

Analiza zmogljivosti oblačnih in strežniških storitev

Uredil prof. dr. Miha Mraz

Maj 2018

Kazalo

Predgovor	iii
1 Testiranje oblačne strežbe dražb v realnem času (J. Bešter, J. Debelak, A. Orehek)	1
1.1 Opis problema	1
1.2 Opis ponudnikov in tehnologij	2
1.3 Analiza lokacije	4
1.4 Metrike	7
1.5 Načrt dela v prihodnje	9

Predgovor

Pričujoče delo je razdeljeno v deset poglavij, ki predstavljajo analize zmogljivosti nekaterih tipičnih strežniških in oblačnih izvedenk računalniških sistemov in njihovih storitev. Avtorji posameznih poglavij so slušatelji predmeta *Zanesljivost in zmogljivost računalniških sistemov*, ki se je v štud.letu 2017/2018 predaval na 1. stopnji univerzitetnega študija računalništva in informatike na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Vsem študentom se zahvaljujem za izkazani trud, ki so ga vložili v svoje prispevke.

prof. dr. Miha Mraz, Ljubljana, v maju 2018

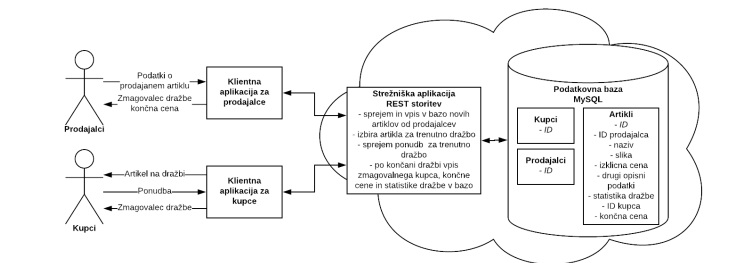
Poglavje 1

Testiranje oblačne strežbe dražb v realnem času

Jurij Bešter, Juš Debelak, Andrej Orehek

1.1 Opis problema

Realizirali bomo storitev strežbe dražbe v realnem času z beleženjem statistike. V oblaci strežnik se bosta povezovala dva tipa klientov in sicer manj številčni prodajalci, ki bodo v dražbo ponudili artikel z izklicno ceno, ter številčnejši kupci, ki bodo v omejenem času oddali ponudbe za nakup artikla. Strežnik bo moral posredovati ponudbo prodajalca trenutno prijavljenim kupcem, nato pa od kupcev sprejeti njihove pravilne (oddane v definiranem omejem času) ponudbe. Ponudbe bo moral v omejenem času sprocesirati, torej najti najvišjo ponudbo ter povezati zmagovalnega kupca in prodajalca, ter izdelati in shraniti statistiko posamezne dražbe. Shema delovanja je prikazana na sliki 1.1.



Slika 1.1: Shema delovanja dražbene storitve.

1.2 Opis ponudnikov in tehnologij

Za ponudnika oblačnih storitev smo si izbrali Amazonov AWS (*Amazon Web Services*). To je zbirka storitev, ki olajša razvoj spletnih storitev. Omogoča razvoj aplikacijske logike v različnih okoljih (framework) in programskih jezikih. Za hranjenje podatkov nudi široko paleto različnih podatkovnih baz.

Aplikacijska logika opisane rešitve je implementirana v okolju node.js in programskim jezikom JavaScript. Za hranjenje podatkov je uporabljena relacijska podatkovna baza MySQL. Komunikacija med odjemalci in strežnikom je realizirana po REST pristopu. REST pristop predvideva uporabo HTTP zahtevkov za manipulacijo (POST, DELETE) in pridobivanje podatkov (GET).

Uporabljene storitve v rešitvi, ki jih AWS ponuja:

- **Amazon API Gateway:** Vstopna točka za zahtevke odjemalcev, ki preusmerja zahtevke na določene Lambda funkcije glede na tip;
- **Amazon Lambda:** Lambda funkcije vsebujejo aplikacijsko logiko, ki obdelujejo posamezne zahtevke in so povezane z relacijsko podatkovno bazo (RDS);
- **Amazon RDS:** Omogoča izbiro in uporabo različnih relacijskih podatkovnih baz za hranjenje podatkov. V opisani rešitvi je uporabljena MySQL relacijska podatkovna baza;

Rešitev vsebuje naslednje dostopne točke:

- **/user:** z zahtevkom POST ustvarimo novega uporabnika. Strežnik ob uspešnem vnosu vrne id uporabnika. Primer POST zahtevka:

```
{
  name: "User name"
}
```

- **/user/{id}:** z zahtevkom POST posodobimo podatke za uporabnika z izbranim id-jem. Oblika POST zahtevka je enaka kot pri vnosu novega uporabnika. Z zahtevkom DELETE pa uporabnika izbrišemo.
- **/item:** z zahtevkom GET pridobimo seznam vseh predmetov na dražbi, z zahtevkom POST pa pošljemo na dražbo nov predmet. Primer POST zahtevka:

```
{
  "name":"Test item",
  "description":"Test item description",
  "starting_price":1000,
  "user_id":4,
  "bidding_time":3600
}
```

- **/item/{id}:** z zahtevkom POST posodobimo predmet z izbranim id-jem. Oblika POST zahtevka je enaka kot pri vnosu novega predmeta. Z zahtevkom DELETE pa predmet izbrišemo.

- **/bid/{id}**: z zahtevkom POST podamo novo ponudbo za predmet z izbranim **id**-jem, z zahtevkom GET pa pridobimo že oddane ponudbe za ta predmet. Primer POST zahtevka:

```
{
  "user_id": 5,
  "bid": 2000
}
```

- **/bid/{id}/max**: z zahtevkom GET pridobimo trenutno najvišjo ceno za predmet z izbranim **id**-jem;

Na vse POST zahtevke strežnik odgovori z podatki o ospešnosti operacije. Primer odgovora:

```
{
  "fieldCount": 0,
  "affectedRows": 1,
  "insertId": 0,
  "serverStatus": 2,
  "warningCount": 0,
  "message": "(Rows matched: 1 Changed: 0 Warnings: 0)",
  "protocol41": true,
  "changedRows": 0
}
```

Za lokacijo naših storitev smo izbrali *us-west-2*, ki se nahaja v državi Oregon, saj je to lokacija, ki jo ponudnik predlaga za brezplačno raven uporabe.

1.3 Analiza lokacije

Lokacijo *us-west-2* smo analizirali s 24-urnim testom, med katerim smo vsako minuto izvedli PING, s katerim smo izmerili čas med klientom in strežnikom (angl. *round-trip time* - *RTT*). Hkrati smo s TRACEROUTE beležili pot paketa med klientom in strežnikom. Test smo izvajali 1. maja 2018.

Test smo hkrati izvajali iz treh različnih lokacij:

- 1. lokacija: Ljubljana [Telemach - kabelski internet - 150Mbps/4Mbps]
- 2. lokacija: Šentvid pri Stični [Telekom - ADSL - 10Mbps/1Mbps]
- 3. lokacija: Falkenstein-DE [Hetzner - optika - 1Gbps/1Gbps] (Najet namenski strežnik pri ponudniku gostovanja strežnikov Hetzner. Ponudnik oglašuje privatne povezave v omrežja večjih ponudnikov spletnih vsebin, med drugimi tudi Amazon.)

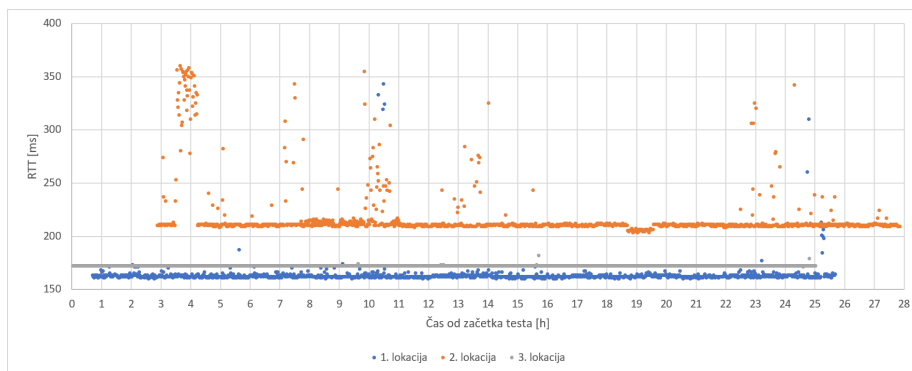
Na sliki 1.2 lahko vidimo, da so *RTT* časi in njihova konsistentnost med testnimi lokacijami različni. Izračunali smo povprečno *RTT* vrednost, in standardni odklon *RTT* vrednosti za posamezne lokacije:

- 1. lokacija: Povprečen *RTT*: 164,0ms, Standardni odklon *RTT*: 10,2ms
- 2. lokacija: Povprečen *RTT*: 217,5ms, Standardni odklon *RTT*: 24,5ms
- 3. lokacija: Povprečen *RTT*: 173,0ms, Standardni odklon *RTT*: 0,3ms

Na izpisih 1.1, 1.2 in 1.3 so vidne začetne poti med klienti in strežnikom. Poti smo natančneje pregledali v časovnih obdobjih ko so *RTT* rezultati za 1. in 2. lokacijo bistveno odstopali od njunih siceršnjih povprečij. Ugotovili smo, da je na 1. in 2. lokaciji do povečanih časov prišlo predvsem zaradi zakasnitev na poti znotraj domene naših ISPjev. Na lokaciji 3., ki je na komercialni internetni povezavi so bili *RTT* časi tekom testa zelo podobni. Poti so se na vseh lokacijah tekom testa rahlo spreminjale, a med različnimi potmi ni bilo bistvenih časovnih razlik.

Kljub temu, da je 3. lokacija na komercialni internetni povezavi z oglaševanimi privatnimi povezavami pa je bil povprečni *RTT* 3. lokacije višji od 1. lokacije, ki se nahaja na potrošniški internetni povezavi. Iz TRACEROUTE izpisov lahko razberemo, da je bil prihod v ZDA iz 3. lokacije (116,4ms) počasnejši od tistega iz 1. lokacije (102,3ms). Pot preko Atlantskega oceana iz 1. lokacije je potekala Francija - New York (70,3ms), iz 2. lokacije pa Švedska - Chicago (24,3ms). Kljub temu da je pot preko Atlantskega oceana Švedska - Chicago skoraj 3x hitrejša, pa je celotna pot daljša zaradi potratnega skoka iz Velike Britanije v Švedsko, ki znaša kar 66,1ms.

Razlog za veliko višji povprečni *RTT* iz 2. lokacije pa je bila prav dolga pot do vzhodne obale ZDA, ki je znašala kar 181,4ms. Točna pot 2. povezave preko Atlantskega oceana ni razvidna.



Slika 1.2: Rezultati PING testa iz vseh treh testnih lokacij.

Listing 1.1: TRACEROUTE izpis iz 1. lokacije

```
tracert to 54.213.245.230 (54.213.245.230), 30 hops max, 60 byte packets
 1  router.asus.com (192.168.26.1) 0.731 ms
 2  10.224.128.1 (10.224.128.1) 7.901 ms
 3  217-72-74-225.ipv4.telemach.net (217.72.74.225) 13.377 ms
 4  185.66.148.235.ipv4.telemach.net (185.66.148.235) 13.438 ms
 5  185.66.148.235.ipv4.telemach.net (185.66.148.235) 12.655 ms
 6  185.66.148.219.ipv4.telemach.net (185.66.148.219) 19.456 ms
 7  185.66.148.219.ipv4.telemach.net (185.66.148.219) 18.805 ms - (Slovenija)
 8  10ge1-5.core1.viel.he.net (216.66.82.73) 16.948 ms - (Avstrija)
 9  100ge13-1.core1.par2.he.net (184.105.65.5) 32.024 ms - (Francija)
10  100ge14-1.core1.nyc4.he.net (184.105.81.77) 102.326 ms - (ZDA, NY)
11  100ge14-1.core1.tor1.he.net (184.105.80.10) 122.009 ms
12  100ge6-1.core1.ywg1.he.net (184.105.64.102) 133.523 ms
13  100ge10-1.core1.yyc1.he.net (184.105.222.98) 146.205 ms
14  100ge10-2.core1.yvr1.he.net (184.105.64.113) 161.939 ms
15  100ge10-2.core1.sea1.he.net (184.105.64.109) 158.615 ms
16  paix01-sea4.amazon.com (198.32.134.41) 158.612 ms
17  52.95.52.120 (52.95.52.120) 157.282 ms
18  52.95.52.97 (52.95.52.97) 156.636 ms
19  *
20  54.239.42.223 (54.239.42.223) 167.441 ms
21  54.239.43.136 (54.239.43.136) 167.747 ms
22  *
23  52.93.13.4 (52.93.13.4) 169.798 ms
24  52.93.12.37 (52.93.12.37) 166.918 ms
25  52.93.12.63 (52.93.12.63) 166.653 ms
26  52.93.240.19 (52.93.240.19) 164.178 ms
```

Listing 1.2: TRACEROUTE izpis iz 2. lokacije

```
tracert to 54.213.245.230 (54.213.245.230), 30 hops max, 60 byte packets
 1  TP-LINK (192.168.16.1) 0.563 ms
 2  95.176.255.241 (95.176.255.241) 14.349 ms
 3  95.176.241.151 (95.176.241.151) 13.821 ms
 4  212.162.28.5 (212.162.28.5) 14.237 ms - (Slovenija)
 5  *
 6  4.16.146.90 (4.16.146.90) 181.369 ms - (ZDA, NY)
 7  *
 8  *
 9  *
10  *
11  *
12  *
13  52.93.15.10 (52.93.15.10) 214.913 ms
14  *
15  52.93.14.142 (52.93.14.142) 219.543 ms
16  52.93.240.47 (52.93.240.47) 212.191 ms
```

Listing 1.3: TRACEROUTE izpis iz 3. lokacije

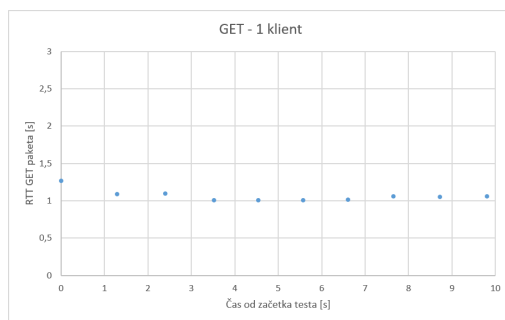
POGLAVJE 1. TESTIRANJE OBLAČNE STREŽBE DRAŽB V REALNEM
6 ČASU (J. BEŠTER, J. DEBELAK, A. OREHEK)

```
traceroute to 54.213.245.230 (54.213.245.230), 30 hops max, 60 byte packets
 1  static .65.57.243.136, clients.your-server.de (136.243.57.65) 0.369 ms
 2  core24.fsn1.hetzner.com (213.239.203.193) 0.333 ms
 3  core11.nbg1.hetzner.com (213.239.245.225) 2.761 ms
 4  juniper4.dc2.nbg1.hetzner.com (213.239.203.138) 2.808 ms
 5  hbg-b1-link.telia.net (213.248.70.0) 16.184 ms
 6  hbg-bb4-link.telia.net (213.155.135.86) 16.125 ms - (Nemcija)
 7  ldn-bb3-link.telia.net (80.91.249.10) 26.039 ms - (UK)
 8  nyk-bb4-link.telia.net (62.115.136.185) 92.127 ms - (Svedska)
 9  chi-b21-link.telia.net (80.91.246.162) 116.386 ms - (ZDA, Chicago)
10  sea-b2-link.telia.net (62.115.117.48) 170.325 ms
11  sea-b1-link.telia.net (62.115.115.2) 164.815 ms
12  amazon-ic-307562-sea-b1.c.telia.net (213.248.92.242) 157.532 ms
13  *
14  *
15  *
16  *
17  *
18  *
19  *
20  *
21  *
22  *
23  54.239.48.183 (54.239.48.183) 185.700 ms
```

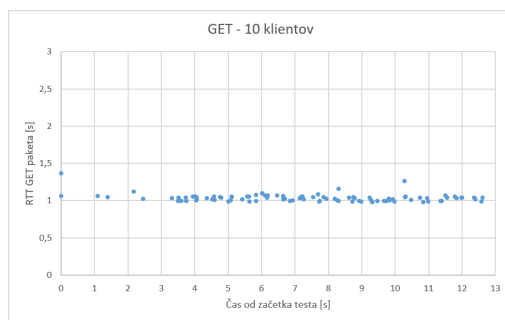
1.4 Metrike

Za testiranje delovanja smo opravili dva testa. Pri obeh smo teste izvajali iz 3. lokacije (gl. poglavje 1.3), 13. aprila 2018 ob 12. uri.

V 1. testu so klienti od strežnika 10 krat zapored pridobili seznam vseh predmetov na dražbi z metodo GET /**item** . Merili smo RTT čas GET paketa, od pošiljanja zahteve do prejete odgovora. Test smo ponovili z 1, 10 in 100 klienti.

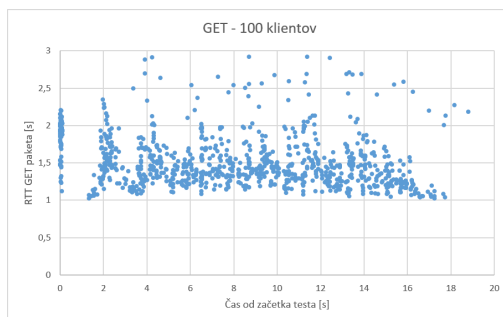


Slika 1.3: 1. test - 1 klient: povprečni čas = 1,067s, trajanje testa = 9,8s.



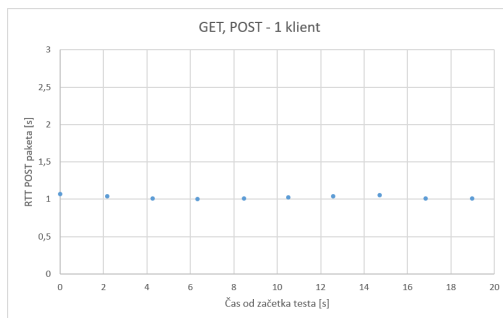
Slika 1.4: 1. test - 10 klientov: povprečni čas = 1,258s, trajanje testa = 12,6s.

POGLAVJE 1. TESTIRANJE OBLAČNE STREŽBE DRAŽB V REALNEM
8 ČASU (J. BEŠTER, J. DEBELAK, A. OREHEK)

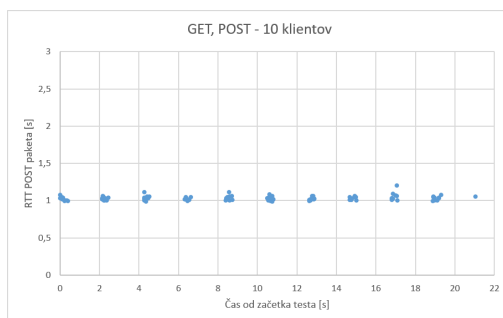


Slika 1.5: 1. test - 100 klientov: povprečni čas = 1,550s, trajanje testa = 18,8s.

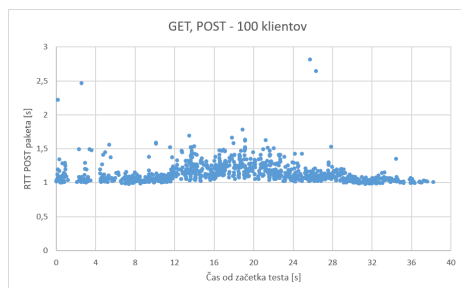
V 2. testu so klienti od strežnika 10 krat zapored pridobili trenutno najvišjo ceno za predmet z metodo GET /**bid**/**{id}**/**max**, ter nato podali novo ponudbo za predmet z metodo POST /**bid**/**{id}**. Merili smo RTT čas POST paketa, od pošiljanja zahteve do prejete odgovora. Test smo ponovili z 1, 10 in 100 klienti.



Slika 1.6: 2. test - 1 klient: povprečni čas = 1,028s, trajanje testa = 19,0s.



Slika 1.7: 2. test - 10 klientov: povprečni čas = 1,033s, trajanje testa = 21,0s.



Slika 1.8: 2. test - 100 klientov: povprečni čas = 1,325s, trajanje testa = 38,2s.

1.5 Načrt dela v prihodnje

- implementacija testa, pri katerem bomo lahko v rezultatih jasno ločili čas potovanja paketa med klientom in strežnikom ter čas strežbe.
- daljši test obremenitve iz ene tesne lokacije, z višanjem števila klientov do točke odpovedi
- daljši test obremenitve iz večih testnih lokacij, z višanjem števila klientov do točke odpovedi

Literatura