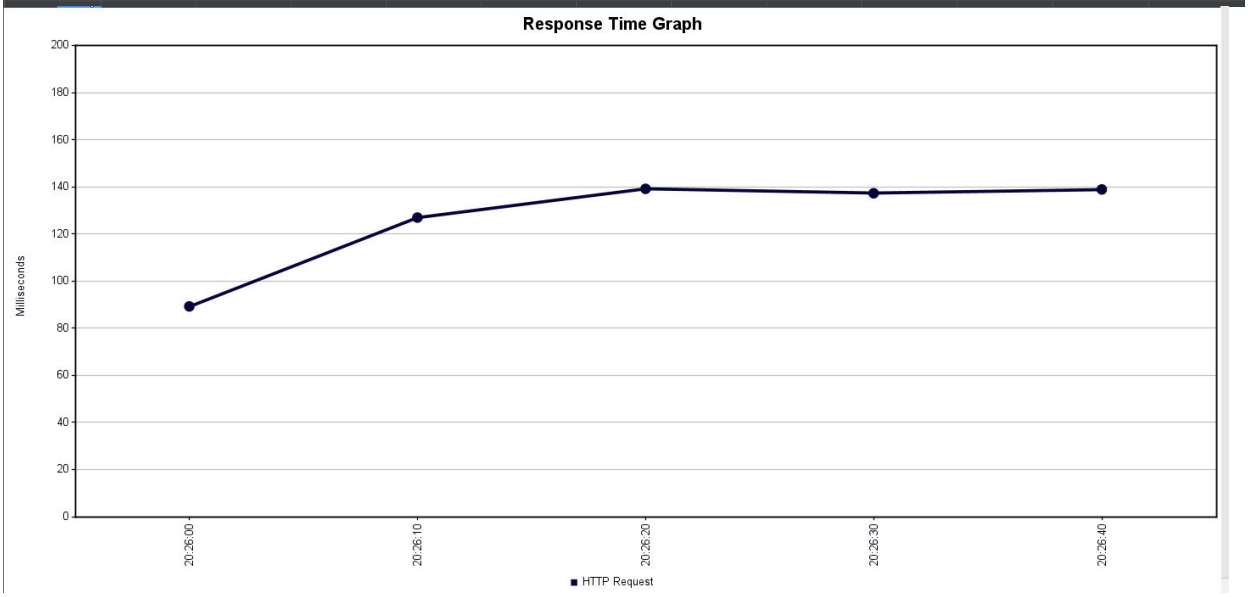
50 korisnika

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/...	Sent KB/sec
HTTP Request	30170	71	68	130	161	223	4	653	0.00%	669.3/sec	1889.34	104.57
TOTAL	30170	71	68	130	161	223	4	653	0.00%	669.3/sec	1889.34	104.57



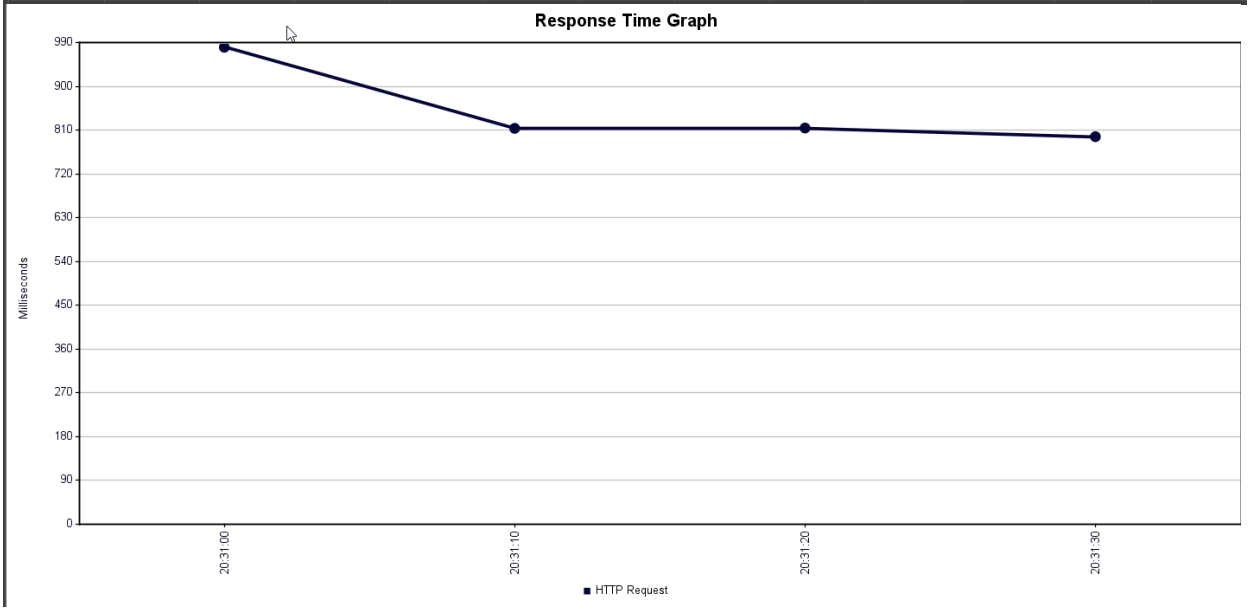
100 korisnika

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/...	Sent KB/sec
HTTP Request	32813	132	129	256	335	479	5	1264	0.00%	725.3/sec	2047.41	113.32
TOTAL	32813	132	129	256	335	479	5	1264	0.00%	725.3/sec	2047.41	113.32



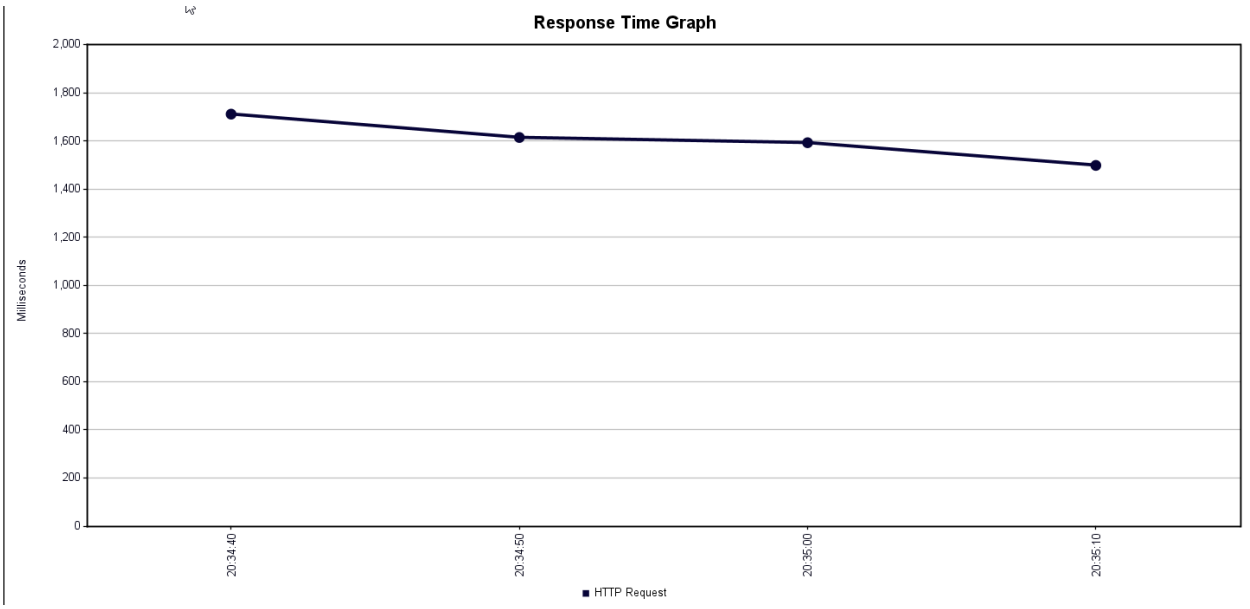
500 korisnika

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/...	Sent KB/sec
HTTP Request	26654	827	768	1621	2179	3083	10	6991	0.00%	578.8/sec	1634.00	90.43
TOTAL	26654	827	768	1621	2179	3083	10	6991	0.00%	578.8/sec	1634.00	90.43



1000 korisnika

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/...	Sent KB/sec
HTTP Request	26418	1573	1510	3142	4371	6165	12	12436	0.00%	557.1/sec	1572.91	87.05
TOTAL	26418	1573	1510	3142	4371	6165	12	12436	0.00%	557.1/sec	1572.91	87.05



Br. korisnika	1	50	100	500	1000
Propusnost(sec)	171.8	669.3	725.3	578.8	557.1
Greska (%)	0.0	0.0	0.0	0.0	0-0 – 4.0

4. Zakazivanje rada

Testni slučaji: Za jedan uređaji u bazi, za savku sekundu se vrsi zakazivanje. Vreme trajanja pranja je jedna 1s. Optimizacija koja je uvedena je upotreba optimistickog zaljučavanja kako bi se mogli koristiti i rekordi koji se edituju. Za istog korisnika za raličite veš mašine.

1 koirsni

Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
POST	/api/auth/login	1	0	631	631	631	227	0.0	0.0
PUT	Schedule	44	0	38	25	143	125	0.8	0.0
	Aggregated	**45**	**0**	**51**	**25**	**631**	**128**	**0.8**	**0.0**

Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
POST	/api/auth/login	630	630	630	630	630	630	630	630
PUT	Schedule	33	34	35	42	58	60	140	140
	Aggregated	**33**	**34**	**36**	**47**	**59**	**65**	**630**	**630**

10 korisnika

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
POST	/api/auth/login	10	0	261	192	433	227	0.2	0.0
PUT	Schedule	392	45	56	18	406	112	6.5	0.8
	Aggregated	**402**	**45**	**61**	**18**	**433**	**115**	**6.7**	**0.8**

Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
POST	/api/auth/login	230	250	250	420	430	430	430	430
PUT	Schedule	52	56	60	66	76	83	190	410
	Aggregated	**52**	**56**	**61**	**67**	**78**	**130**	**250**	**430**

50 korsnika

Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
POST	/api/auth/login	50	0	2228	1240	4082	227	0.8	0.0
PUT	Schedule	1720	1058	178	23	2191	49	28.8	17.7
	Aggregated	**1770**	**1058**	**236**	**23**	**4082**	**54**	**29.6**	**17.7**

Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
POST	/api/auth/login	2200	2400	2600	2900	3400	3600	4100	4100
PUT	Schedule	130	150	180	210	250	440	1300	2200
	Aggregated	**140**	**160**	**180**	**210**	**280**	**880**	**2500**	**4100**

5. Otkazivanje rad

Testni slučaji: Otkazivanje zakazanih termina za jedan uređaji u bazi.

Vrši se zaključavanje reda koji se brise zbog transakcije.

100 korsnika

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
POST	/api/auth/login	100	0	1400	2400	3800	1468	266	3826	227	0	0
GET	cancel	100	0	1900	2700	3500	1921	59	3466	131972	0	0
PUT	cancel	3549	3478	1400	2500	3500	1567	133	4907	33	45.5	44.6
	Aggregated	3749	3478	1400	2500	3500	1573	59	4907	3557	45.5	44.6

Br. korisnika	1	50	100
Prosecno vreme odogovra (ms)	50	800	1400
Greska (%)	0.0	> 90%	> 90%

Testiranje Klima uređaja

Odgovor od strane klima uredjaja se salje na svakih 5s

Kao i u prethodnim slucajevima, dolazi do pucanja unutar mqtt-a.

Br. Uređaja	1	10	100	1000
Vreme odgovora zakazivanje (ms)	30 - 50	40 - 60	120 - 150	180 - 250
Vreme odgovora otkazivanje (ms)	30 - 50	40 - 60	120 - 150	180 - 250
Vreme odgovora istorija događaja (ms)	30 - 50	40 - 60	120 - 150	180 - 250

6. Istorija klima uređaja

Br. korisnika	1	50	100	500	1000
Propusnost(sec)	174.8	672.1	733.1	559.4	533.9
Greska (%)	0.0	0.0	0.0	0.0	0-0 – 4.2

Permisije

7. Dobavljanje permisije nad uređajem

Testni slučaji: Svaka kuca ima po jedan uređaji. Testirati dodavanje

NAPOMENA: Greske koje se vode pod failiures su greske izazvane od strane provere validnosti podataka. Testirano je za 1000 korisnika, 1000 kuca I 1000 uređaja u bazi

1 korinik

Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
POST	/api/auth/login	1	0	744	744	744	231	0.0	0.0
POST	permissions	10	0	73	54	190	222	0.4	0.0
	Aggregated	**11**	**0**	**134**	**54**	**744**	**223**	**0.5**	**0.0**

Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
POST	/api/auth/login	740	740	740	740	740	740	740	740
POST	permissions	60	64	66	69	190	190	190	190
	Aggregated	**60**	**64**	**66**	**69**	**190**	**740**	**740**	**740**

50 korisnika

Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
POST	/api/auth/login	1	0	614	614	614	231	0.0	0.0
POST	permissions	1143	246	16	12	167	191	52.5	11.3
	Aggregated	**1144**	**246**	**17**	**12**	**614**	**191**	**52.5**	**11.3**

Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
POST	/api/auth/login	610	610	610	610	610	610	610	610
POST	permissions	15	16	16	18	21	23	44	170
	Aggregated	**15**	**16**	**16**	**18**	**21**	**24**	**52**	**610**

100 korisnika

Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
POST	/api/auth/login	10	0	255	100	406	231	0.6	0.0
POST	permissions	1615	1315	72	9	324	88	97.0	79.0
	Aggregated	**1625**	**1315**	**73**	**9**	**406**	**89**	**97.6**	**79.0**

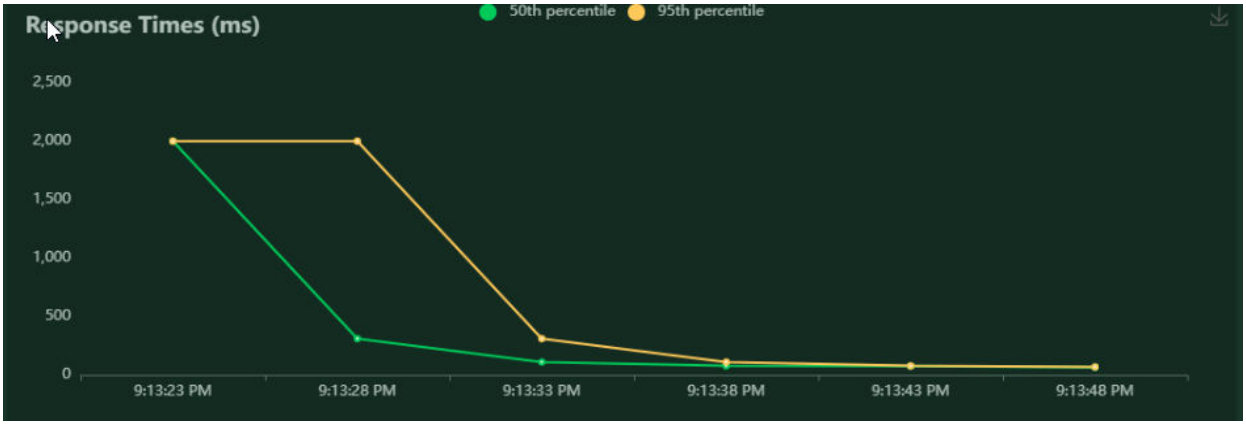
Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
POST	/api/auth/login	250	310	380	380	410	410	410	410
POST	permissions	70	79	86	97	110	140	170	320
	Aggregated	**70**	**79**	**86**	**97**	**120**	**140**	**180**	**410**

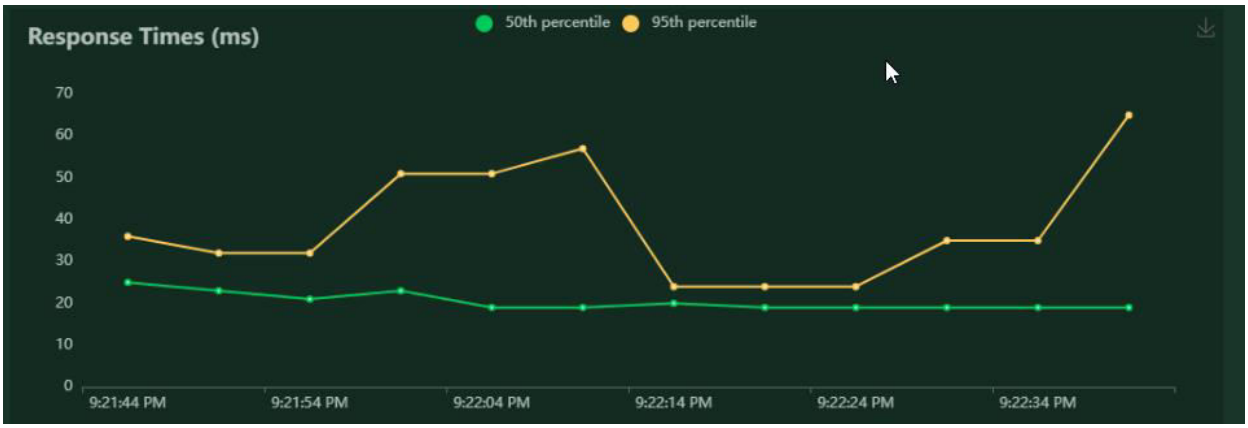
Br. korisnika	1	50	> 100
Vreme odgovora(ms)	66	44	86
Greska (%)	0.0	0.0	0.0

8. Uklanjanje permisije nad ure

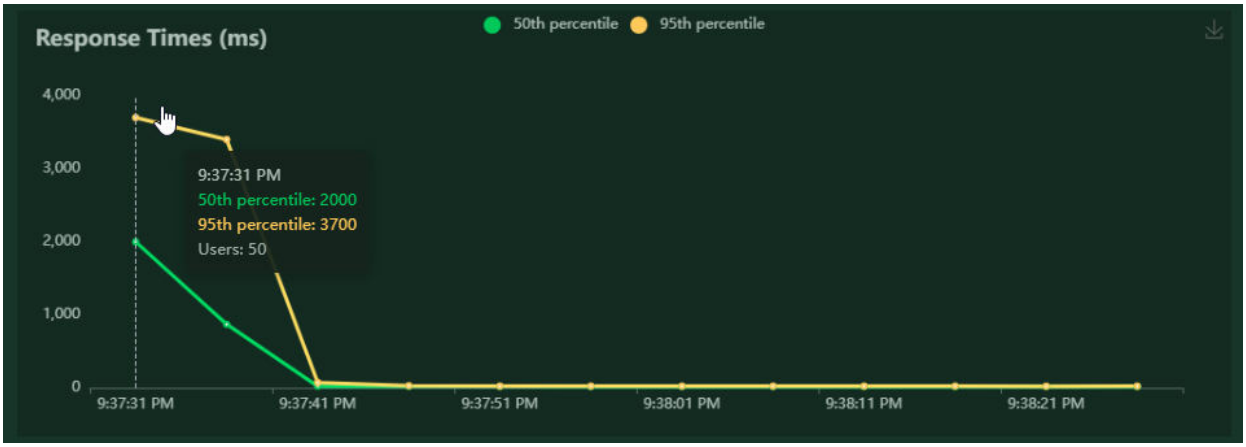
1 korisnik



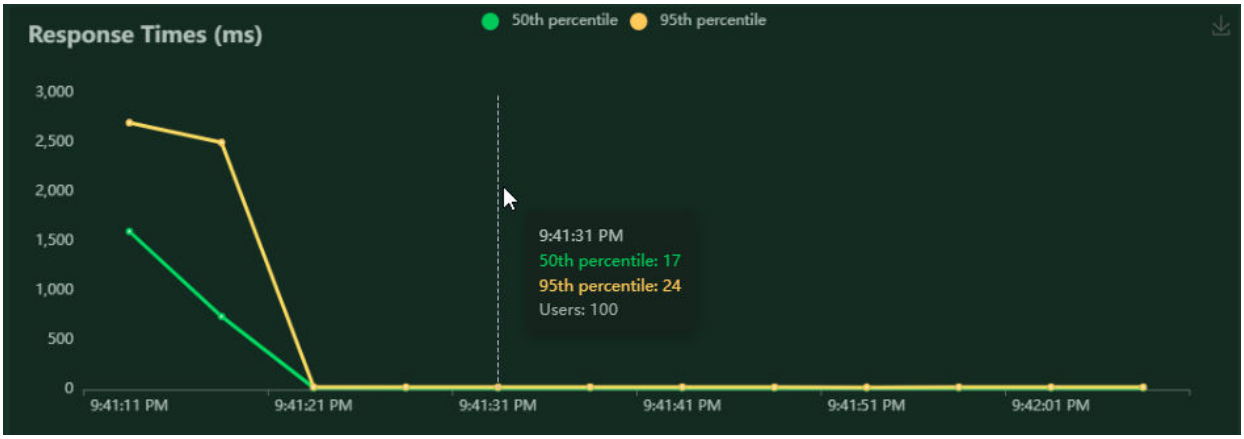
10 korisnika



50 korisnika



100 korsnika



Br. korisnika	1	10	50	100
Vreme odgovora(ms)	110	44	86	95
Greska (%)	0.0	0.0	0.0	0.0

9. Dobavljanje datih permisija

Za 5 razlicita korisnika dobavlaju se date permisije. Svaki korisnik ima ukupno 100 datih permisija.

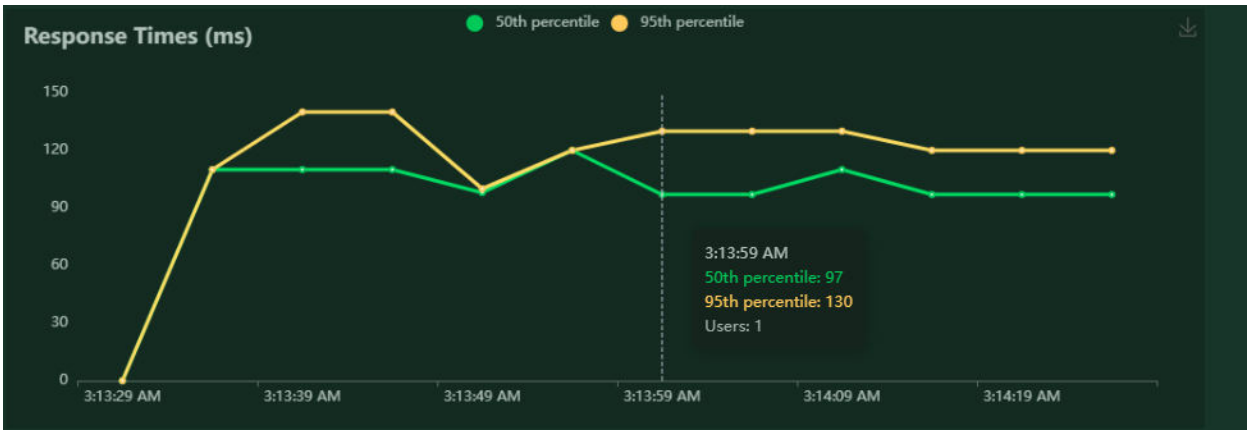
Br. korisnika	1	50	100	500	1000
Propusnost(sec)	164.8	645.2	715.3	569.1	542.1
Greska (%)	0.0	0.0	0.0	0.0	0-0 – 3.8

Dodavanje korisnika

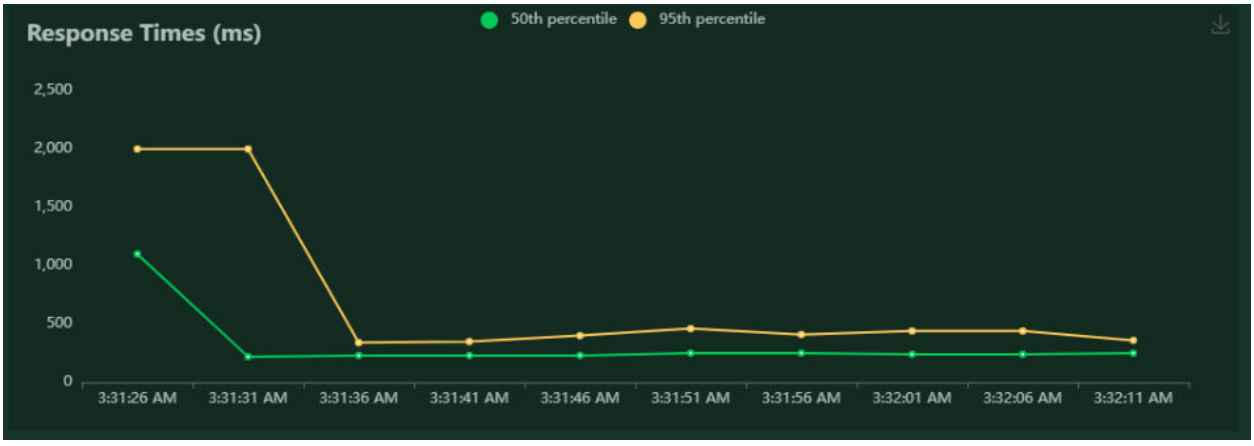
10. Registracija obicnog korisnika

Tesni slucaji: za nasumican email vrsi se registracija novog korisnika.

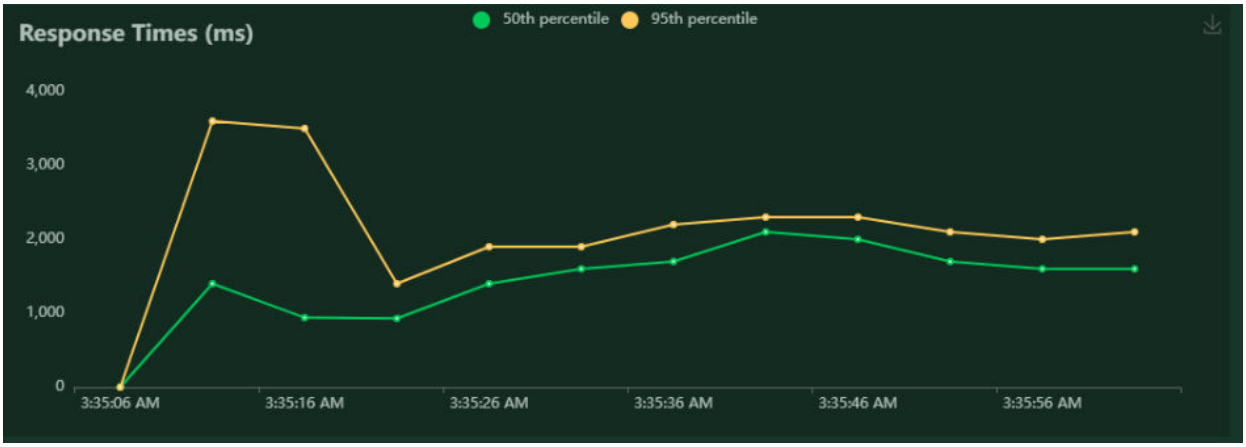
1 korisnik



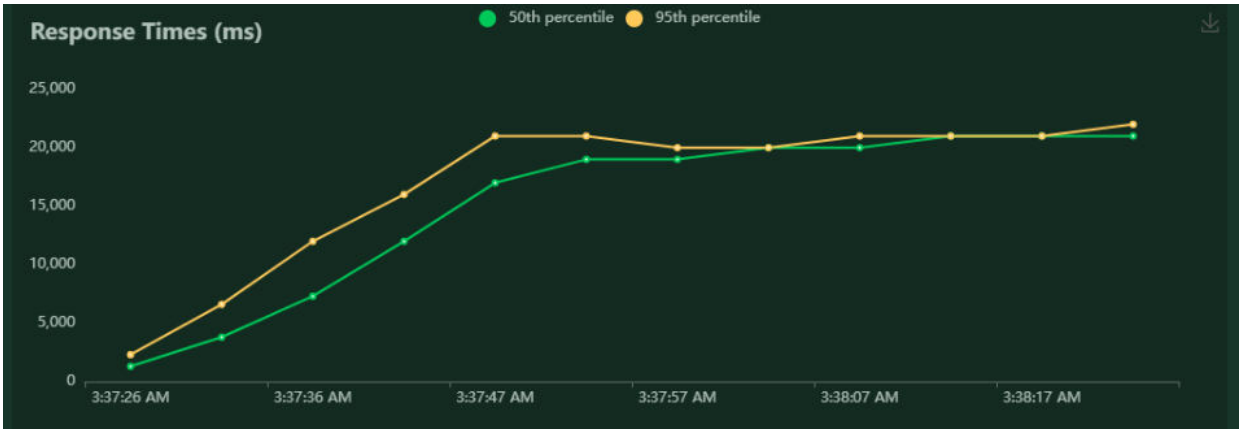
50 korisnik



100 korisnik



500 korisnik



Br. korisnika	1	50	100	500
Vreme odgovora(ms)	120	240	1700	16321
Greska (%)	0.0	0.0	0.0	1.0