

# Analiza zmogljivosti oblačnih in strežniških storitev

Uredil prof. dr. Miha Mraz

Maj 2017



# Kazalo

<b>Predgovor</b>	<b>iii</b>
<b>1 Analiza zmogljivosti oblačne storitve c9</b>	<b>1</b>
1.1 Opis problema . . . . .	1
1.2 Namen . . . . .	1
1.3 Izbira ponudnikov . . . . .	1
1.3.1 Cloud 9 . . . . .	3
1.4 Izbira tehnologij . . . . .	3
1.4.1 Tehnologija v oblaku . . . . .	3
1.4.2 Tehnologija za avtomatizacijo odjemalcev . . . . .	3
1.5 Rezultati meritev . . . . .	3
1.5.1 Testiranje I. . . . .	3



# 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 2016/2017 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 2017*



## Poglavje 1

# Analiza zmogljivosti oblačne storitve Cloud9

Žiga Kokelj, Tadej Hiti,  
Miha Bizjak, Matej Kristan

### 1.1 Opis problema

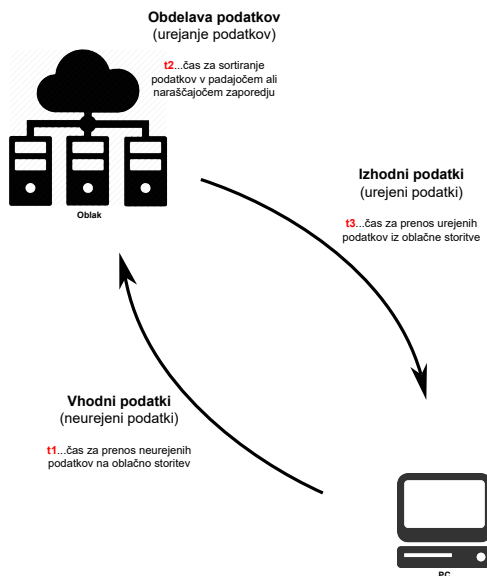
Današnje dni se uveljavljajo oblačne storitve, saj so s stališča uporabnika najenostavnejše za uporabo. Naša naloga je implementirati prenos datoteke na oz. z oblačne storitve in breme na oblačni storitvi, za katero smo si izbrali sortiranje numeričnih podatkov. Na sliki 1.1 je grafičen prikaz opisanega problema.

### 1.2 Namen

Naše testiranje bo obsegalo merjenje različnih izvajalnih časov na podlagi katerih bi prišli do podatkov o zmogljivosti sistema. Breme sistema bodo različni algoritmi sortiranja podatkov. Namen naše naloge bo ugotoviti zmogljivost zastonjske ponudbe z vidika različnih metrik zmogljivosti.

### 1.3 Izbira ponudnikov

Zaradi predhodnih izkušenj z oblačno storitvijo Cloud9 smo se odločili za njihovo zastonjsko ponudbo. Cloud9 ponuja razvojno okolje in operacijski sistem Ubuntu v katerem lahko pišemo ali pa izvajamo različne programe.



Slika 1.1: Shema delovanja sistema.

### 1.3.1 Cloud 9

Cloud9 je delavno orodje, ki je namenjeno programiranju v brskalniku. Torej vpišemo URL v brskalnik in že imamo vse pripravljeno za pisanje prvega programa. Pri zastonski naročnini nam dajo 512MB primarnega pomnilnika, in 2GB prostora na disku.

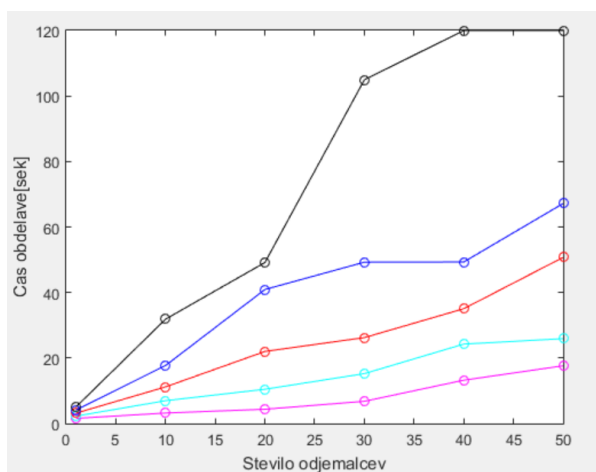
## 1.4 Izbira tehnologij

V tem razdelku so na kratko opisane vse izbrane tehnologije, ki smo jih realizirali sami za našo analizo.

### 1.4.1 Tehnologija v oblaku

V oblachni storitvi smo implementirali strežnik, ki je napisan v jeziku javascript z uporabo knjižnice Node.js [1], ki v odvisnosti od URL zahteve posreduje temu primerno datoteko na odjemalca. Strežnik poskrbi za prenašanje datotek in zagon sortirnega programa za sortiranje podatkov na strežniku.





Slika 1.2: Graf časa v odvisnosti od števila odjemalcev.

### 1.4.2 Tehnologija za avtomatizacijo odjemalcev

Zaradi avtomatskega testiranja smo napisali tudi skripto v programskem jeziku python [2], ki omogoča avtomatsko pošiljanje datoteke in URL zahteve na strežnik, ter kot odgovor prejme urejeno datoteko z urejenimi podatki. Seveda pa ob tem še zabeležimo čas pred pošiljanjem zahteve in čas po prejemu urejene datoteke, da dobimo izvajalni čas celotne procedure. Ker je odjemalcev lahko večje število, smo ta problem rešili z nitmi, kjer vsaka nit predstavlja enega odjemalca in pošilja zahteve na strežnik.

## 1.5 Rezultati meritev

V tem razdelku so opisani način testiranja naše storitve in rezultati meritev.

### 1.5.1 Testiranje I.

Naključno smo generirali po eno datotek z 10000/15000/20000/25000/30000 integer števili, ki jih je nato 1/10/20/30/40/50 odjemalcev pošiljalo na strežnik. Izmerili smo čase potrebne za prejetje urejene datoteke in prišli do povprečnih časov. Testiranje je bilo izvedeno okoli popoldneva. Vse meritve so prikazane na sliki 1.2 in v tabelah 1.3, 1.4, 1.5, 1.6, 1.7, kjer je prikazan povprečen čas obdelave. Pri datotekah z 25 tisoč števili in 40 do 50 odjemalci je prišlo do napake 'existing connection was forcibly closed by remote host'. Pri datotekah z 30 tisoč števili in 40 do 50 odjemalci pa nekateri odjemalci po dvo minutnem čakanju niso dobili odgovora in povezava se prekine. Zakaj in kje pride do napake še nismo ugotavljali.

Število odjemalcev	Čas obdelave[sek]
1	1.57
10	3.20
20	4.36
30	6.78
40	13.23
50	17.62

Slika 1.3: Tabela časov obdelav datoteke z 10 tisoč integer števil.

Število odjemalcev	Čas obdelave[sek]
1	2.27
10	6.97
20	10.46
30	15.23
40	24.25
50	25.91

Slika 1.4: Tabela časov obdelav datoteke z 15 tisoč integer števil.

Število odjemalcev	Čas obdelave[sek]
1	3.18
10	11.18
20	22.01
30	26.23
40	35.05
50	50.74

Slika 1.5: Tabela časov obdelav datoteke z 20 tisoč integer števil.

Število odjemalcev	Čas obdelave[sek]
1	4.16
10	17.71
20	40.92
30	49.26
40	49.32(napake)
50	67.18(napake)

Slika 1.6: Tabela časov obdelav datoteke z 25 tisoč integer števil.

Število odjemalcev	Čas obdelave[sek]
1	5.09
10	31.85
20	49.17
30	104.93
40	120+(no answer)
50	120+(no answer)

Slika 1.7: Tabela časov obdelav datoteke z 30 tisoč integer števili.



# Literatura

- [1] “Node.js.” Dosegljivo: <https://nodejs.org/en/>.
- [2] “Python.” Dosegljivo: <https://www.python.org/>.