

Izveštaj testiranja performansi sistema

Projekat iz Naprednih veb tehnologija

Autor: Katarina Spremić SV36/2020

Fakultet Tehničkih Nauka, Novi Sad

Zimski semestar, školska 2023/24

Uvod

Testiranje sistema aplikacije za rukovanje pametnim uređajima unutar nekretnina.

Testiranje sistema izvedeno je uz pomoć alata Locust.

Performanse se gledaju kroz nekoliko metrika za različite scenarije korišćenja aplikacije, gde se svaki sastoji od prijavljivanja korisnika na sistem (login) i izvršavanje zahteva.

Izabrani scenariji vezani su za zadatke studenta 2.

Specifikacije sistema

Backend: REST u Java Spring tehnologiji

SQL baza: Postgres

Time series baza: InfluxDB

Uređaji (simulatori) su implementirani u Go programskom jeziku i komunikaciju ostvaruju putem MQTT protokola

Specifikacije mašine na kojoj se izvode testovi

16GB RAM

Ryzen 7 procesor

Testiranje opterećenja sistema za česte scenarije korišćenja

Za sve test scenarije se koristi wait time od 1-2 sekunde.

Test scenario 1: **Login + Pravljenje nekretnine**

REST api se gađa kroz endpointe:

- POST /api/**login**
- POST /api/**property**
- POST /api/**images/property/upload**

Broj različitih korisnika u bazi za koje se vrši testiranje: 10. Svaki scenario poziva nasumično jedan od tih korisnika. Broj nekretnina po korisniku je ravnomerno raspoređen.

Početni broj nekretnina	Broj korisnika kroz locust	Spawn rate	Koliko dugo se izvršavao test	Prosečan response time za sve zahteve iz scenarija (ms)	Prosečan RPS	Ukupan broj zahteva	Uspešnost	Broj nekretnina na kraju
10	10	1	60 sekundi	28	18.6	1104	100%	400
440	100	5	60 sekundi	50	185	9297	100%	3536
3500	100	5	120 sekundi	37	184	18974	100%	8932
9000	100	5	60 sekundi	185	168.4	10154	100%	12090

Test scenario 2: Login + Prihvatanje nekretnine

REST api se gađa kroz endpointe:

- POST /api/**login**
- POST /api/property/pending/**accept**/{property-id}

Broj različitih korisnika u bazi za koje se vrši testiranje: 5. Korisnik ima tip naloga *admin*. Svaki scenario poziva nasumično jedan od tih korisnika.

Početni broj <i>pending</i> nekretnina	Broj korisnika kroz locust	Spawn rate	Koliko dugo se izvršavao test	Prosečan response time (ms)			Prosečan RPS	Ukupan broj zahteva	Uspešnost
				Login	Accept	Scenario			
9300	10	1	60 sekundi	261	354	615	13	708	100%
8000	50	5	60 sekundi	261	278	539	65	5671	100%
8000	100	5	60 sekundi	261	146	407	126	9117	83%*

Komentari i zapažanja:

* U scenario su uključeni slučajevi kada korisnik pokuša da prihvati nekretninu koja je već prihvaćena, te se zbog odgovor na zahtev sa *status code*-om različitim od 200 smatra neuspešnim. Iz tog razloga deo zahteva nije stigao do servisnog sloja aplikacije, i response time je manji u proseku.

Test scenario 3: Login + Odbijanje nekretnine

REST api se gađa kroz endpointe:

- POST /api/**login**
- POST /api/property/pending/**reject**/{property-id}

Broj različitih korisnika u bazi za koje se vrši testiranje: 5. Korisnik ima tip naloga *admin*. Svaki scenario poziva nasumično jedan od tih korisnika.

Početni broj <i>pending</i> nekretnina	Broj korisnika kroz locust	Spawn rate	Koliko dugo se izvršavao test	Prosečan response time (ms)			Prosečan RPS	Ukupan broj zahteva	Uspešnost
				Login	Reject	Scenario			
9300	10	1	60 sekundi	261	314	575	13	708	100%
8000	50	5	60 sekundi	261	342	603	65	5671	100%
8000	100	5	60 sekundi	265	440	705	126	8843	100%

Komentari i zapažanja:

Kroz odbijanje nekretnine koristi se servis *SendGrid* za slanje email-ova korisnicima, te je zapažen veći response time zbog čekanja na odgovor servisa, dok je pored toga poslovna logika za ovaj zahtev veoma jednostavna.

Test scenario 4: **Login + Pregled nekretnina + Pregled uređaja za jednu nekretninu**

REST api se gađa kroz endpointe:

- POST /api/**login**
- GET /api/**property**
- GET /api/device/**appliances**/ {propertyId}
- GET /api/device/**outdoor**/ {propertyId}
- GET /api/device/**energyDevices**/ {propertyId}

Broj različitih korisnika u bazi za koje se vrši testiranje: 10. Svaki scenario poziva nasumično jedan od tih korisnika. Broj nekretnina po korisniku je ravnomerno raspoređen.

Broj nekretnina po korisniku	Broj korisnika kroz locust	Spawn rate	Koliko dugo se izvršavao test	Prosečan response time za zahtev (ms)			Prosečan RPS	Ukupan broj zahteva	Uspešnost
				Login	Property	Devices			
350	10	1	60 sekundi	63	17	17	13	1175	100%
350	100	5	60 sekundi	97	25	26	126	14940	100%
1200	100	5	60 sekundi	317	158	150	126	10106	100%

Komentari i zapažanja:

Prosečan *response time* za bilo koji zahtev iz scenarija raste sa brojem podataka i brojem korisnika koji upućuju zahteve. Sa rastom broja podataka i korisnika se za isto vreme obradi manje zahteva.

Grupa scenarija: **Login + Pregled dostupnosti uređaja za vremenski interval**

REST api se gađa kroz endpointe:

- POST /api/**login**
- GET /api/device/**onlinePercent**{deviceId}
- GET /api/device/**onlineIntervals**{deviceId}
- GET /api/device/**onlinePerUnit**{deviceId}

gde svi GET zahtevi imaju i query parametre za pocetak i kraj vremenskog intervala

Broj različitih korisnika u bazi za koje se vrši testiranje: 1.

Različiti endpoint-i su za različite agregacije podataka i računanje statistika. Postoji endpoint za ukupno procentualno stanje, endpoint za pokazivanje zavisnosti stanja i vremena, i endpoint za procentualno stanje po vremenskoj jedinici.

Influx baza podataka je popunjena sa podacima za prethodna 3 meseca. Point u bazi ima vrednost nula ili 1 za measurement “online/offline”, sto oznacava da je uređaj postao online ili offline. Vrednosti su rasporedjene sa vremenskim razmakom od 15 do 27 sati.

Scenario za vremenski interval	Broj korisnika kroz locust	Spawn rate	Koliko dugo se izvršavao test	Prosečan response time (ms)		Prosečan RPS	Ukupan broj zahteva	Uspešnost
				Za zahteve	Za scenario			
Prethodnih 6 sati	100	5	60 sekundi	43	168	250	12287	100%
Prethodna 24 sata	100	5	60 sekundi	51	189	241	11000	100%
Prethodnih nedelju dana	100	5	60 sekundi	81	305	201	10106	100%
Prethodnih nedelju dana (optimizovano)	100	5	60 sekundi	48	152	241	12588	100%
Prethodnih mesec dana	100	5	60 sekundi	50	312	199	10053	100%
Prethodnih mesec dana (optimizovano)	100	5	60 sekundi	33	165	245	12488	100%

Komentari i zapažanja:

Scenario za podatke za prethodnih nedelju dana i scenario za prethodnih mesec dana daju lošije performanse. Radi optimizacije, uvodi se *downsampling*, gde će se vrednosti agregirati u nekom vremenskom okviru. S obzirom na dužinu intervala u kom se traže podaci, tražiće se *mean* podataka u vremenskom okviru. Sa ovom metodom performanse traženja podataka za period od nedelju dana i mesec dana su izjednačene sa performansama za 6 sati i 24 sata.

Test scenario 9: Login + Pregled događaja kapije za mesec dana

REST api se gađa kroz endpointe:

- POST /api/login
- GET /api/gate/{device-id}/range

Broj različitih korisnika u bazi za koje se vrši testiranje: 1.

Ukupan broj događaja za uređaj	Broj događaja za mesec dana	Broj korisnika kroz locust	Spawn rate	Koliko dugo se izvršavao test	Prosečan response time za dobavljanje događaja (ms)	Prosečan RPS za dobavljanje događaja	Prosečna veličina u bajtovima
1	1	100	1	60 sekundi	10	60	328
12 000	360	100	5	60 sekundi	32	63	72331
12 000	360	1 000	500	60 sekundi	4980	95	72331
40 000	16000	100	5	60 sekundi	419	79	560360
40 000	16000	1 000	500	60 sekundi	10493	71	560360

Test scenario 10: Login + Pregled komandi prskalice za mesec dana

REST api se gađa kroz endpointe:

- POST /api/login
- GET /api/sprinkler/{device-id}/range

Broj različitih korisnika u bazi za koje se vrši testiranje: 1.

Ukupan broj događaja za uređaj	Broj događaja za mesec dana	Broj korisnika kroz locust	Spawn rate	Koliko dugo se izvršavao test	Prosečan response time za dobavljanje komandi (ms)	Prosečan RPS za dobavljanje komandi	Prosečna veličina u bajtovima
2200	800	100	5	60 sekundi	475	36	84354
4343	1463	100	5	60 sekundi	1633	22.3	165702
4343	1463	250	25	60 sekundi	5540	21.9	165702
4343	1463	1 000	100	60 sekundi	17295	22.1	165702

Komentari i zapažanja:

Scenario za podatke za za prethodnih mesec dana daje loše performanse za velik broj korisnika i veliku količinu podataka koja se dobavlja po zahtevu. Poželjno bi bilo optimizovati rad uvođenjem paginacije, da se dobavi odabran deo podataka.

Testiranje performansi sistema usled povećanja broja uređaja (simulatora) koji komuniciraju sa platformom

Generalni opis funkcionalnosti

U nastavku sledi opis toka komunikacije u okviru funkcionalnosti uređaja. Web klijent šalje HTTP zahtev serveru, koji dalje komunicira sa uređajem putem MQTT protokola. Uređaj na odgovarajućim kanalima sluša poruke i izvršava dobijenu komandu. Rezultat izvršenja uređaji šalju na odgovarajuće kanale na kojima server sluša poruke. Server čuva rezultate i ažurirane podatke šalje web klijentu putem web socket protokola.

Slanje izmerenih vrednosti sa senzora se obavlja putem MQTT protokola uz pomoć alata Telegraf, koji sluša poruke na odgovarajućim kanalima i upisuje merenja u Influx bazu podataka.

Test scenario 1: Čuvanje vrednosti senzor osvetljenja lampe

Potencijalno usko grlo ovog scenarija je opterećenje kanala za komunikaciju kojima vrednosti stižu preko Telegrafa do Influx-a. Ispitaće se i performanse dobavljanja vrednosti senzora za prethodnih mesec dana.

Test je sproveden kroz pokretanje većeg broja instanci uređaja kroz skriptu, koji šalju vrednosti senzora, i pokretanje jedne instance uređaja umnoženu kroz Locust koja će slati zahtev za dobavljanje vrednosti senzora. Ta dva uređaja su različiti u bazi. Uređaj 1, za kog se dobavljaju vrednosti senzora ima vrednosti u Influx bazi za poslednja 3 meseca. Posmatra se response time za dobavljanje vrednosti senzora uređaja 1, usled otprećenja zbog upisa vrednosti senzora uređaja 2.

Broj korisnika kroz Locust za uređaj 1	Broj instanci kroz skriptu za uređaj 2	Spawn rate	Koliko dugo se izvršavao test	Prosečan response time za dobavljanje vrednosti senzora (ms)	Prosečan RPS za dobavljanje vrednosti senzora
10	10	5	60 sekundi	23	6
25	25	5	60 sekundi	20	16.2
100	50	25	60 sekundi	28	65.3
75	75	25	60 sekundi	28	50.3

Komentari i zapažanja:

Zbog ograničenja mašine na kojoj se vrši testiranje, moguće je bilo pokrenuti maksimalno do 75 instanci uređaja. Nije zapaženo opadanje preformansi dobavljanja vrednosti senzora za uređaj 1. Preko web klijenta ispitano je dobavljanje vrednosti uređaja 2 i zapaženo je da se vrednosti za grafik prikazivanja metrike senzora uspešno dobavljaju u zadovoljavajućem vremenskom periodu, međutim onemogućeno je menjanje sijalice i automatskog režima usled višestrukog pokretanja simulatora za uređaj i preplitanja izvršavanja komandi. Web socketi su uspešno radili.

Grupa scenarija: Uključivanje i isključivanje sijalice lampe i prskalice, uključivanje automatskog režima lampe i kapije

S obzirom na tok komunikacije, uređaji nezavisno jedan od drugog komuniciraju sa platformom i usko grlo bi moglo da bude izazvano opterećenjem servera, da response time za zahteve koji klijent upućuje bude duže, i da komunikacija kroz MQTT bude sporija jer se šalje veliki broj poruka.

U prvom scenariju ustanovljeno je da je onemogućeno menjanje sijalice i automatskog režima za uređaj ukoliko je višestruko pokrenut simulatora za isti uređaj.

Performanse će se testirati izvršavanjem scenarija za jedan uređaj iz grupe spoljašnjih pametnih uređaja preko web klijenta, uz dodatno pokretanje više simulatora za svaki tip uređaja.

Broj instanci po tipu uređaja iz grupe SPA kroz skriptu	Ukupan broj instanci
10	30
30	90
50	150

Komentari i zapažanja:

Sa do 75 instanci uređaja dostignut je maksimum kapaciteta mašine. Nije zapaženo opadanje preformansi funkcionalnosti za uređaj dok su pokrenute instance.

Test scenario 7: Događaj na kapiji

Performanse će se testirati izvršavanjem scenarija za jedan uređaj iz grupe spoljašnjih pametnih uređaja preko web klijenta, uz dodatno pokretanje više simulatora za svaki tip uređaja.

Sve vrednosti slate sa simuliranih senzora pokrenutih u više instanci stizale su na upis u Influx. Web socketi su radili ispravno i obavestavali o događajima na kapiji za uređaj koji je višestruko pokrenut. Uređaj koji ima jednu instancu takođe je radio ispravno, i što se tiče pristizanja vrednosti i ažuriranja web klijenta kroz web socket-e.

Broj instanci po tipu uređaja iz grupe SPA kroz skriptu	Ukupan broj instanci
10	30
30	90
50	150

Test scenario 8: **Postavljanje rasporeda za prskalicu**

Performanse će se testirati izvršavanjem scenarija za jedan uređaj iz grupe spoljašnjih pametnih uređaja preko web klijenta, uz dodatno pokretanje više simulatora za svaki tip uređaja.

Sve vrednosti slate sa simuliranih senzora pokrenutih u više instanci stizale su na upis u Influx. Web socketi su radili ispravno i obavestavali o događajima na kapiji za uređaj koji je višestruko pokrenut. Uređaj koji ima jednu instancu takođe je radio ispravno, i što se tiče pristizanja vrednosti i ažuriranja web klijenta kroz web socket-e.

Broj instanci po tipu uređaja iz grupe SPA kroz skriptu	Ukupan broj instanci
10	30
30	90
50	150