Izveštaj testiranja performansi sistema

Projekat iz Naprednih veb tehnologija

Autor: Selena Milutin SV39/2020

Fakultet Tehničkih Nauka, Novi Sad

Zimski semestar, školska 2023/24

Uvod

Testiranje performansi ispituje da li i kako radi svaki deo aplikacije kao jedinstven i kao integrisan deo sistema. U daljem tekstu su dati izveštaji performasi sistema pod različitim slučajevima korišćenja i u različitim uslovima. Rađeno je testiranje opterećenja sa simuliranih 50, 150 i 300 korisnika, nad bazom sa manje i više podataka, kao i testiranje performansi sistema usled povećanja broja uređaja (simulatora) koji komuniciraju sa platformom. Za testiranje je korišćen alat Locust koji prilikom pokretanja nudi mogućnost biranja broja korisnika i brzinu njihovog kreiranja. Za svrhe ovog testiranja brzina kreiranja korisnika je stavljena na 5.

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 (*load testing*) za česte scenarije korišćenja

Za ovu grupu testinja su ispitani sledeći scenariji:

1. Jednostavna registracija uređaja
   Podrazumevani koraci za ovaj scenario su prijava na sistem, dobavljanje svih nekretnina i slanje zavhteva za kreiranje uređaja

   REST api se gađa kroz endpointe:
   - POST /api/login
   - GET /api/property
   - POST /api/device
   - POST /api/images/device/upload

Br uređaja na početku	Br korisnika	Br zahteva	Mediana(ms)	Br zahteva po sek	Br uređaja na kraju
11	50	405	124	28	146
11	150	2746	178	105	917
11	300	6558	593	130	2294
3810	50	429	47	33	8953
3810	150	2616	77	119	4683
3810	300	7111	464	146	6189

2. Registracija uređaja preko nekretnine
   Podrazumevani koraci za ovaj scenario su prijava na sistem, dobavljanje svih nekretnina, dobavljanje nekretnine čiji uređaj želimo da dodamo i slanje zavhteva za kreiranje uređaja

   REST api se gađa kroz endpointe:
   - POST /api/login
   - GET /api/property
   - GET /api/property/id
   - POST /api/device
   - POST /api/images/device/upload

Br uređaja na početku	Br korisnika	Br zahteva	Mediana(ms)	Br zahteva po sek	Br uređaja na kraju
11	50	445	135	29	149
11	150	2656	197	114	927
11	300	6838	603	117	2284
3810	50	409	66	32	8963
3810	150	2735	112	134	4688
3810	300	7322	473	152	6151

3. Tabelarni izveštaj za admina

   Podrazumevani koraci za ovaj scenario su prijava na sistem, dobavljanje svih nekretnina zajedno sa akumuliranim podacima za potrošnju i proizvodnju energije i dobavljanje svih gradova zajedno sa akumuliranim podacima za potrošnju i proizvodnju energije

   REST api se gađa kroz endpointe:
   - POST /api/login
   - GET /api/property/accepted?start=x?end=y
   - GET /api/property/byCity?start=x?end=y

Vremenski interval (dani)	Br korisnika	Br zahteva	Mediana(ms)	Br zahteva po sek
2	50	329	277	29
2	150	1123	1192	37
2	300	2083	2993	37
7	50	283	349	24
7	150	1086	1461	36
7	300	2252	1870	37

   Komentar:
   Vidimo da se rezultati ne razlikuju značajno u odnosu na vremenski period. Razlog je smeštanje logike dobijanja agregiranog rezultata za proizvodnju i potrošnju u okviru flux query-a. Primetne razlike u performasi su bile kada je mnogo nekretnina u bazi, tada se na dobavljanje podataka čekalo do 20 sekundi za 2000 nekretnina.

4. Izveštaj za nekretninu za admin

   Početni koraci za ovaj scenario su isti kao i za tabelarni prikaz za admina odakle dobija podatke za potrošnju i proizvodnju energije za zadati vremeski period za mesečni i nedeljni izveštaj, kao i izveštaj o potrošenoj energiji po dobu dana za izabranu nekretninu. Ovde uvek fiksno dobijamo podatke za mesečni izveštaj i nedeljni, jer se oni uvek računaju za celu godinu tj nedelju. Za izveštaj po dobu dana možemo imati varijacije.

   REST api se gađa kroz endpointe:
   - POST /api/login
   - GET /api/property/accepted?start=x?end=y
   - GET /api/property/byCity?start=x?end=y
   - GET /api/property/byMonthProperty/id
   - GET /api/property/byTimeOfDay/id
   - GET /api/property/properyByDay

Vremenski interval (dani)	Br korisnika	Br zahteva	Mediana(ms)	Br zahteva po sek
2	50	410	431	37
2	150	1333	1521	40

2	300	2711	3871	43
7	50	431	608	34
7	150	1143	1677	35
7	300	2354	4443	37

Komentar:
Vidimo da se rezultati razlikuju u odnosu na vremenski period. Za razliku od predhodnog primera agregacija podataka za potrošnju i proizvodnju po dobu dana je rađena kroz logiku na serverskoj strani a ne kroz upit. Optimizacija bi bila promena upita i smanjivanje broja iteracija kroz for petlju.

5. Izveštaj za grad za admina
   Početni koraci za ovaj scenario su isti kao i za tabelarni prikaz za admina odakle dobija podatke potrošnju i proizvodnju energije za izabrani vremeski period za izabrani grad

   REST api se gađa kroz endpointe:
   - POST /api/login
   - GET /api/property/accepted?start=x?end=y
   - GET /api/property/byCity?start=x?end=y
   - POST /api/property/propertyEnergy
   - POST /api/property/propertyEnergy

Vremenski interval	Br korisnika	Br zahteva	Mediana(ms)	Br zahteva po sek
6h	100	1121	742	50
12h	100	1219	740	53
Nedelju dana	100	1238	752	53
Mesec dana	100	1232	791	53

Komentar:
Količina podataka obrađena u toku ovih vremenskih perioda nije mnogo varirala te je vreme izvršavanja ostalo relativno slično. Ovo je omogućeno downsaplingom gde su podaci uzimani u intervalu prilagođenom vremenskom periodu. Funkcija za pronalazak intervala traje oko 200 ms.

6. Isključivanje solarnog panel
   Podrazumevani koraci za ovaj scenario su prijava na sistem, dobavljanje svih nekretnina, dobavljanje nekretnine, dobavljanje panela i slanje zahteva za isključivanje

   REST api se gađa kroz endpointe:
   - POST /api/login
   - GET /api/property
   - GET /api/property/id
   - GET /api/device/id

Br korisnika	Br zahteva	Mediana(ms)	Br zahteva po sek
50	605	42	31
150	3897	110	170
300	9905	457	194

7. Dobavljanje podataka za bateriju
   Podrazumevani koraci za ovaj scenario su prijava na sistem, dobavljanje svih nekretnina, dobavljanje nekretnine, dobavljanje baterije i slanje zahteva za dobavljanje stanja baterije u izabranom vremenskom periodu

   REST api se gađa kroz endpointe:
   - POST /api/login
   - GET /api/property
   - GET /api/property/id
   - GET /api/device/id
   - POST /api/device/largeEnergy/readings

Vremenski interval	Br korisnika	Br zahteva	Mediana(ms)	Br zahteva po sek
6h	50	769	36	56
12h	100	2543	68	142
Nedelju dana	150	4042	110	178
Mesec dana	300	9488	447	165

8. Izveštaj za nekretninu za običnog korisnika
   Podrazumevani koraci za ovaj scenario su prijava na sistem, dobavljanje svih nekretnina, dobijanje podatkaka za potrošnju i proizvodnju energije za izabrani vremeski period za izabranu nekretninu

   - POST /api/login
   - GET /api/property
   - POST /api/device/largeEnergy/propertyEnergy
   - POST /api/device/largeEnergy/propertyEnergy

Vremenski interval	Br korisnika	Br zahteva	Mediana(ms)	Br zahteva po sek
6h	50	556	48	40
12h	100	1746	67	100
Nedelju dana	150	3196	178	129
Mesec dana	300	7231	618	137

9. Menjanje procenta punjenja na punjaču

   Podrazumevani koraci za ovaj scenario su prijava na sistem, dobavljanje svih nekretnina, dobavljanje nekretnine, dobavljanje punjača i slanje zahteva za menjanje procenta punjenja

   REST api se gađa kroz endpointe:
   - POST /api/login
   - GET /api/property
   - GET /api/property/id
   - GET /api/device/id
   - PUT /api/device/charger/id/port

Br korisnika	Br zahteva	Mediana(ms)	Br zahteva po sek
50	730	36	50
150	4151	82	188
300	9466	411	169

10. Istorija akcija za punjač

    Podrazumevani koraci za ovaj scenario su prijava na sistem, dobavljanje svih nekretnina, dobavljanje nekretnine, dobavljanje punjača i slanje zahteva za dobavljanje akcija nad punjačem u zadatom vremenskom periodu.

    U dva dana imamo 8640 akcija a u sedam dana 30240.

    REST api se gađa kroz endpointe:
    - POST /api/login
    - GET /api/property
    - GET /api/property/id
    - GET /api/device/id
    - GET /api/device/largeEnergy/id/range

Vremenski interval (dani)	Br korisnika	Br zahteva	Mediana(ms)	Br zahteva po sek
2	50	2000	73	63
2	150	3113	213	119
2	300	7166	590	140
7	50	579	55	43
7	150	4000	132	140
7	300	8380	563	149

Komentar

Prilikom testiranja nije bilo razlike u brzini testova ali količina podataka u velikoj meri narušava performanse na frontu gde učitavanje i filtriranje podataka traje po par sekundi.

Zaključak:

Svi slučajevi su izvedeni sa 100% uspešnosti. Izveštaji za admina su najiscrpniji i zahtevaju najviše vremena te možemo zaključiti da su oni usko grlo sistema. Konkretno kod dobavljanja izveštaja za jednu nekretninu imamo 5 različitih poziva za dobavljanje i obradu podataka. Sama količina podataka koja se dobavi iz Influx baze podataka je smanjena koliko je to moguće downsaplingom. Sama jačina uređaja na kojoj su izvršeni testovi i dužina izvršavanja testova takođe utiče na performanse.

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

Korisnička aplikacija šalje informacije serveru gađajući njene endoint-e dok informacije dobija od servera preko socketa. Server dobijene zahteve po potrebi šalje na određene MQTT kanale preko kojih komunicira sa simulatorima. Korisnička aplikacija i simulatori nemaju direktnu komunikaciju. Svaka stranica koju otvori korisnik otvori i socket konekciju preko koje dobavlja neophodne podatke, dok svaki simulator sluša zahteve isključivo za njega. Većina podataka se upisuje u Iflux preko Telegraf-a. Izuzetak su podaci za proizvodnju i potrošnju električne energije koji se naknadno računaju na svaki minut na serveru.

Za ovu grupu testinja su ispitani sledeći uređaji:

1. Solarni panel
   U sklopu kojeg se vrši uključivanje i isključivanje panela i dobavljanje podataka za istoriju akcija i proizvodnju energije. Uključivanje i isključivanje panela smo ispitali u predhodnoj grupi testova i možemo primetiti da povečanje simulatora jeste uticalo na performanse. Na vise od 950 pokrenutih niti MQTT konekcija puca svaki put i više se ne dobija ni jedna poruka od simulatora, na 800 niti dobave se za svaku nit ispravni podaci ali prilikom testiranja Locust-om nekad se prekine veza kao posledica prevelikog broja zahteva na isti kanal. Naredni podaci su rađeni za pokretanje 800 niti. Sa 150 korisnika vidimo duplo lošije performanse nego u predhodnoj grupi testiranja.

Br korisnika	Br zahteva	Mediana(ms)	Br zahteva po sek
50	685	36	50
150	2743	488	102
300	9000	616	170

Broj niti	Vreme odziva za iskljucinjanje svih panela (sek)
250	1.98
500	4.79
800	5.8

   Simulacija je uticala na količinu podataka koja se dobavlja za istoriju akcija i za prikaz proizvodnje energije ali downsapling-om je regulisan veći protok podataka i ne predstavlja usko grlo.

2. Baterija
   U sklopu koje imamo prikaz vrednosti baterije kao i određene izveštaje u sklopu nekretnine, reatime prikaz potrošnje nekretnine koji preko socketa dobavlja podatke i istorijski prikaz potrošnje. Statičko dobavljanje informacija o potrošnji i stanju baterije je ostalo pri istim performansama. Kako je realtime pretplaćen na socket koji uvek šalje podatke na minut a ne kako stignu određeni podaci iz simulatora nema kočenja ili kašnjenja sa te strane. Izveštaji su sami po sebi usko grlo sistema jer ih ima mnogo i barataju nad velikim skupom podataka.

3. Punjač

   U sklopu kojeg imamo menjanje nivoa punjenja, dobavljanje podataka za istoriju akcija i potrošnju i socket konekciju za dobavljanje informacija o vozilu na punjaču. Za istoriju akcija smo ispitali vreme dobavljanja u prehodnoj grupi testova i postoji usporenje prilikom dobavljanja većeg broja podataka na frontu. Promena podataka o vozilu na punjaču je radila normalno i vrednosti su se redovno menjale tako da slanje velikog broja poruka na otvoreni socket nije bio problem.

Broj niti	Vreme odziva za iskljucinjanje panela (sek)
250	1.54
500	5.28
800	5.87

Zaključak:

Najveći broj niti koji je pušten je 800 gde sa jednim korisnkom radi MQTT konekcija, dok uz testiranje Locus-om sa vise od 100 korisnika dolazi do prekida. Socketi nisu predstavljali usko grlo i na korisničkoj aplikaciji nije bilo vidnog kašnjenja. Više niti nije bilo moguće pokrenuti jer neki simulatori nisu dobili ispravne podatke i MQTT je prekidao konekciju.