

# Izveštaj

Student: Strahinja Sekulić

## Specifikacija računara

Navedeni testovi su izvršeni na jednom laptopu sledećih specifikacija:

- Procesor: I7 10875H " 2.30GHz (up to 4.3GHz)
- Ram: 16gb 3GHz
- SSD: NVMe
- GPU: RTX 2060 max-q

## Test čestih scenarija korišćenja

- Testovi su rađeni uz pomoć JMeter alata. Svaki test je pokretan više puta sa različitim konfiguracijama (broj korisnika, broj ponovnog pokretanja - Number of Threads).
- Cilj ovog tipa testiranja jeste da vidimo koliki load-a sistem može da podrži i da saznamo koliko vremena je potrebno za obradu nekog zahteva u takvim ekstremnim uslovima.
- U toku testiranja sam naišao na dosta sličnih rezultate te njih nisam navodio, već sam naglasio zašto su slični (gotovo identični) i sa kojim scenariom se poklapaju.

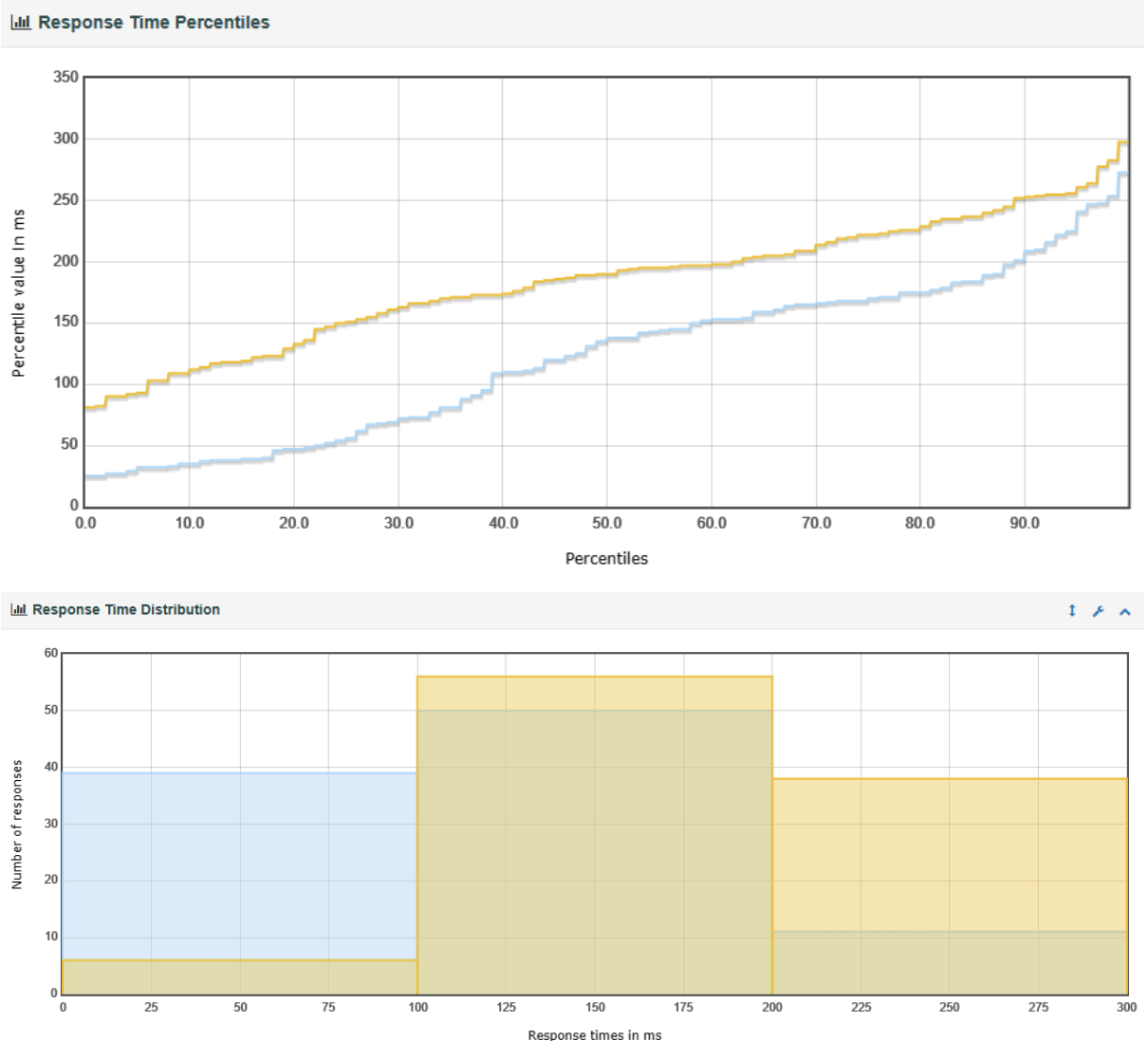
# #1 Kreiraj kapiju

## Scenario

- Korisnik se prijavi i kreira novu kapiju.

## Rezultati

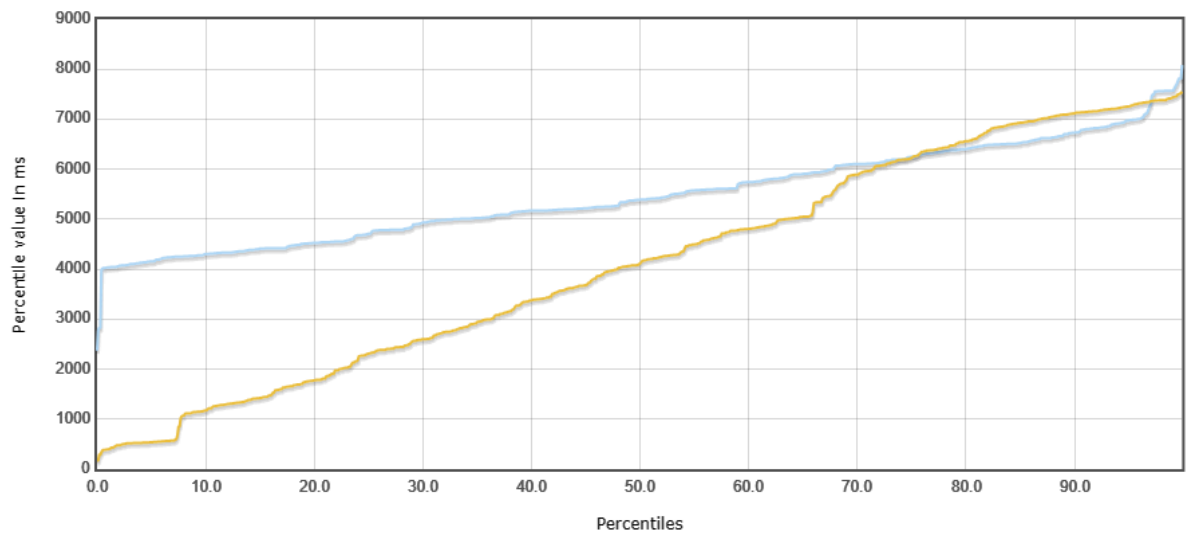
- Za kapiju sam pokretao više slučajeva kako bih stekao osećaj o ponašanju sistema i slede dobijeni rezultati
- 100 korisnika zahteva (pokrenuto 100 simulacija) - ukupno 1s



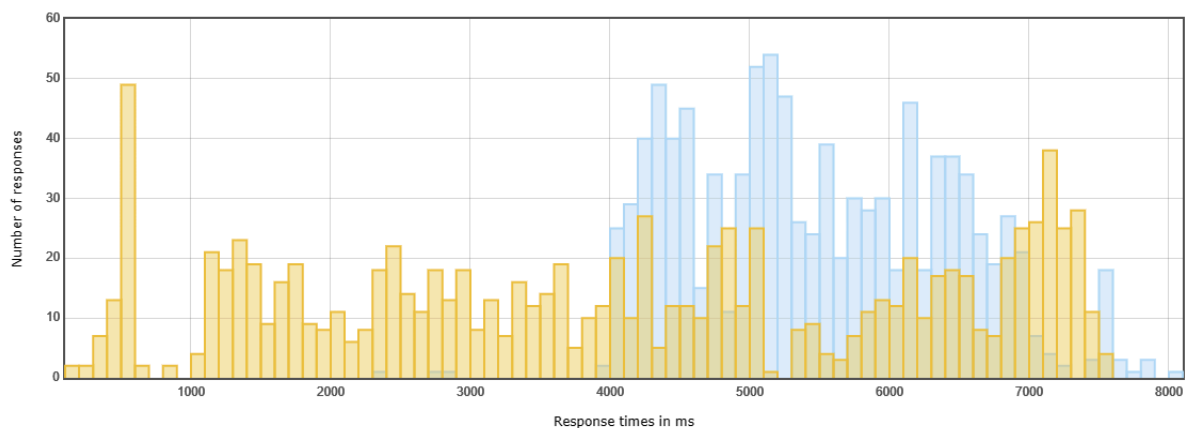
- Svi zahtevi su prošli.
- Primetimo da je sistem brz i da korisnici nisu čekali duže od 300ms

- 1000 korisnika zahteva (pokrenuto 100 simulacija) - ukupno 12s

**Response Time Percentiles**



**Response Time Distribution**

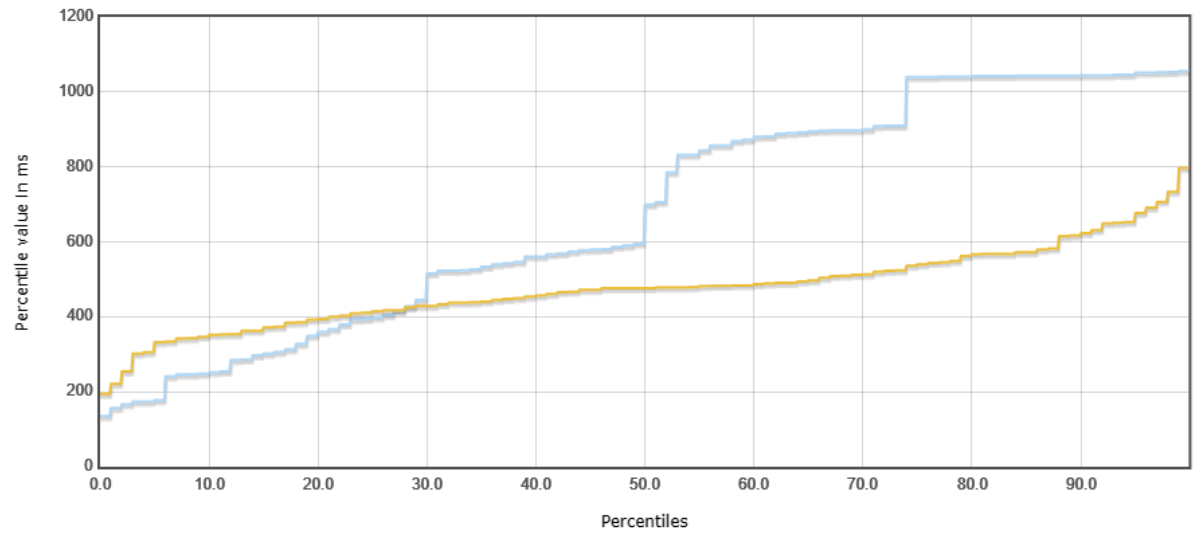


- Sada već primetimo da je sistem daleko sporiji te da se za odgovor čeka i do 8 sekundi ali su i dalje svi zahtevi bili ispunjeni
- >1000 korisnika (pokrenuto 100 simulacija) - Ovaj slučaj je već bacao grešku da nema dovoljno memorije kako bi se izvršio

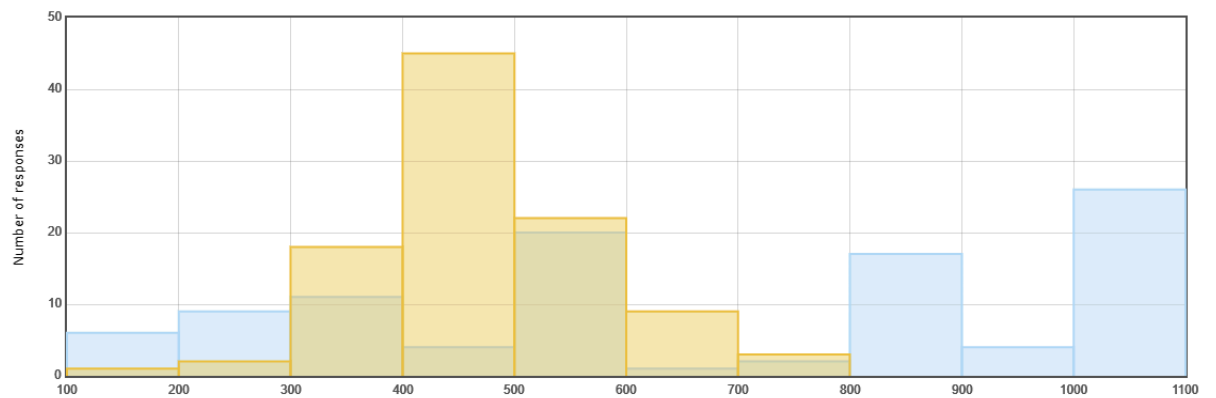
```
java.lang.OutOfMemoryError
#
# There is insufficient memory for the Java Runtime Environment to continue.
# Native memory allocation (malloc) failed to allocate 32756 bytes for ChunkPool::allocate
# An error report file with more information is saved as:
# C:\Users\XPS17\Downloads\apache-jmeter-5.6.3\apache-jmeter-5.6.3\bin\hs_err_pid20804.log
errorlevel=1
Press any key to continue . . .
```

- 100 korisnika (pokrenuto 1000 simulacija) - 2s, svih 100 je prošlo

**Response Time Percentiles**

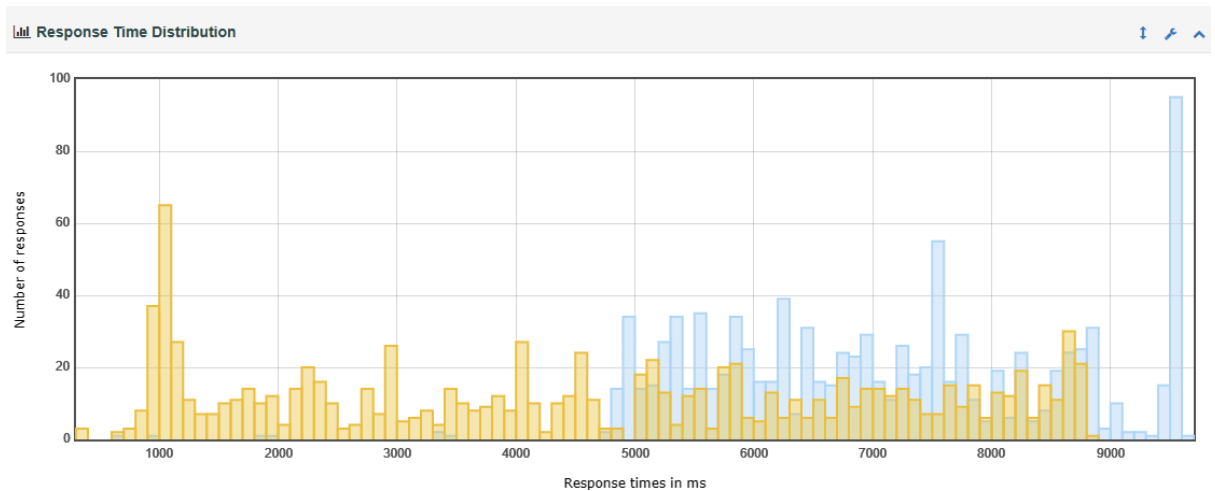
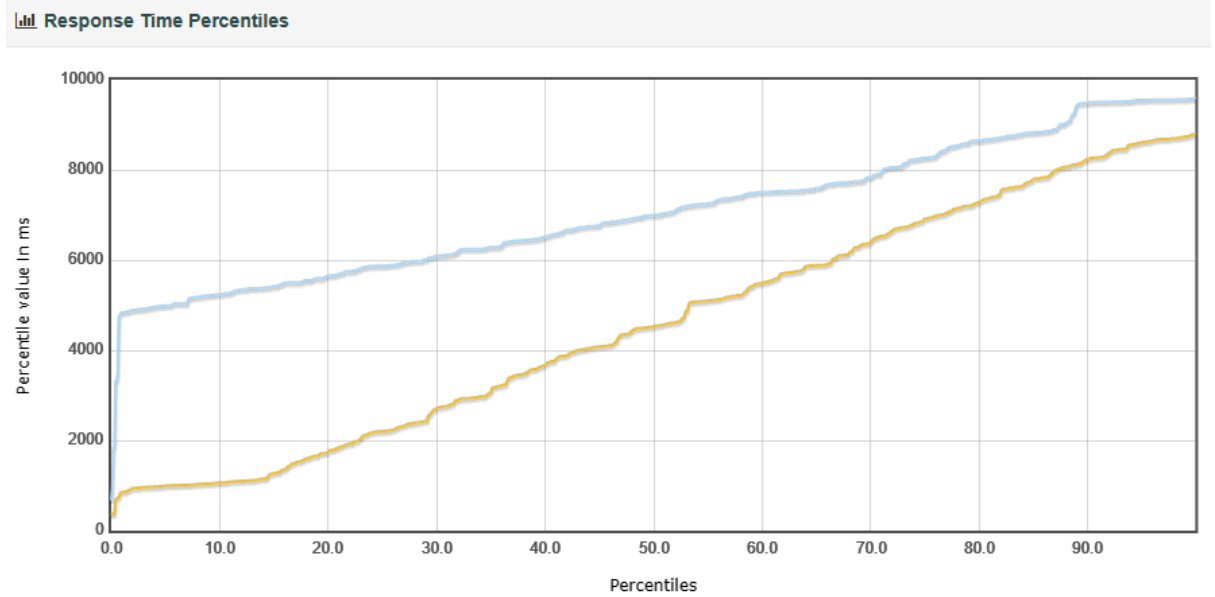


**Response Time Distribution**



- Primetimo da je sistem znatno sporiji zbog 1000 uređaja ali i dalje je prilično prihvatljivo


- 1000 Korisnika (pokrenuto 1000 simulatora) - 15s, svi su prošli

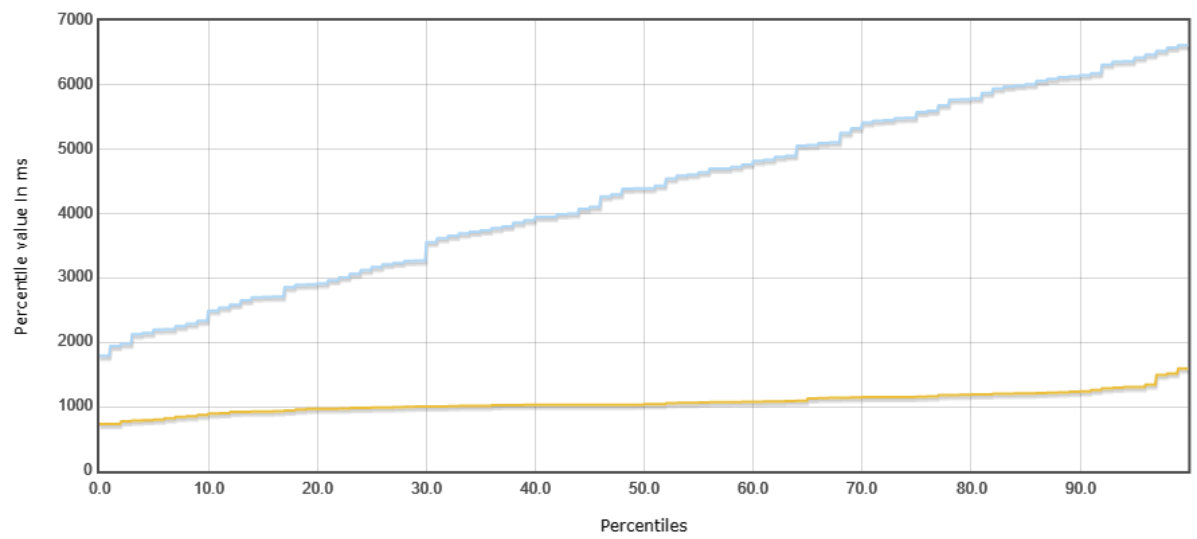


- Sistem je nešto sporiji u odnosu na 1000/100, ali ne previše

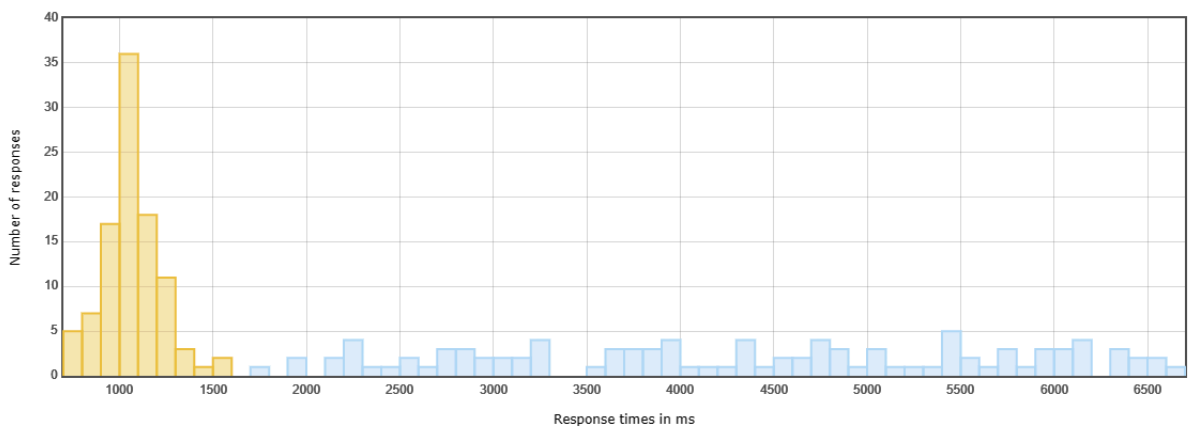
- >1000 korisnika (pokrenuto 1000 simulacija) - Ovaj slučaj također ne radi

- 100 korisnika (pokrenuto 10.000 simulacija) - 9s, svi su prošli

 Response Time Percentiles

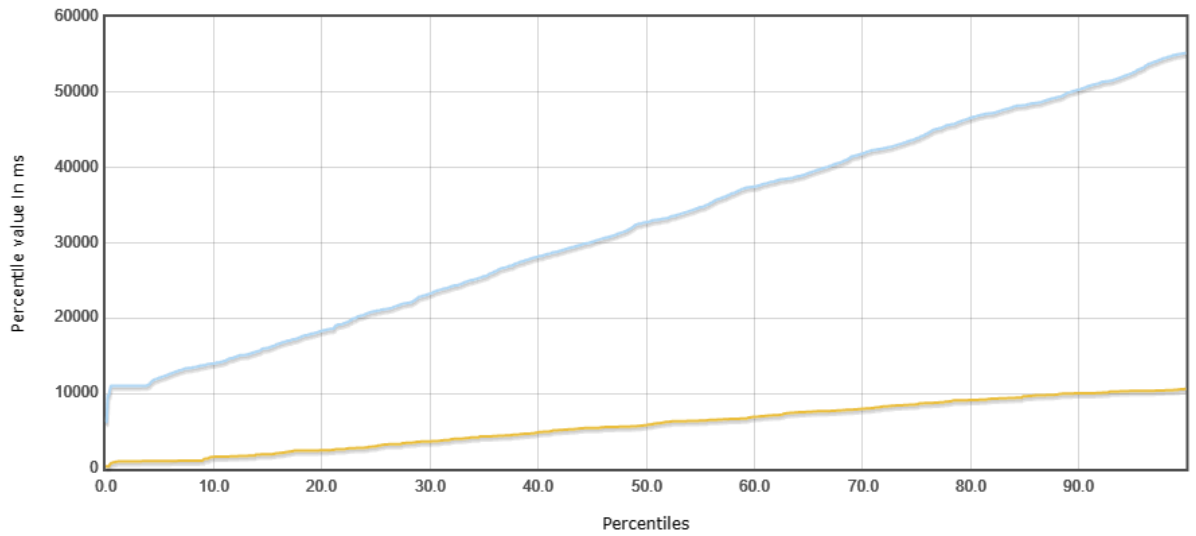


 Response Time Distribution

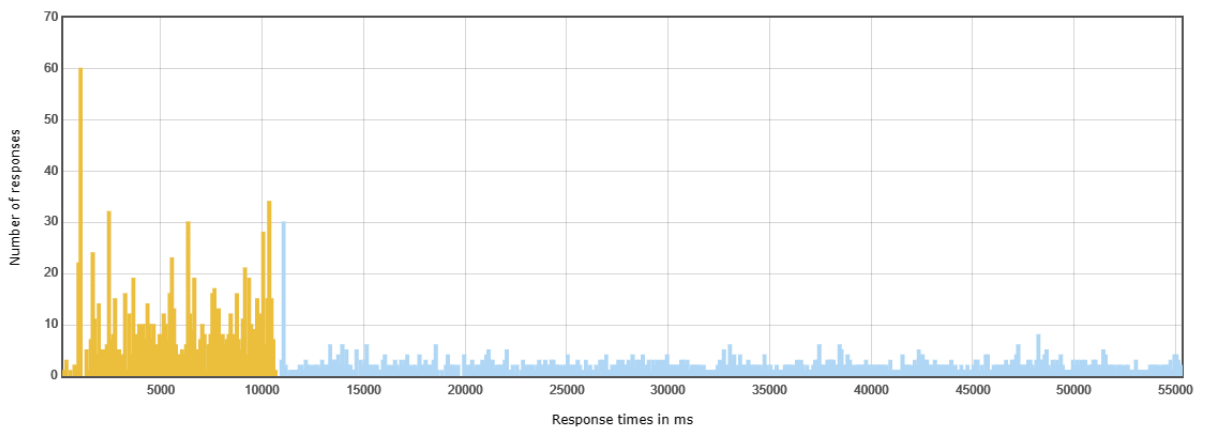


- 1000 korisnika (pokrenuto 10.000 simulatora) - 67s, svi su prošli

Response Time Percentiles



Response Time Distribution



- Primetimo da je sistem izuzetno spor i da su neki korisnici čekali i po 60s ali i dalje su svi prošli

## #2 Kreiraj lampu

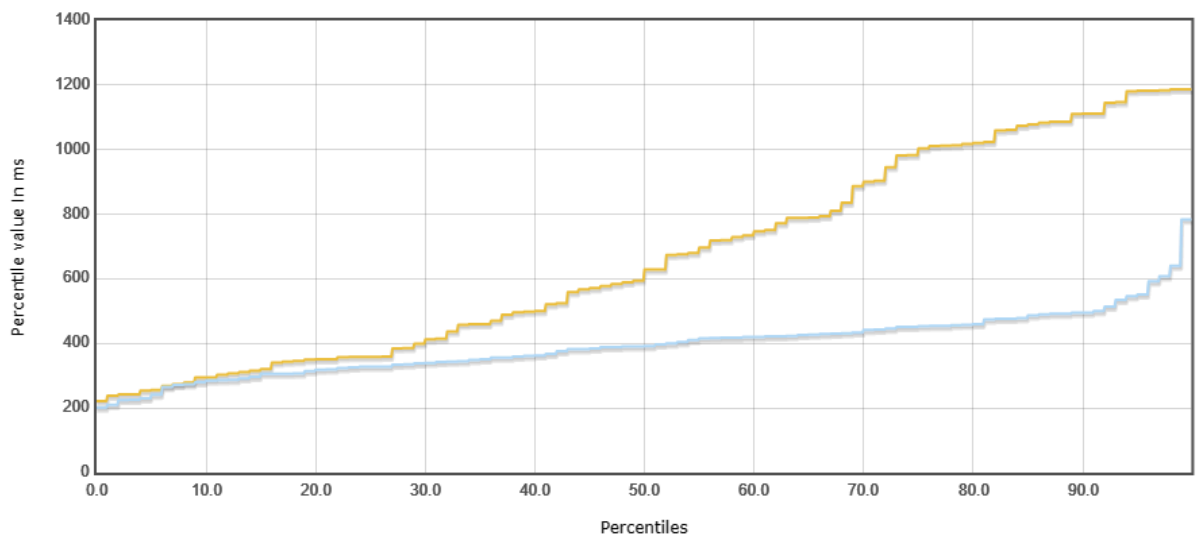
### Scenario

- Korisnik se prijavi i kreira novu lampu.

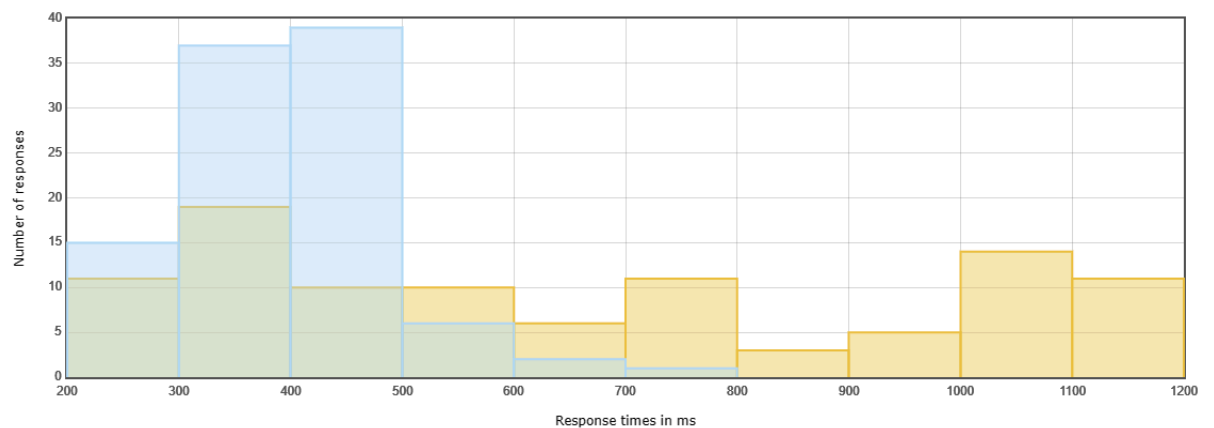
### Rezultati

- Rezultati su slični kao kod kreiranja kapije ali primetimo malo ubrzanje koje je rezultat jednog manje parametra koji treba obraditi. Iz ovog razloga nisam dokumentovao istu količinu scenarija jer su rezultati bili srazmerni sa #1
- 100 korisnika (1000 pokrenutih simulacija) - 1s, svi su prošli

Response Time Percentiles



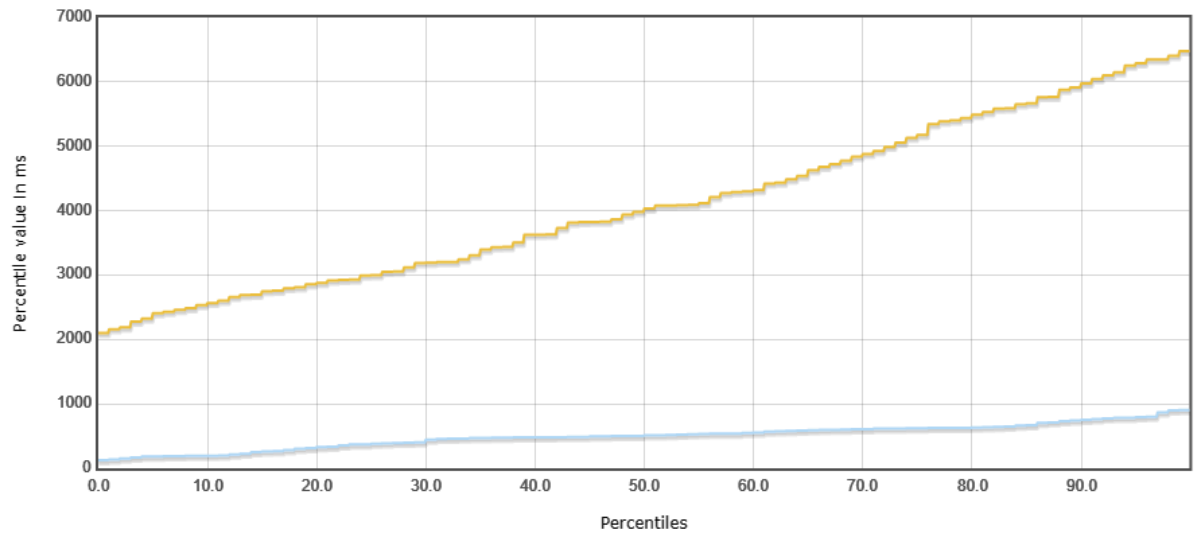
Response Time Distribution



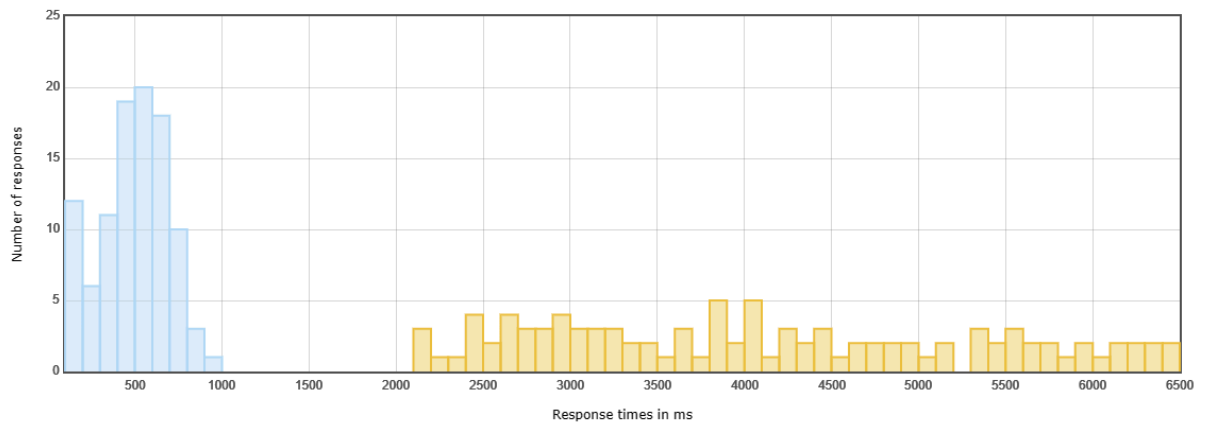


- 100 korisnika (10.000 pokrenutih simulacija) - 8s, svi su prošli

 Response Time Percentiles

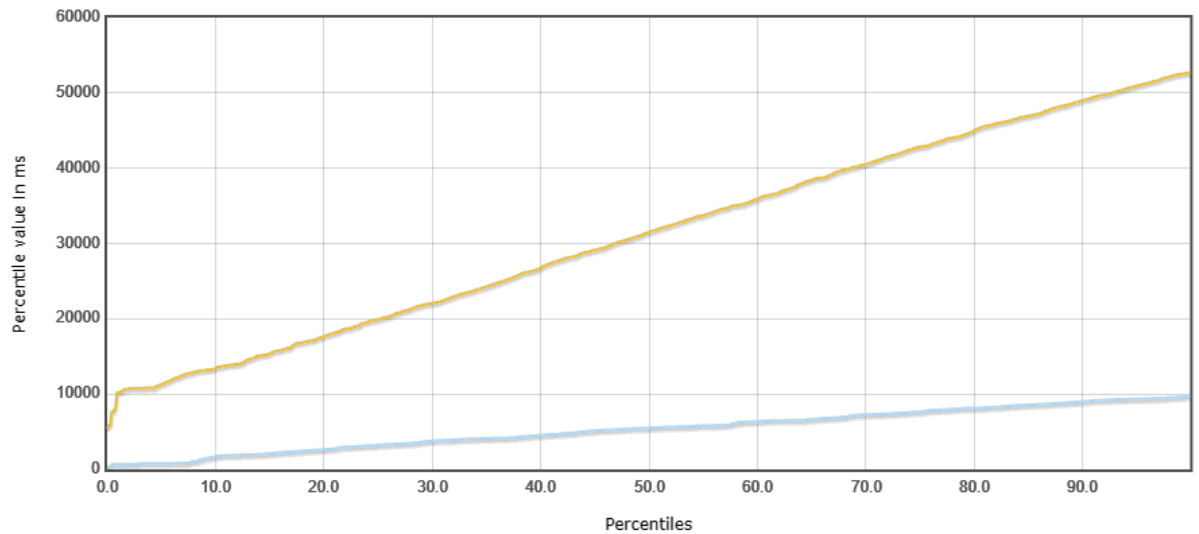


 Response Time Distribution

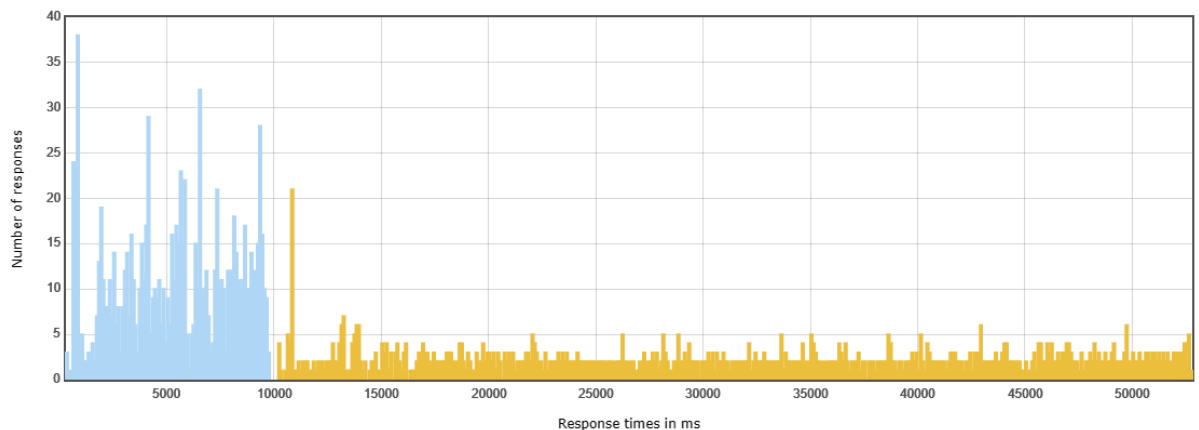


- 1000 korisnika (10.000 pokrenutih simulacija) - 65s, svi su prošli

**Response Time Percentiles**



**Response Time Distribution**



### #3 Kreiraj sistem prskalice

#### Scenario

- Korisnik se prijavi i kreira novu prskalicu.

#### Rezultati

- Ovaj test scenario je dao gotovo identične rezultate kao i scenario kreiranje lampe (#2) jer su podaci isti.

## #4 Promena režima rada kapije

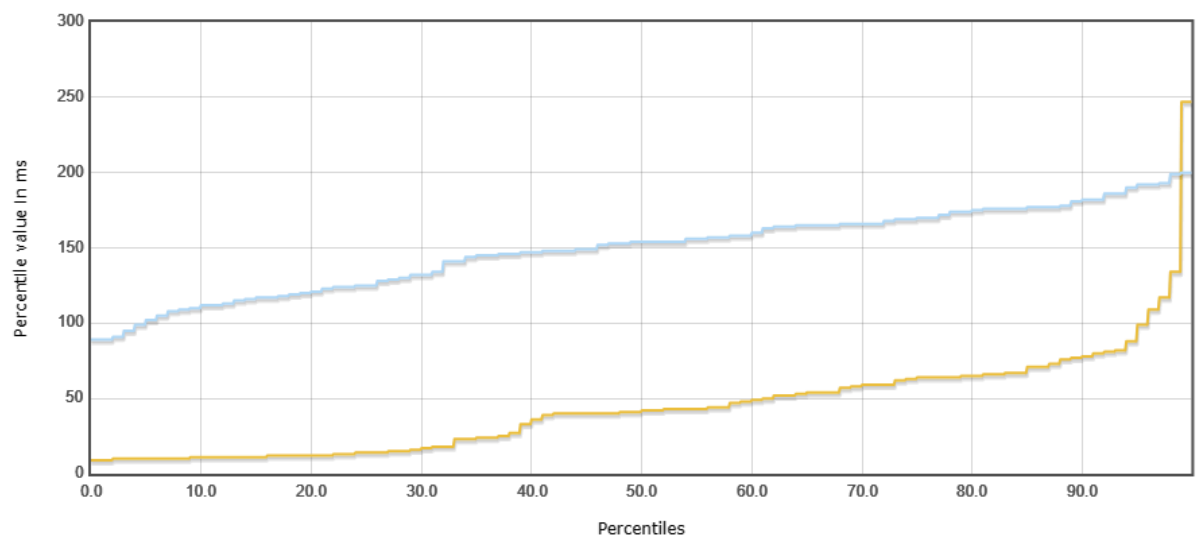
### Scenario

- Korisnik se prijavi na sistem i promeni režim rada kapije iz privatnog u javni.

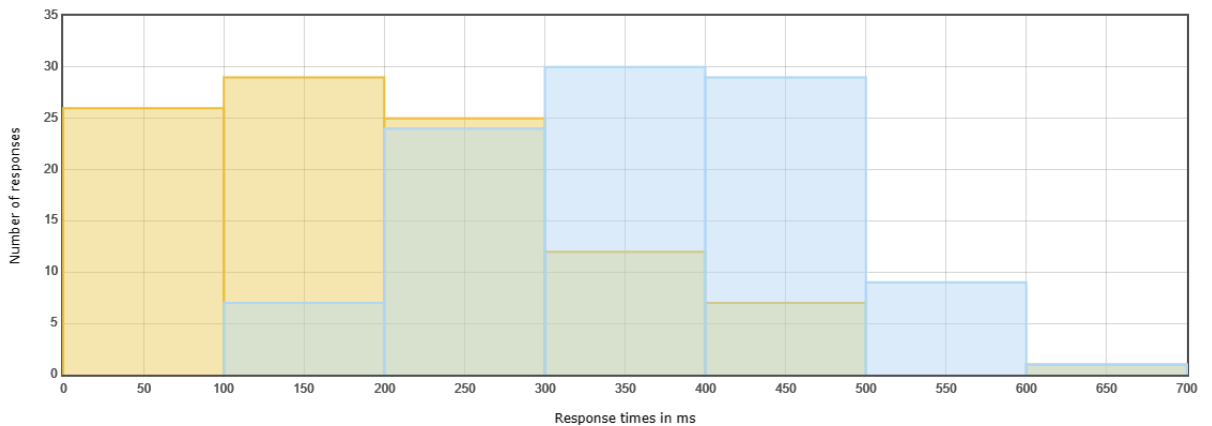
### Rezultati

- 100 korisnika (100 pokrenutih simulacija) - 1s, svi su prošli

Response Time Percentiles



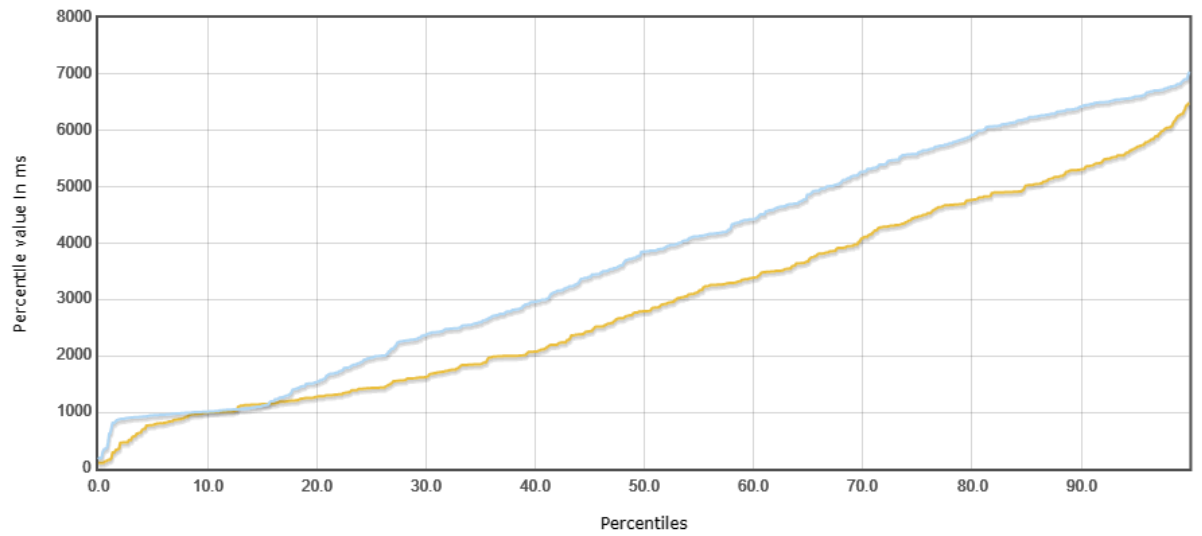
Response Time Distribution



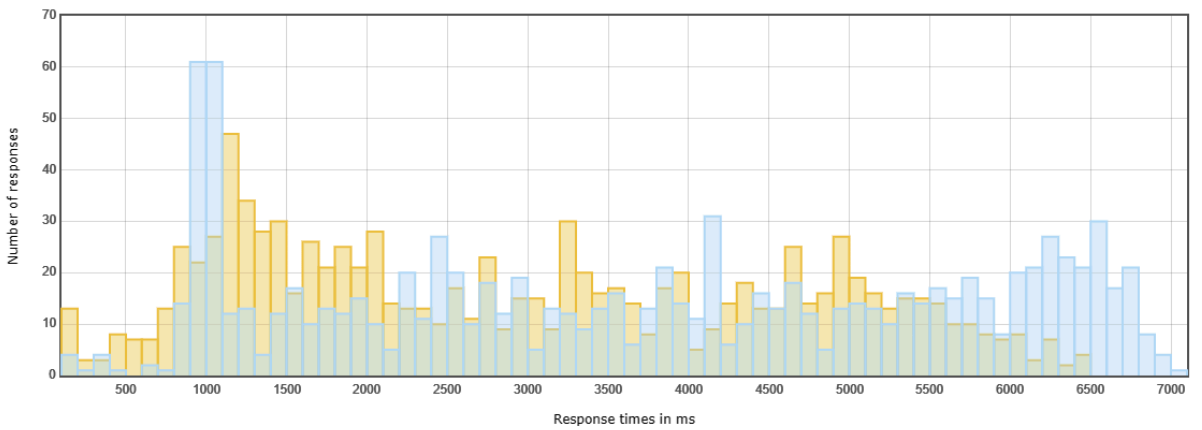
- Ovo bi bio neki baseline. Prosečan response time je oko 300ms

- 1000 korisnika (100 pokrenutih simulacija) - 9s, svi su prošli

Response Time Percentiles



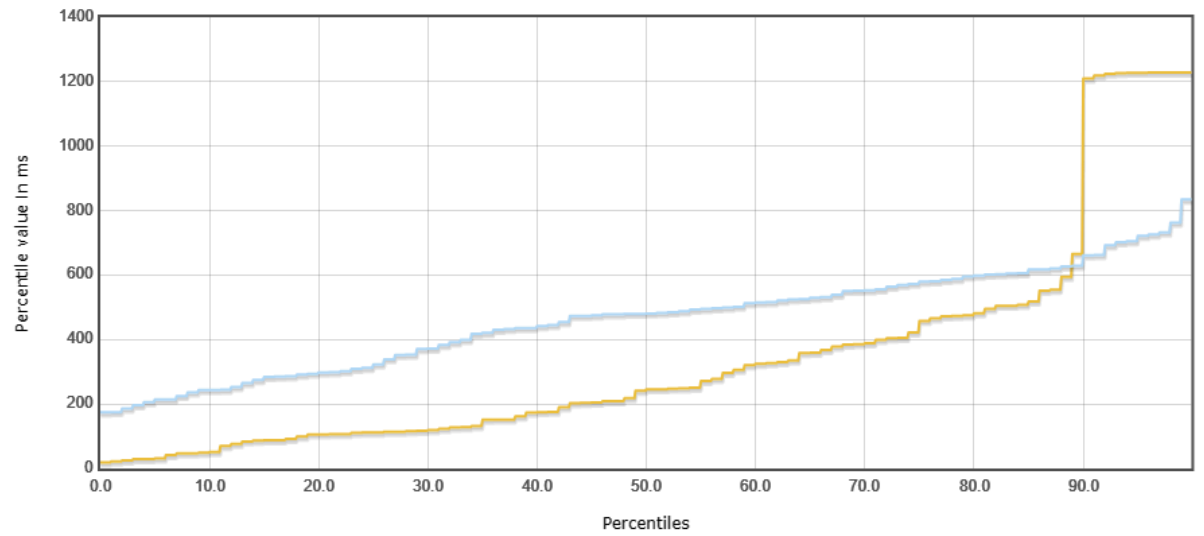
Response Time Distribution



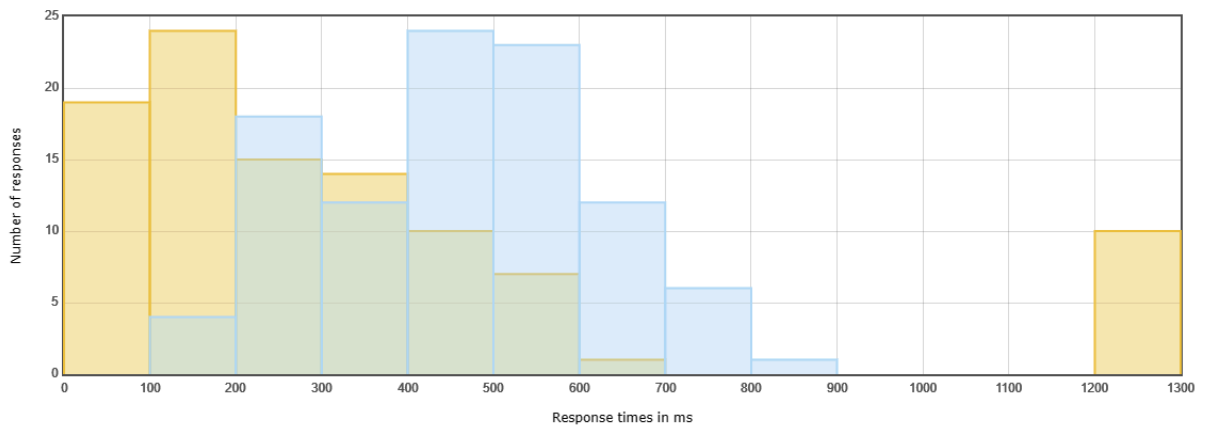
- Primetimo da je response time znatno opao (u proseku 3s)

- 100 korisnika (1000 pokrenutih simulacija) - 1s, svi su prošli

**Response Time Percentiles**



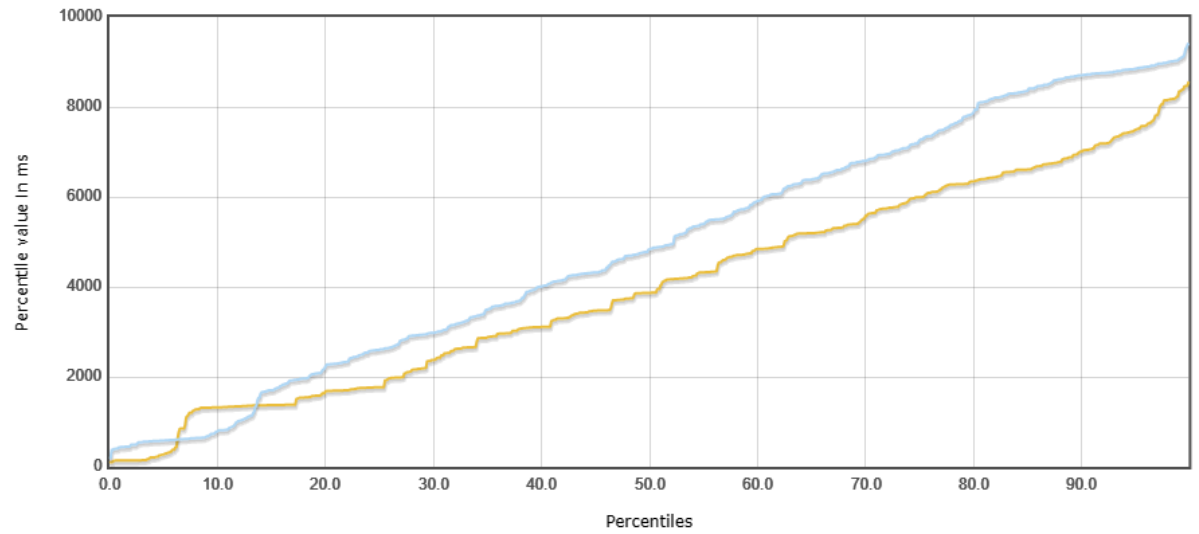
**Response Time Distribution**



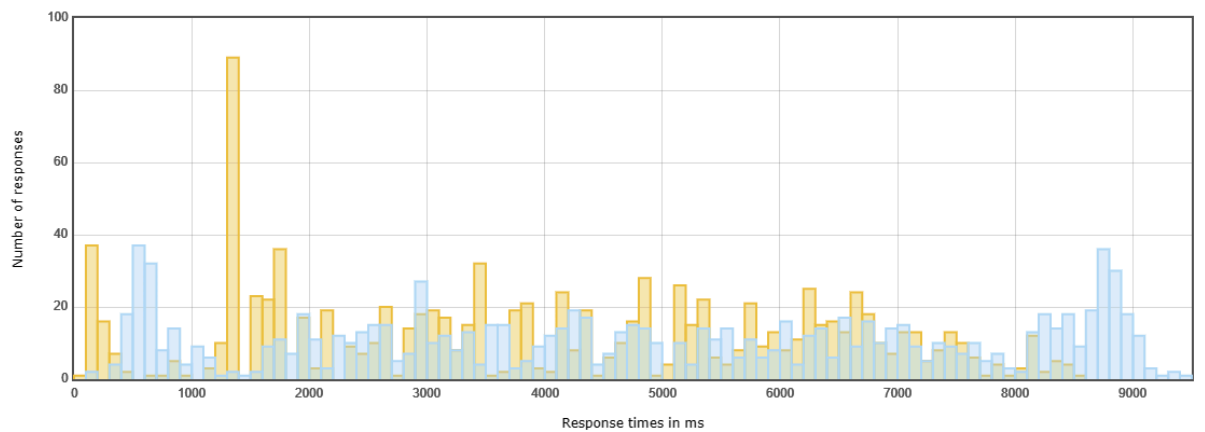
- Response time je ostao prihvatljiv za dobar deo korisnika ali primetimo da je neki broj cekao >1s zbog povecanog broja simulacija

- 1000 korisnika (1000 pokrenutih simulacija) - 11s, svi su prošli

 Response Time Percentiles



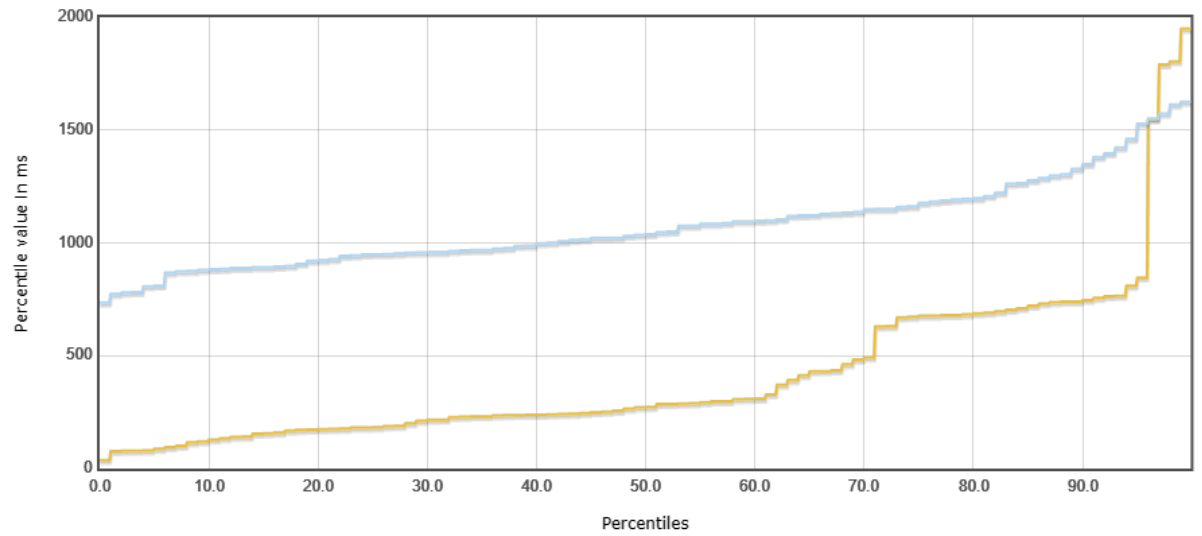
 Response Time Distribution



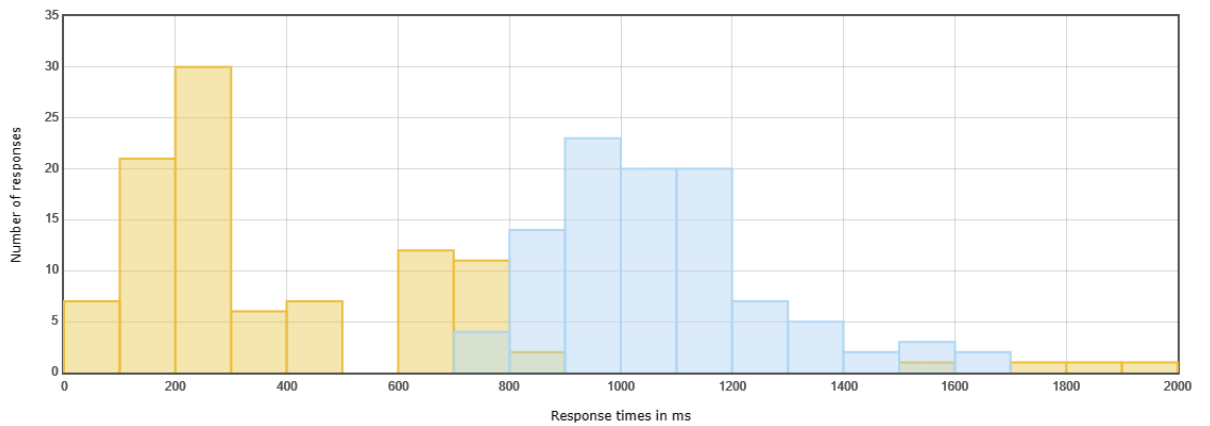
- Prosečena response time je 4s

- 100 korisnika (10.000 pokrenutih simulacija) - 3s, svi su prošli

 Response Time Percentiles



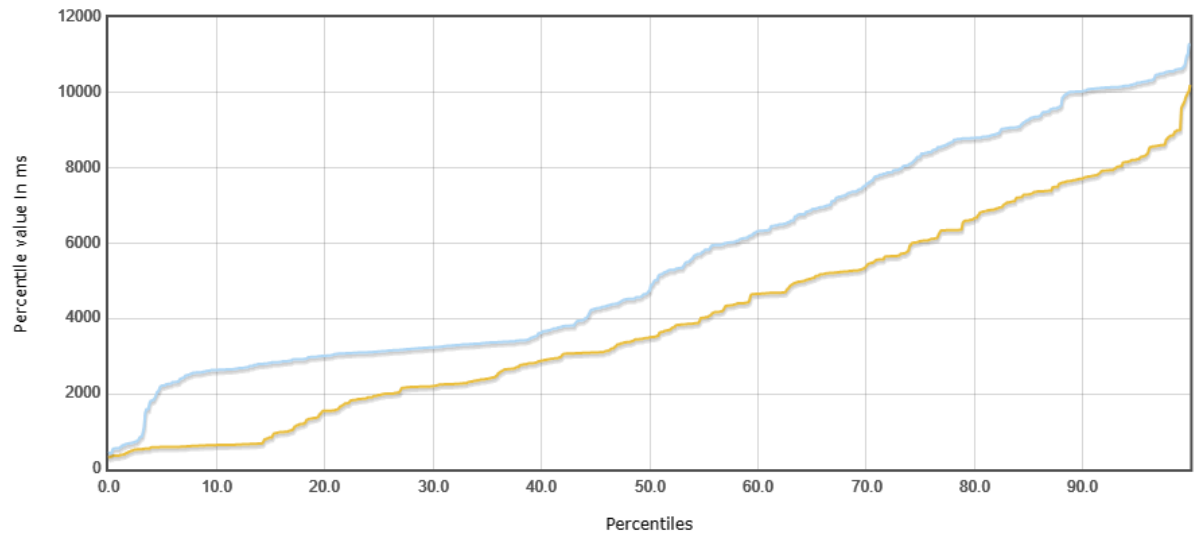
 Response Time Distribution



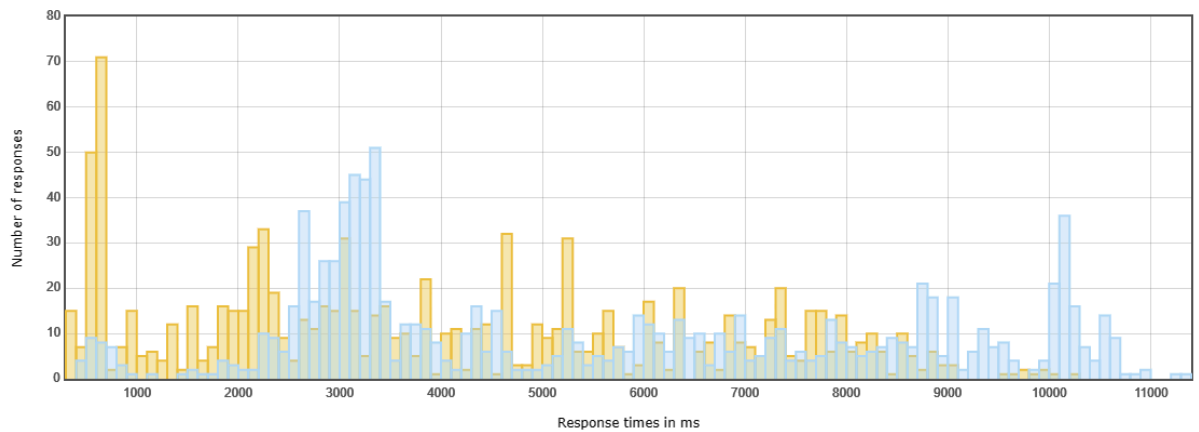
- Prosečena response time je 800ms

- 1000 korisnika (10.000 pokrenutih simulacija) - 16s, svi su prošli

Response Time Percentiles



Response Time Distribution



- Prosečena response time je 8s



## #5 Upali Lampu

### Scenario

- Korisnik se prijavi i upali lampu.

### Rezultati

- Ovaj test scenario je identičan kao i scenario #4 (promena jednog parametra unutar jednog modula)
- Pojavio se isti problem i rezultati su bili gotovo identični (unutar marginalne greške)

## #6 Dobavi istoriju prisutnosti automobila ispred kapije

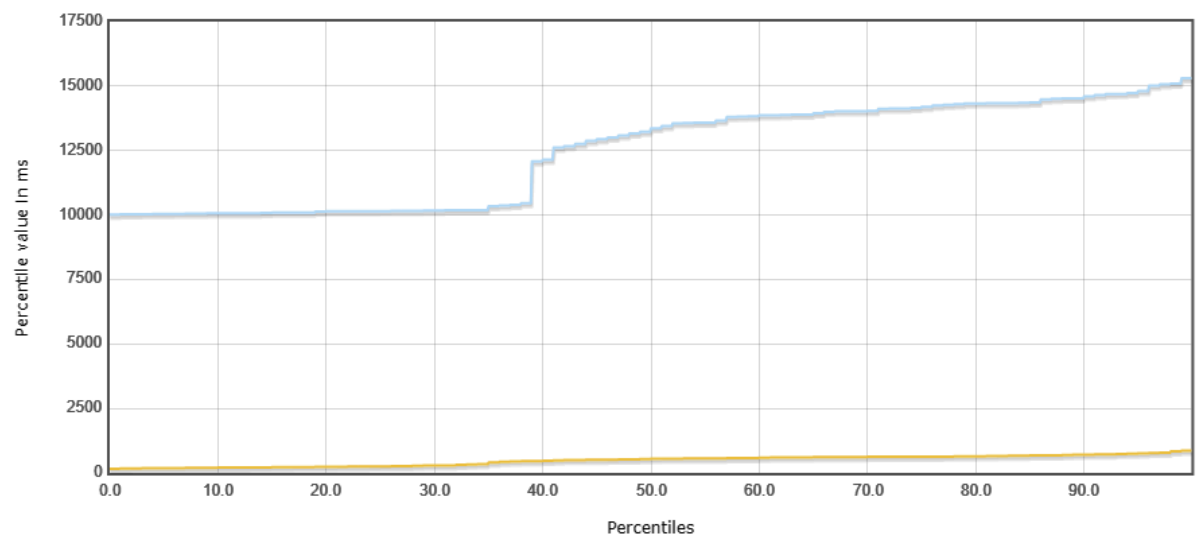
### Scenario

- Korisnik se prijavi i zahteva jednu stranicu tabele koja prikazuje koja vozila su bila pred kapijom.

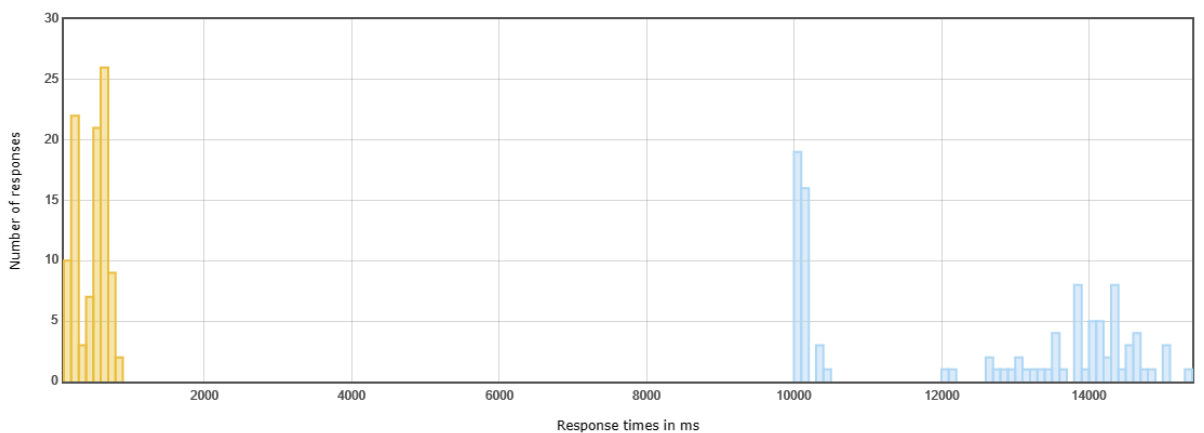
### Rezultati

- U ovom test scenariju se jasno vidi da je potraga podataka iz Influx baze vrlo zahtevna operacija te da su ti zahtevi mnogo duže cekali, čak i pri malom load-u sistema. Takođe, neki od njih su i odbijeni jer je influx read pucao (zbog prevelikog opterećenja)
- 100 korisnika (100 pokrenutih simulacija) - 15s, 61% prošlo

Response Time Percentiles

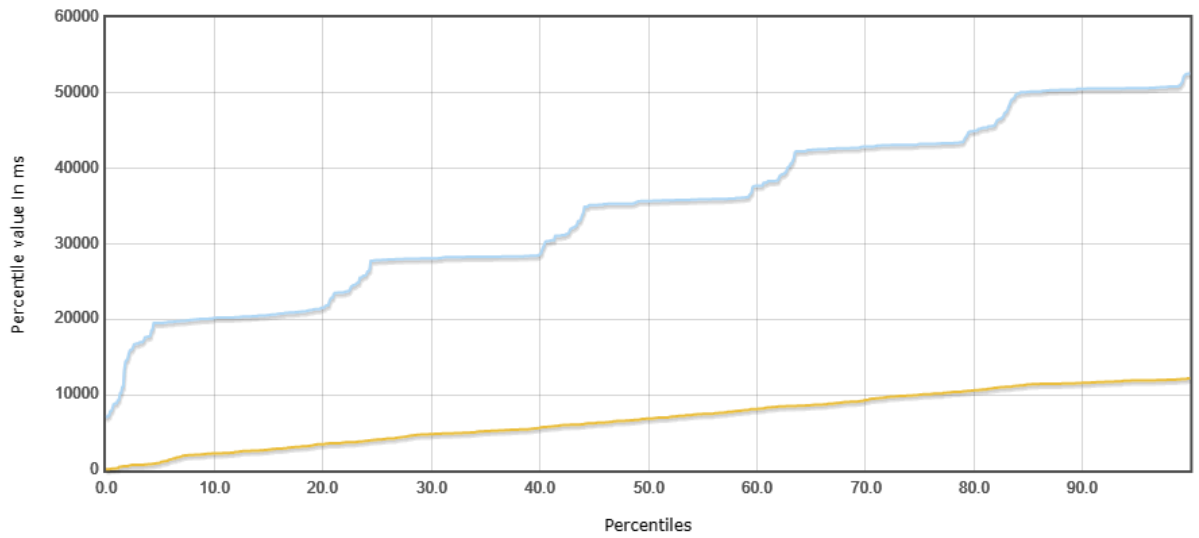


Response Time Distribution

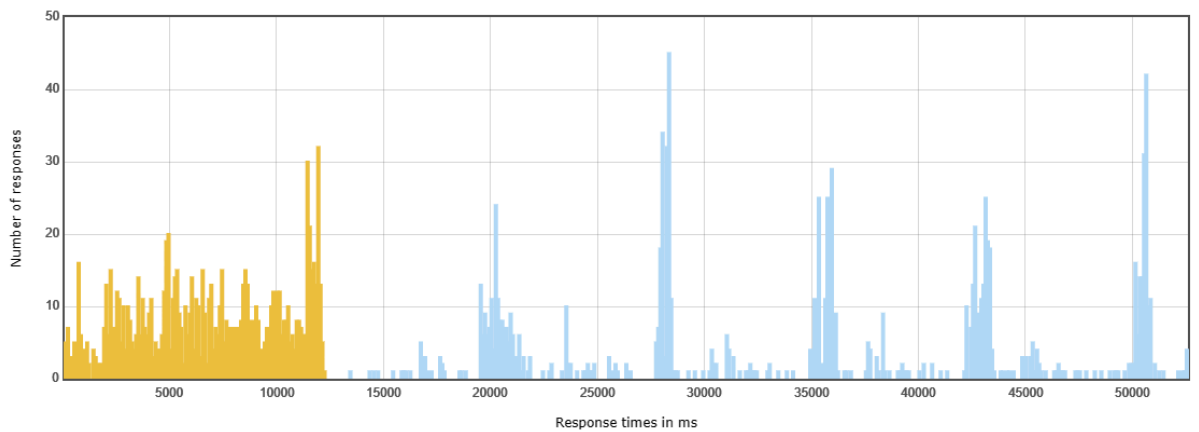


- 1000 korisnika (100 pokrenutih simulacija) - 66s, 2.5% prošlo

Response Time Percentiles



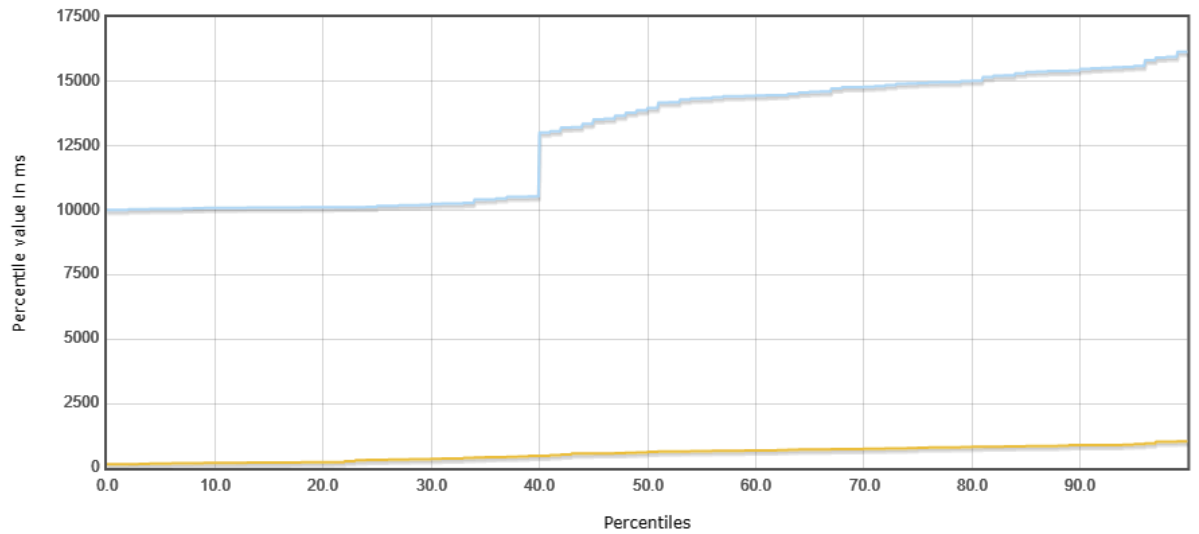
Response Time Distribution



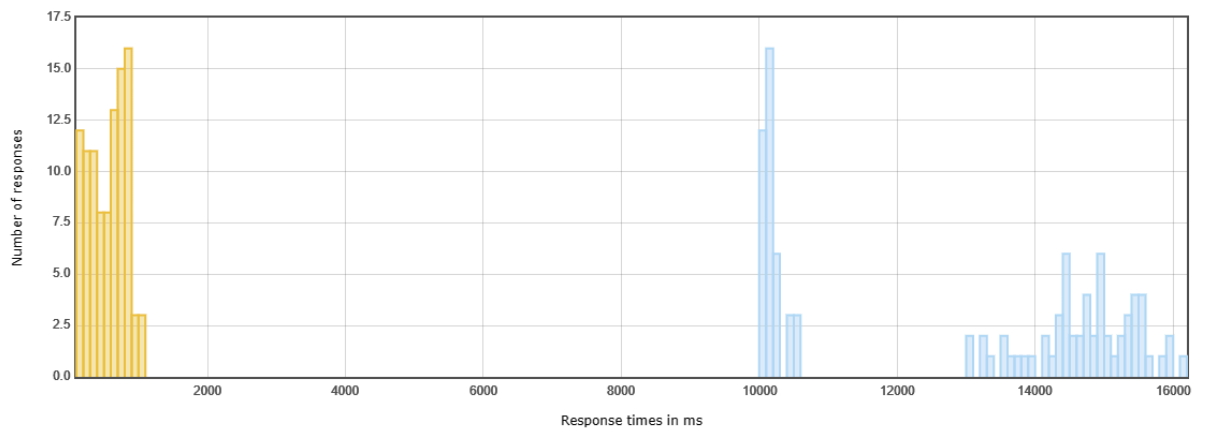
- Ovde vidimo jasne batch-eve u kojima su podaci obrađivani ali influx je i dalje pravio problem i zbog velike količine requestova nije dao mnogo odgovora

- 100 korisnika (1000 pokrenutih simulacija) - 16s, 60% prošlo

**Response Time Percentiles**



**Response Time Distribution**



- Performanse su gotovo identične kao kod slučaja sa 100 simulatora

- Na 10.000 simulatora influx nije uspeo da vrati podatke ni jednom korisniku jer je previše bilo istovremenih upisa.

## #7 Dobavi akcije paljenja/gašenja lampe

### Scenario

- Korisnik se prijavi i zahteva jednu stranicu tabele sa akcijama paljenja i gašenja lampe.

### Rezultati

- Kako se ovde radi o istom endpointu kao i u (#6), rezultati su bili dovoljno slični te ih nisam navodio

## #8 Dobavi istoriju osvetljenosti

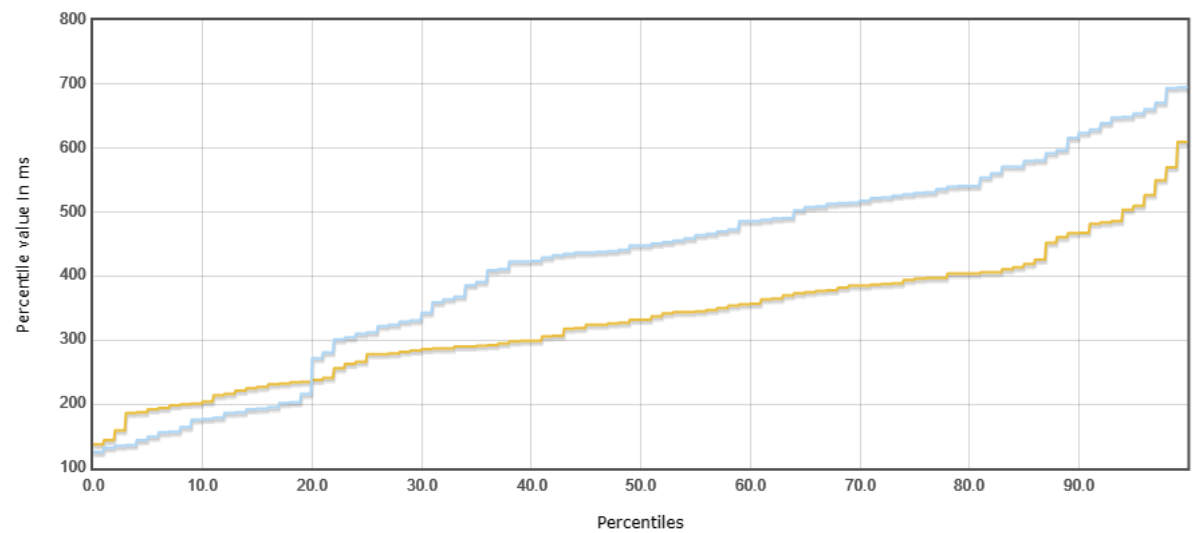
### Scenario

- Korisnik se prijavi i zahteva podatke za graf osvetljenosti.

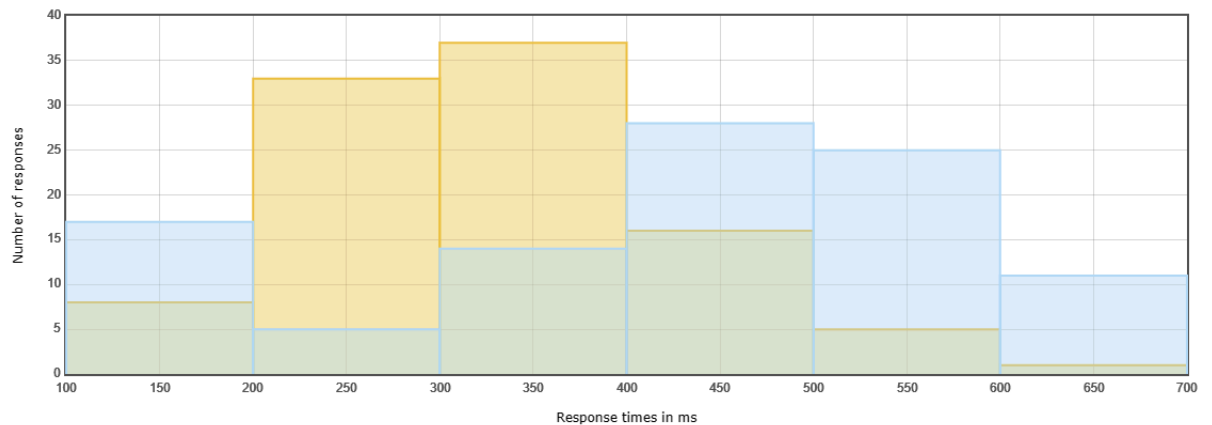
### Rezultati

- 100 korisnika (100 pokrenutih simulacija) - 2s, 100% prošlo

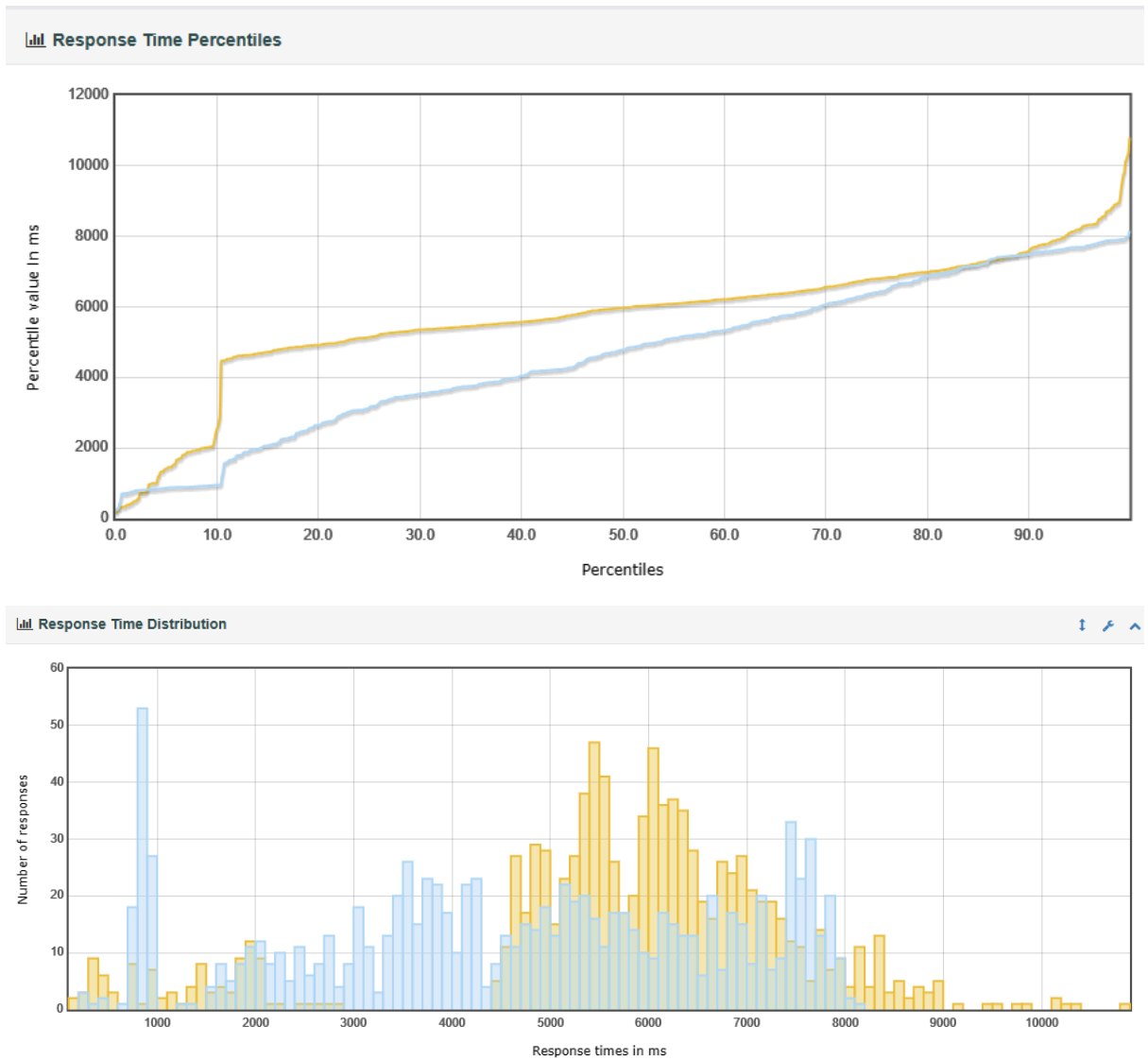
Response Time Percentiles



Response Time Distribution



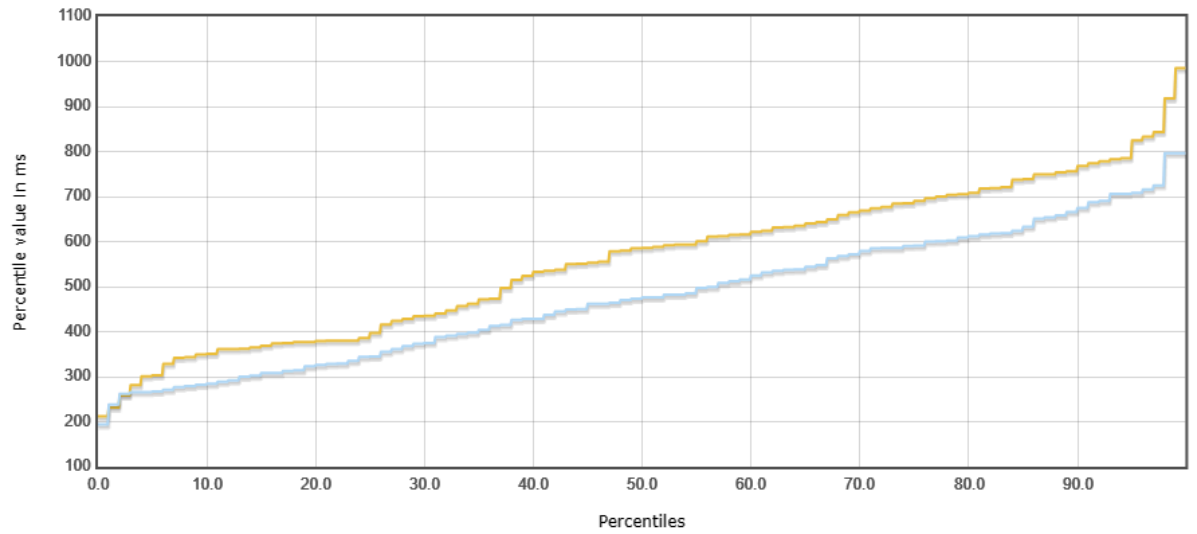
- 1000 korisnika (100 pokrenutih simulacija) - 15s, 100% prošlo



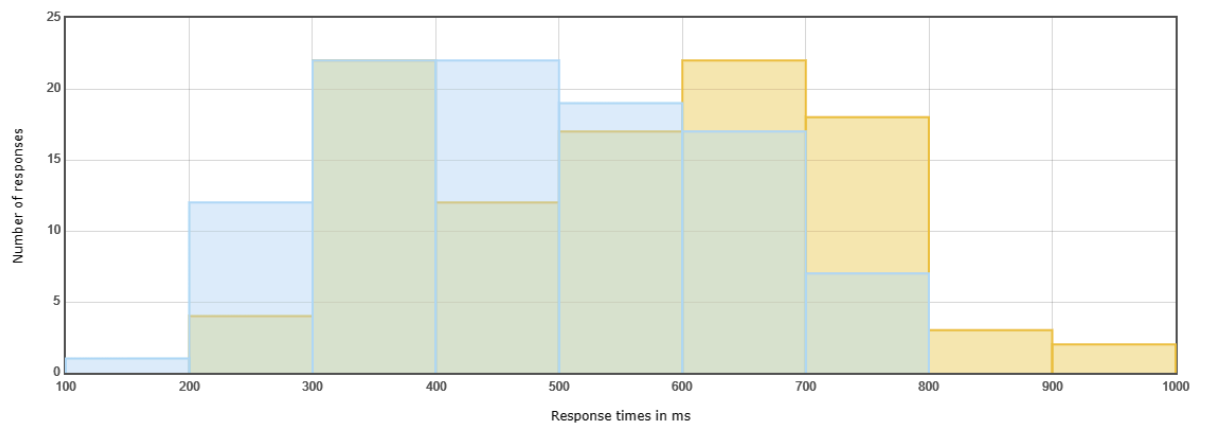
- Response time je sada 5s u proseku

- 100 korisnika (1000 pokrenutih simulacija) - 2s, 100% prošlo

**Response Time Percentiles**



**Response Time Distribution**

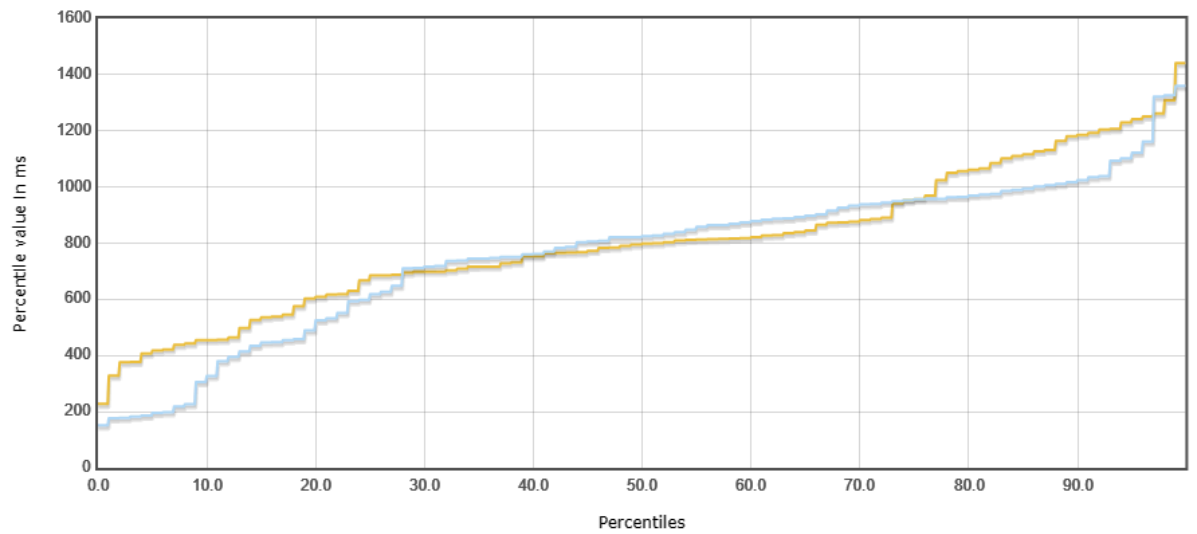


- Response time je prilično sličan slučaju kada imamo isti broj korisnika a 900 simuliranih uređaja manje

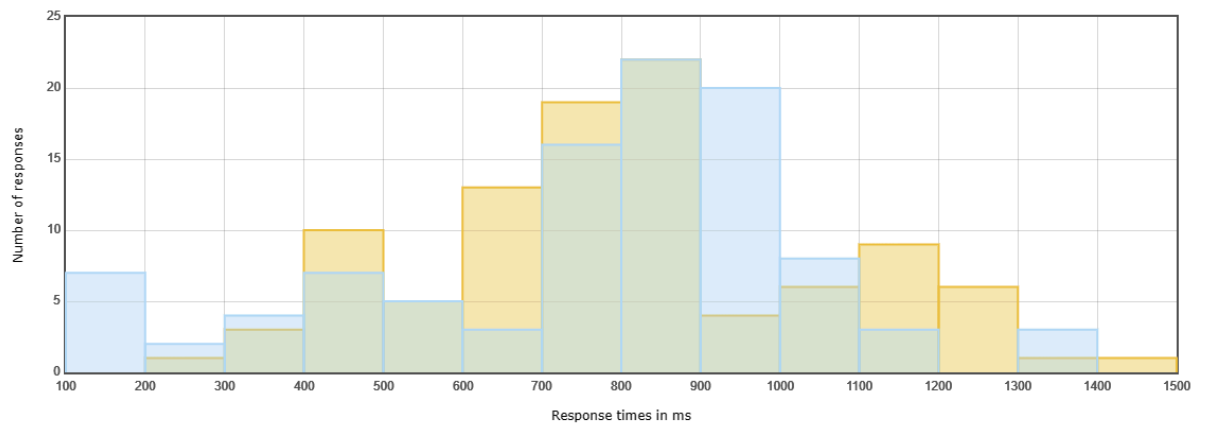


- 100 korisnika (10.000 pokrenutih simulacija) - 3s, 100% prošlo

Response Time Percentiles



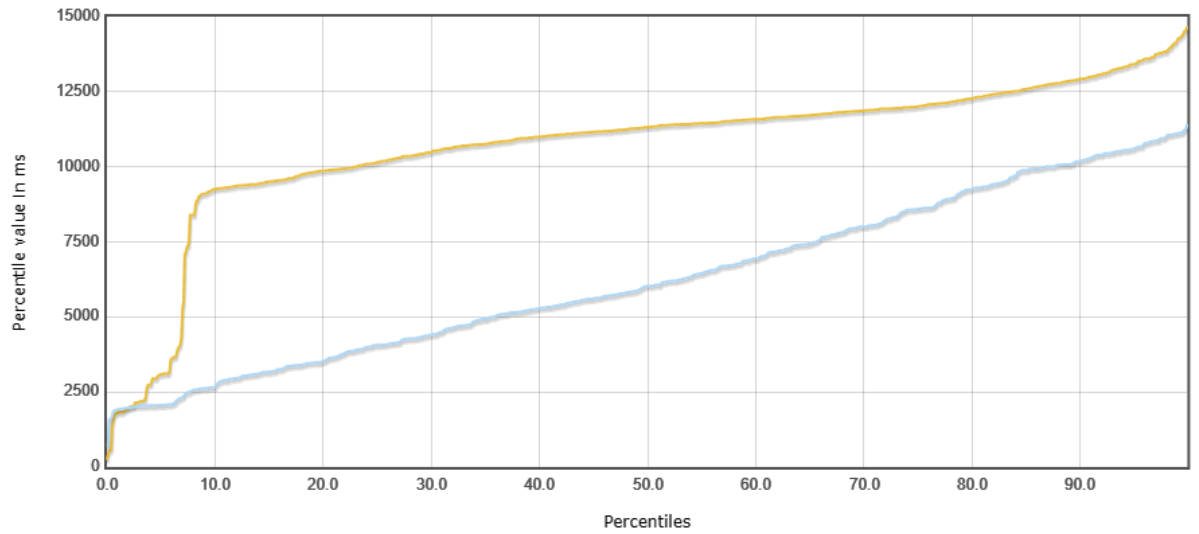
Response Time Distribution



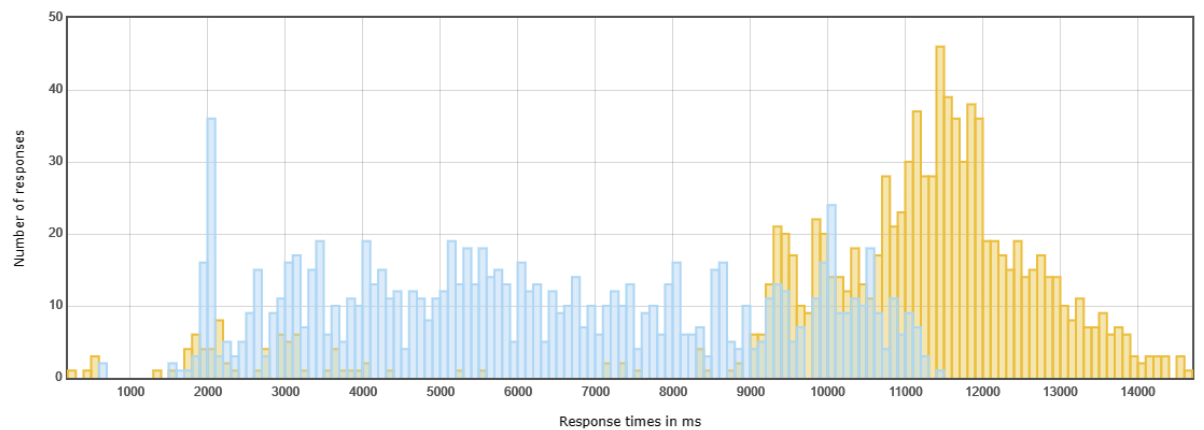
- Primetimo da je rezultat i dalje prilično dobar iako sada simuliramo 10.000 uređaja

- 1000 korisnika (10.000 pokrenutih simulacija) - 26s, 100% prošlo

Response Time Percentiles



Response Time Distribution



- Prosečan response time je oko 10s sto je prihvatljivo s obzirom na load pod kojim se sistem nalazi

## #9 Dobavi istoriju dostupnosti lampe

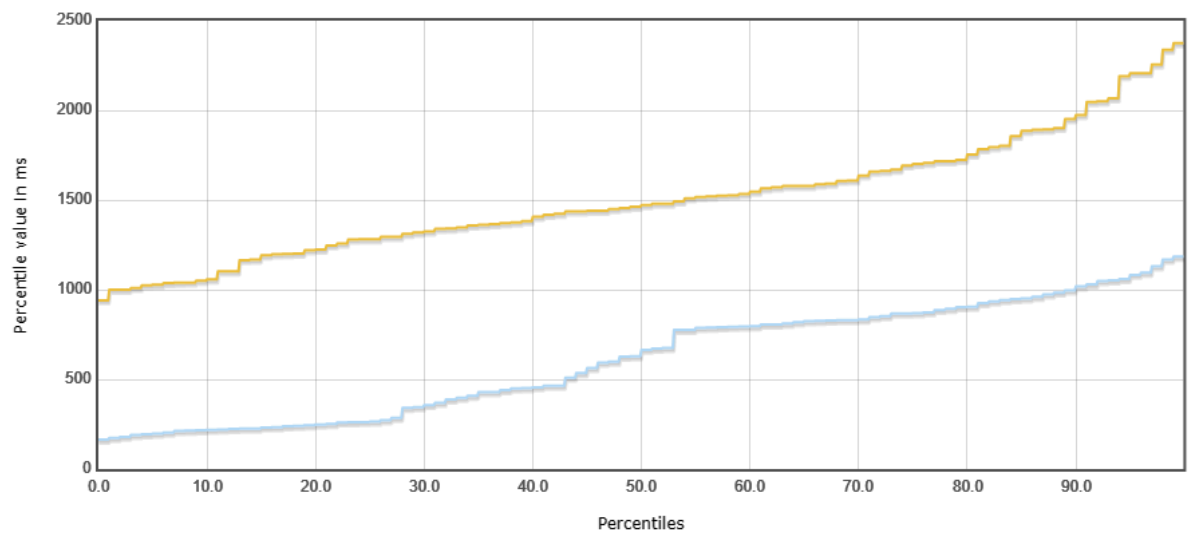
### Scenario

- Korisnik se prijavi i zahteva poslednjih mesec dana informacija o dostupnosti lampe.

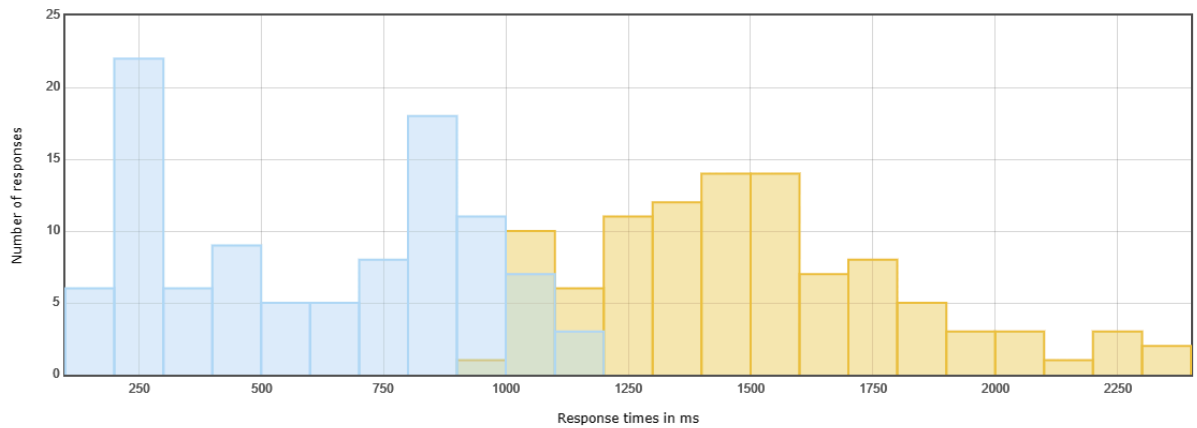
### Rezultati

- 100 korisnika (100 pokrenutih simulacija) - 3s, 100% prošlo

Response Time Percentiles

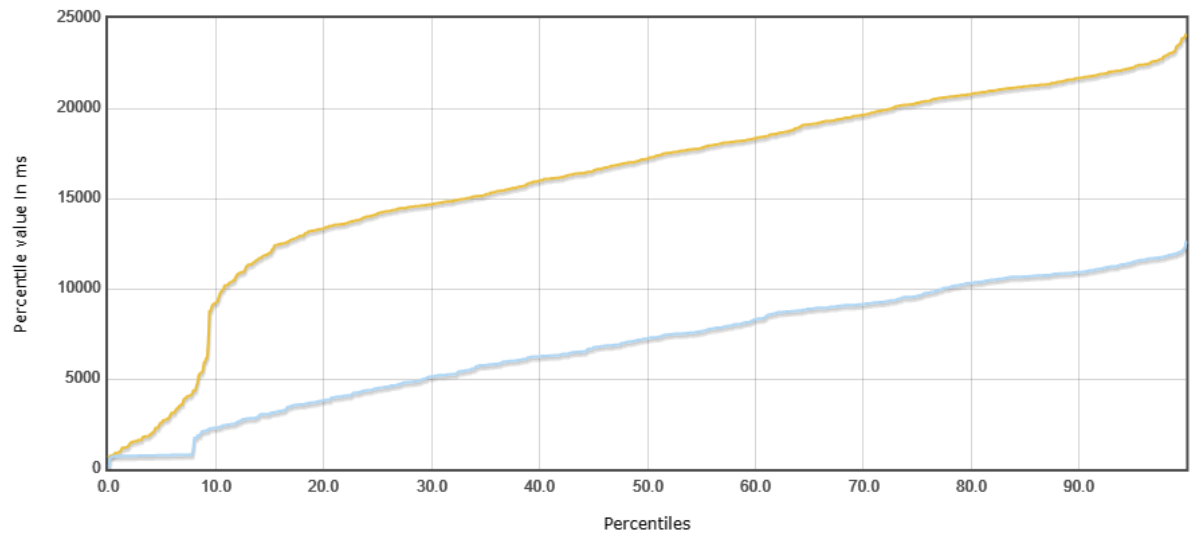


Response Time Distribution

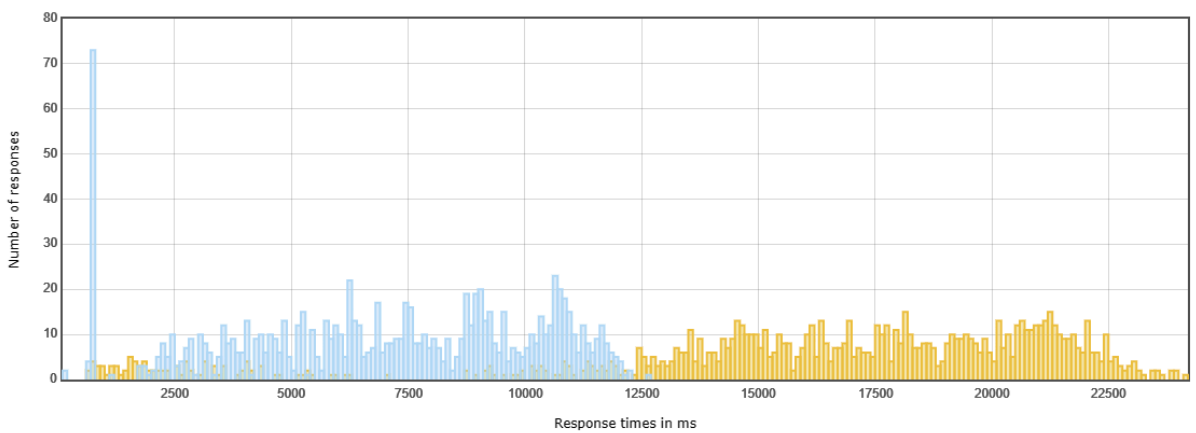


- 1000 korisnika (100 pokrenutih simulacija) - 33s, 97.2% prošlo

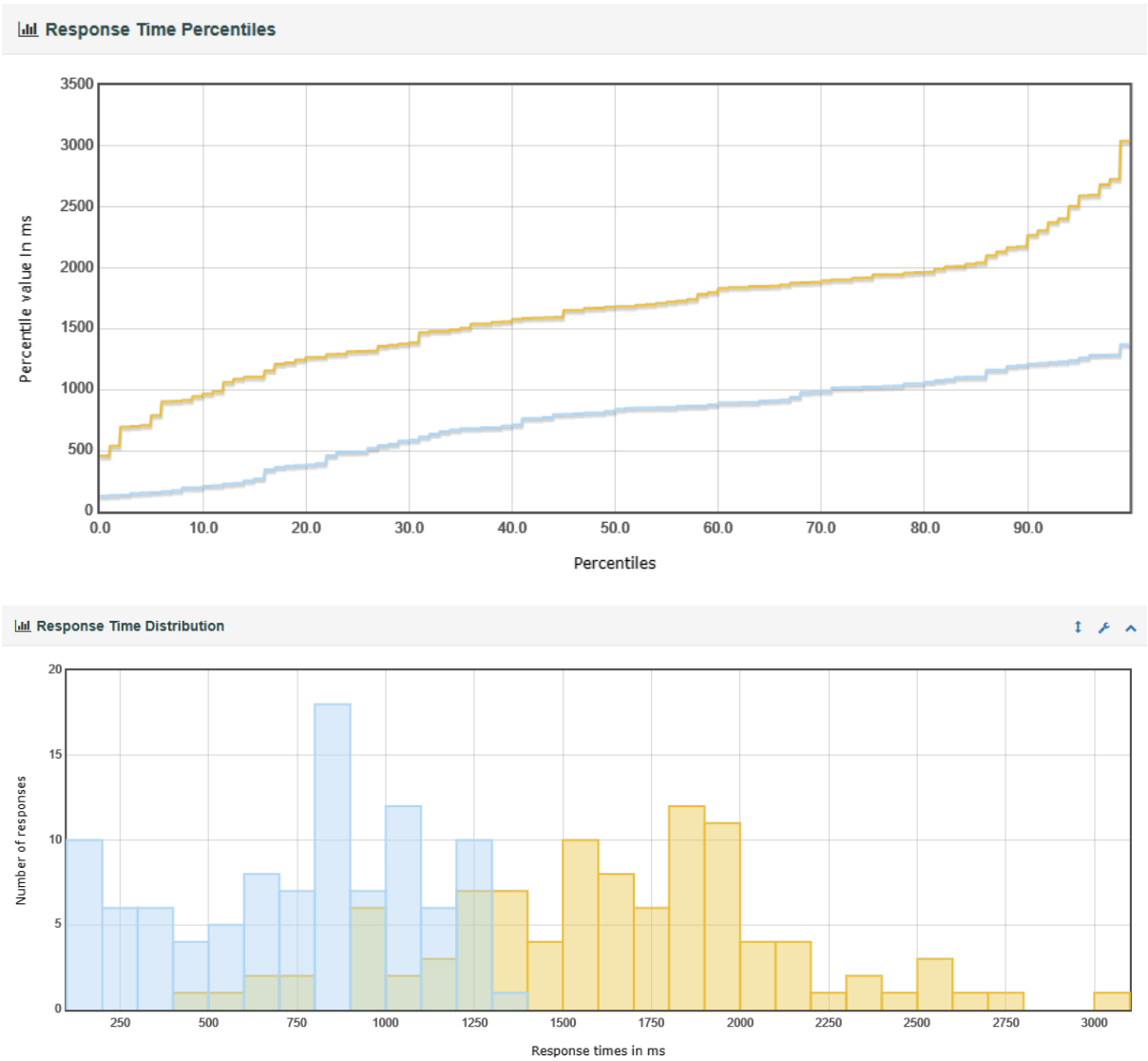
 Response Time Percentiles



 Response Time Distribution

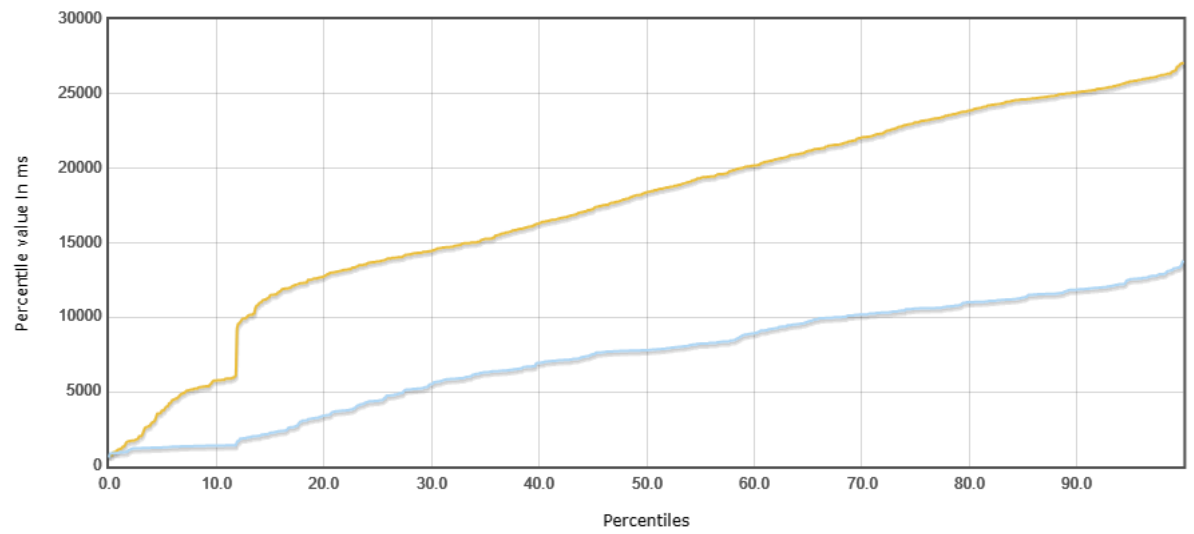


- 100 korisnika (1000 pokrenutih simulacija) - 3s, 100% prošlo

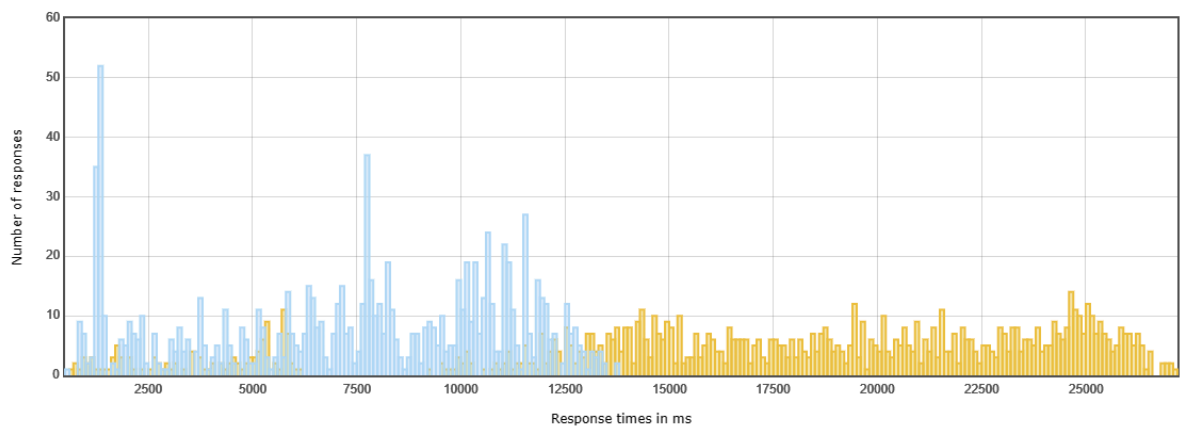


- 1000 korisnika (1000 pokrenutih simulacija) - 39s, 96.7% prošlo

Response Time Percentiles

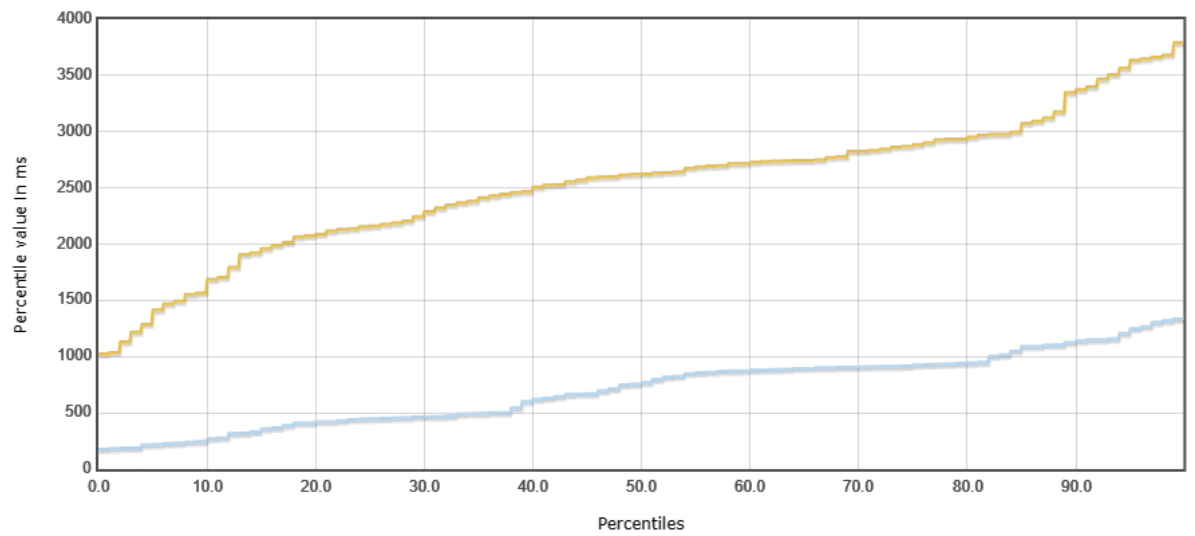


Response Time Distribution

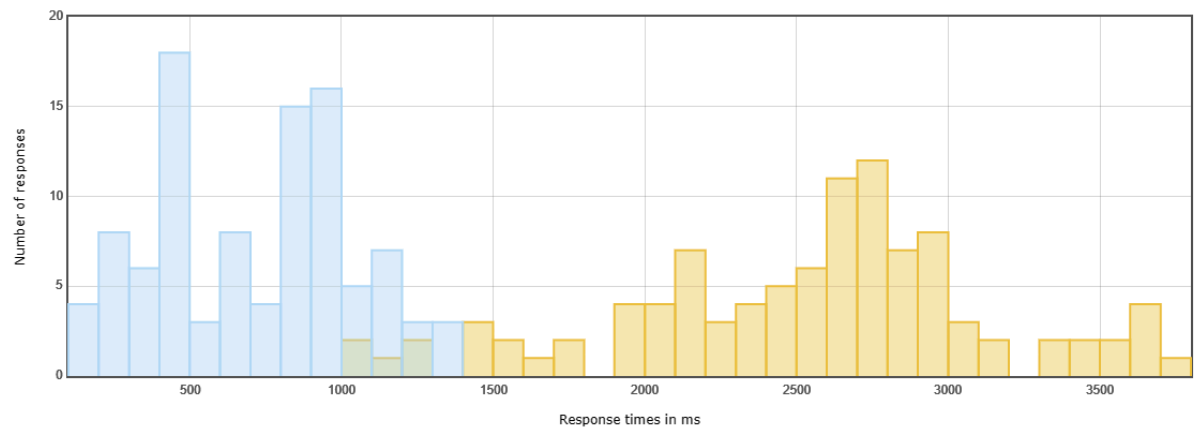


- 100 korisnika (10.000 pokrenutih simulacija) - 3s, 100% prošlo

Response Time Percentiles

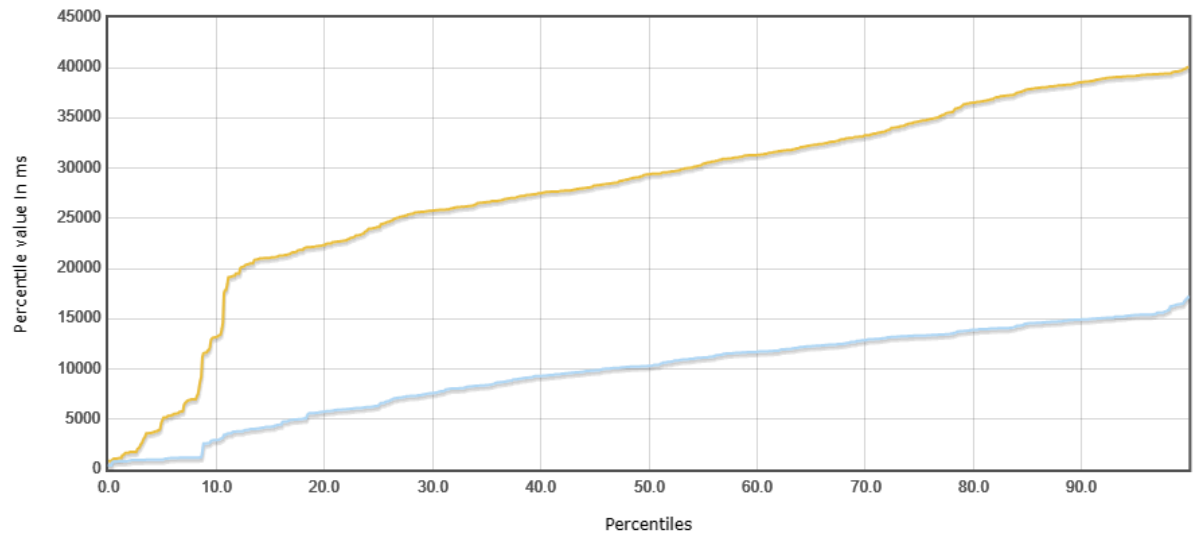


Response Time Distribution

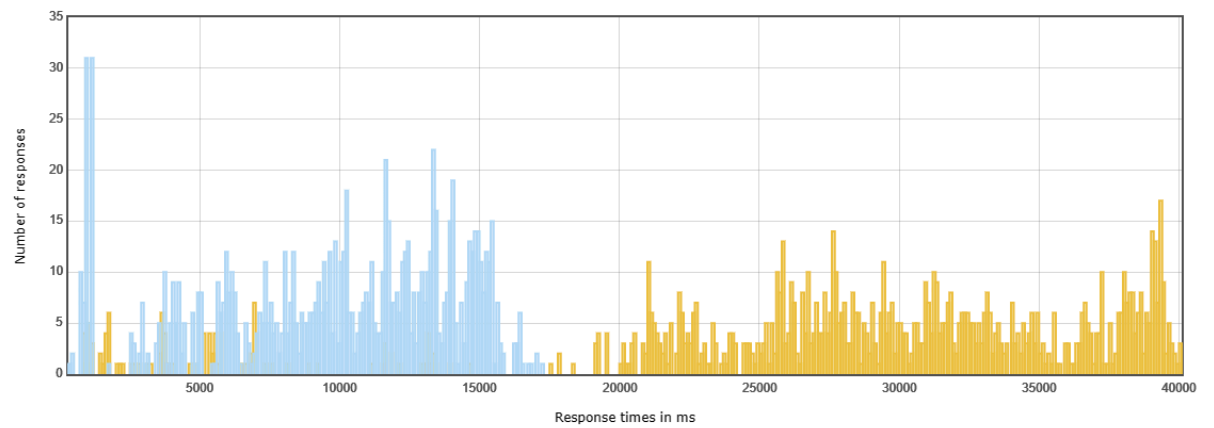


- 1000 korisnika (10.000 pokrenutih simulacija) - 57s, 63.5% prošlo

Response Time Percentiles



Response Time Distribution





## #10 Dobavi istoriju dostupnosti prskalica

### Scenario

- Korisnik se prijavi i zahteva poslednjih mesec dana informacija o dostupnosti prksalica.

### Rezultati

- Rezultati su gotovo identični kao i scenario (#9) jer se testira ista funkcionalnost za drugi uređaj

## Testiranje Performansi Simulatora

Testovi su izvedeni tako što se pokrene server i određeni broj simulatora. Za svaku mqtt poruku koju simulator pošalje pamti se vreme slanja. Kada server obradi poruku pamti se vreme završetka obrade i meri se prosečna razlika ova dva vremena za svaki tip mqtt poruke.

## Lampa

### Sve funkcionalnosti lampe su jednako responsive

- 10 uređaja - response time je ~5 sekundi
- 100 uređaja - response time je ~5 sekundi
- 200 uređaja - response time je ~5.5 sekundi
- 500 uređaja - response time je ~6 sekundi
- 1000 uređaja - response time je >70 sekundi

## Kapija

Simulacija kapije se sa pokretanjem većeg broja uređaja usporila iz prostog razloga sto je sada veliki broj simulator thread-ova na mali broj fizičkih thread-ova pa mnogo duže čekaju da odrade posao.

Akcije poput open/close ili private/public, koje ne zahtevaju odgovor simulatora, već se obavljaju na backend delu ostaju gotovo jednako responsive i na 500 uređaja. Konkretno na 10 uređaja, izmena se desi za manje od 0.5s dok se na 500 uređaja odvija za manje od 1.5s.

*(Merenje je vršeno od izvršenja akcije do pojave akcije u live tabeli)*

## Prskalice

Jedna od funkcionalnosti je dodavanje schedule-a i kao kod kapije ova funkcionalnost se odvija na backend delu tako da ostaje prilično responsive.

Druga opcija je paljenje i gašenje koje zahteva potvrdu simulatora te se sa pokretanjem 500 uređaja response time diže na preko 5 sekundi dok je inače response time ispod 1 sekunde.

## Ostatak sistema

Sto se tiče ostalih funkcionalnosti sistema one se mapiraju na performanse prikazane u load testiranju. Npr dobavljanje raznih grafova postaje nešto sporije ali i dalje dovoljno responsivo da bi korisnik bio zadovoljan (Ovo je pod uslovom da ne simuliramo više od 500 uređaja). Na 1000 i više simulacija sistem postaje dosta sporiji ali i dalje responsivan dok tek na 10.000 uređaja počinje da obija zahteve i pokazuje ne očekivano ponašanje.

## Poboljšanje sistema

- Jedan od problema na koje smo naleteli jeste da MQTT broker nije bio u stanju da odreaguje na veliki broj poruka tako da bismo mogli probati promeniti config ili poraditi na samom MQTT brokeru (Trenutno je pokrenut u dockeru pored svega ostalog na laptopu tako da radi sa dosta malo resursa)  
*(max\_inflight\_messages 10000 - linija u mqtt configu koja je popravila stanje vezano za slanje velikog broja poruka)*
- Takođe, radili smo provere na nekim mestima kako ne bismo radili istu akciju više puta.
- Logovi dosta utiču na performanse, ali su oni izuzetno bitni tako da smo njih zadržali