

Analiza zmogljivosti obladžnih in
stredžniških storitev

Uredil prof. dr. Miha Mraz

Maj 2017

Kazalo

Predgovor	iii
1 Analiza zmogljivosti oblačne storitve c9	1
1.1 Opis problema	1
1.2 Izbira ponudnikov	1
1.3 Izbira tehnologij	2
1.3.1 Tehnologija v oblaku	2
1.3.2 Tehnologija za avtomatizacijo odjemalcev	2
1.4 Rezultati meritev	3
1.4.1 Testiranje I.	3
1.5 Plan za prihodnje delo	4

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

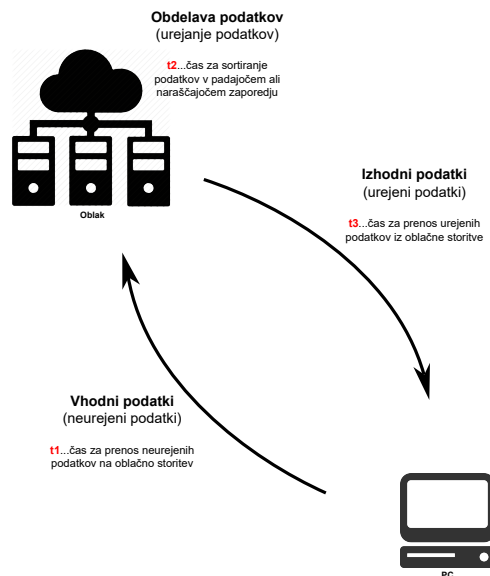
Žiga Kokelj, Tadej Hiti,
Miha Bizjak, Matej Kristan

1.1 Opis problema

Današnje dni se vse bolj 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. Naše testiranje bo obsegalo merjenje različnih izvajalnih časov na podlagi katerih bi prišli do podatkov o zmogljivosti sistema. Breme oblačnega sistema bodo datoteke različnih velikosti, ki bodo vsebovale naključno generirana števila. Oblačna storitev pa bo vsebine datotek uredila po izbranem algoritmu za urejanje števil. Namen naše naloge je ugotoviti zmogljivost zastonske oblačne storitve z vidika različnih metrik.

1.2 Izbira ponudnikov

Zaradi predhodnih izkušenj z oblačno storitvijo Cloud9 smo se odločili za njihovo zastonsko ponudbo. Cloud9 ponuja razvojno okolje in operacijski sistem Ubuntu v katerem lahko pišemo ali pa izvajamo različne programe. Pri brezplačni naročnini nam dajo v uporabo 512MB RAM-a in 2GB prostora na disku.



Slika 1.1: Shema delovanja sistema.

1.3 Izbira tehnologij

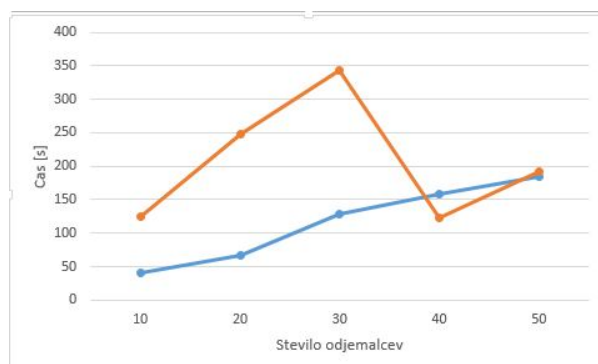
V tem razdelku so na kratko opisane vse izbrane tehnologije, ki smo jih realizirali za našo analizo.

1.3.1 Tehnologija v oblaku

V oblaki storitvi smo implementirali strežnik, ki je napisan v jeziku javascript z uporabo knjižnice Node.js [1]. Strežnik od uporabnika sprejme datoteko, jo avtomatsko sortira z določenim algoritmom za sortiranje števil ter nato sortirano datoteko pošlje nazaj uporabniku.

1.3.2 Tehnologija za avtomatizacijo odjemalcev

Zaradi avtomatskega testiranja smo napisali 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 ob tem zabeležimo še čas pred pošiljanjem zahteve in čas po prejetju 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.



Slika 1.2: Graf časa obdelave v odvisnosti od števila odjemalcev.

1.4 Rezultati meritev

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

1.4.1 Testiranje I.

Naključno smo generirali po eno datoteko z 10000/15000/20000/25000/30000 integer števili, ki jih je nato 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 ob 8 ih zjutraj v Ljubljani v sredo 3.5.2017. Vse meritve so prikazane na sliki 1.2 in v tabelah 1.3, 1.4, kjer je prikazan povprečen čas obdelave. Pri vseh testiranjih se je uporabilo 10 različnih vnaprej naključno generiranih datotek. Za sortiranje števil se uporablja algoritem bubble sort. Pri datotekah z več števili in veliko odjemalci je prihajalo do nekaterih napak, ki jih trenutno še odpravljamo.

Število odjemalcev	Povprečni čas obdelave[s]	Standardna deviacija
10	41.3034	1.0168
20	66.1481	1.7781
30	127.7942	8.3947
40	158.8352	6.2111
50	184.7879	3.6371

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

Število odjemalcev	Povprečni čas obdelave[s]	Standardna deviacija
10	124.4322	2.7692
20	248.6854	10.797
30	344.0970	19.4963
40	123.3930	21.8550
50	192.7521	23.4215

Slika 1.4: Tabela časov obdelav datoteke z 20 tisoč integer števili.

1.5 Plan za prihodnje delo

V prihodnje moramo odkriti in odpraviti težave pri testiranju večjih datotek z veliko odjemalci. Po odpravljenih težavah pa bo potrebno izvesti še več testov v različnih delih dneva in iz različnih lokacij po Sloveniji, da bomo podatke lahko primerjali med seboj.

Literatura

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