Analiza zmogljivosti oblačnih in strežniških storitev

Uredil prof. dr. Miha Mraz

Maj 2018

Kazalo

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 čtudija 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 4

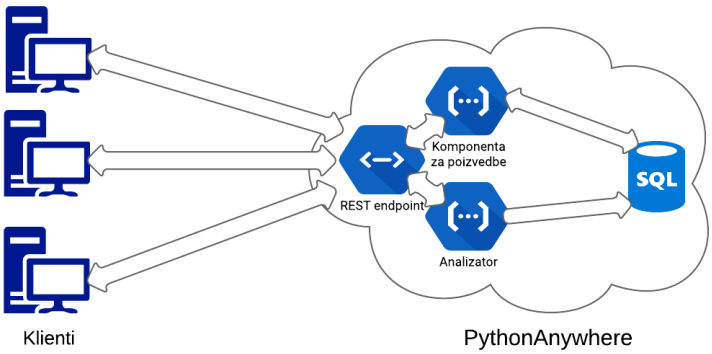
Analiza zmogljivosti ponudnika PythonAnywhere

Žiga Babnik, Mitja Maren, Matej Horvat, Gašper Jelovčan

4.1 Opis oblačne storitve

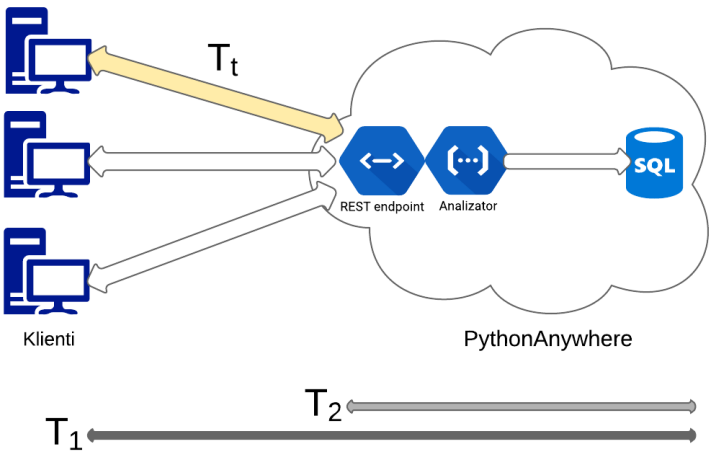
V pričujočem poglavju predstavimo zanesljivost in zmogljivost storitve oblačnega ponudnika PythonAnywhere. Analizirali jo bomo z aplikacijo štetja frekvenc besed v tekstovnih datotekah. Odjemalci bodo na strežnik poslali neko besedilo, strežnik pa bo izračunal število pojavitev posameznih besed v njem. Primer dobrega analizatorja lahko najdemo v članku [4], kjer je tudi dobro opisano o štetju besed. Mi sicer ne bomo imeli zelo naprednega analizatorja, ker nas bolje zanimajo performance strežnika. Članek [6] opisuje o aplikaciji TextArc, ki grafično na elipsi izpiše text ter poudari besede, ki se pojavijo večkrat. Rezultati bodo shranjeni v podatkovni bazi, tako da bo mogoče po njih pozneje poizvedovati in primerjati različna besedila. Kot zanimivost, članek [5] opisuje o zakonu pojavitve besed v besedilih. Aplikacija sicer nima široko uporabne vrednosti, vendar menimo, da se z njo da dobro preveriti procesorsko moč, hitrost omrežne povezave in hitrost delovanja podatkovne baze strežnika. Slika 4.1 prikazuje zgradbo aplikacije.

4.2 Izbira tehnologij



Slika 4.1: Shema delovanja aplikacije.

Uporabili smo ponudnik storitev PythonAnywhere, ki je specializiran za gostovanje aplikacij, napisanih v jeziku Python. V ozadju za svoje delovanje uporablja Amazon EC2 [1] in ga je mogoče z nekaj omejitvami uporabljati zastonj.



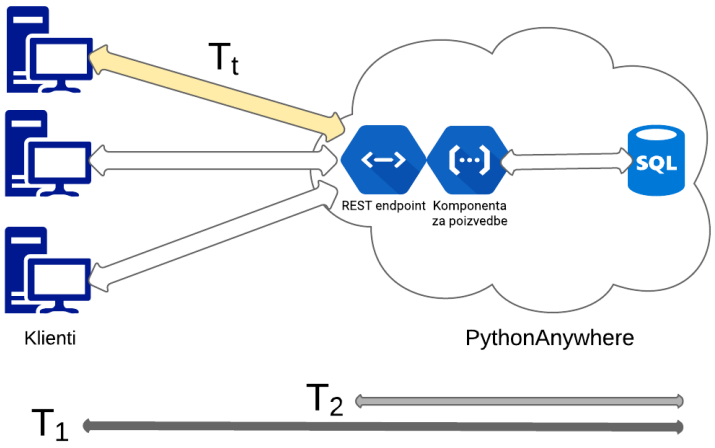
Slika 4.2: Shema delovanja aplikacije z vidika analizatorja.

Slika 4.2 prikazuje tok datotek oziroma zahtev preko aplikacije z vidika ana-

lizatorja. Na skici so označeni tudi časi, ki jih bomo merili oziroma izračunali. T_1 predstavlja celoten čas potreben za zahtevo, meritev za čas T_1 se v celoti izvede na strani klienta. Čas začetka in konca pošiljanja odčitamo s pomočjo ukaza *time.time*(), končni čas T_1 pa izračunamo, kot razliko med časom konca in začetka pošiljanja, kot to prikazuje slika 4.3. T_2 predstavlja čas delovanja analizatorja, ta se izmeri na strani oblačne storitve in sicer se meritev začne, ko oblačna storitev prejeme zahtevo, zaključi pa, ko analizator zaključi s preštevanjem besed. T_t pa je čas potovanja zahteve po omrežju, ki ga lahko izračunamo na podlagi razlike izmerjenih časov T_1 in T_2. Za meritve povezane z analizatorjem je pomembno še dejstvo, da se med vsako posamezno meritvijo počisti baza na oblačni storitvi, tako zagotavimo večjo neodvisnost posameznih meritev.

```
zacetniCas = time.time()
response = requests.post(restResource, data=data)
celotniCas = time.time() - zacetniCas
```

Slika 4.3: Prikaz kode za merjenje časa T_1



Slika 4.4: Shema delovanja aplikacije z vidika komponente za poizvedbe.

Slika 4.4 prikazuje tok datotek oziroma zahtev preko aplikacije z vidika komponente za poizvedbe. Na skici so označeni tudi časi, ki jih bomo merili oziroma lahko izračunali. T_1 predstavlja celoten čas potreben za zahtevo, T_2 predstavlja čas delovanja analizatorja, čas T_t pa je čas potovanja, ki ga lahko izračunamo na podlagi izmerjenih časov T_1 in T_2. Samo merjenje tu poteka drugače kot pri analizatorju, saj je znotraj zahteve potrebno čakati na odgovor baze, pri čemer to ni potrebno v primeru analizatorja.

4.3 Definicija bremena

Breme predstavljajo tekstovne datoteke knjig, pridobljene s spletne strani Project Gutenberg [2] s pomočjo krajše skripte. Te uporabljamo na zaključnih testih, ki predstavijo odziv aplikacije v realni situaciji in potrdijo naše razumevanje pridobljeno s pošiljanjem raznih sintetičnih bremen. Ta so podrobneje predstavljena pri posameznih poizkusih, v splošnem pa so to tekstovne datoteke z nizi znakov, kjer se spreminja povprečna dolžina nizov, število besed in druge lastnosti, ki nas zanimajo.

4.4 Definicija metrik

To poglavje je odveč... metrike se predstavi že v prejšnjem poglavju ... pobriši pred koncem

Meritve se izvajajo na strani odjemalca. Meritev se začne, ko ta sproži zahtevo na oblačno storitev, konča pa se, ko odjemalec prejme odgovor s strežnika. Tako dobimo čas T_1, ki predstavlja celotni čas prenosa in procesiranja na strani oblačne storitve.

Prav tako se meritve izvajajo na strežnikovi strani. Prva meritev je čas procesiranja besedila. Označimo jo s T_2. Sproži jo končna točka REST, ko sprejme zahtevo od odjemalca in konča, ko dobi odgovor od analizatorja. Tretja meritev, ki jo označimo s T_3, predstavlja čas od oddaje podatkov podatkovni bazi do odgovora podatkovne baze. Izvede se po analizi besedila.

Zaradi večjega števila zahtevkov bodo vse zahteve vsebovale enolični identifikator, ki jih bo spremljal na celotni poti.

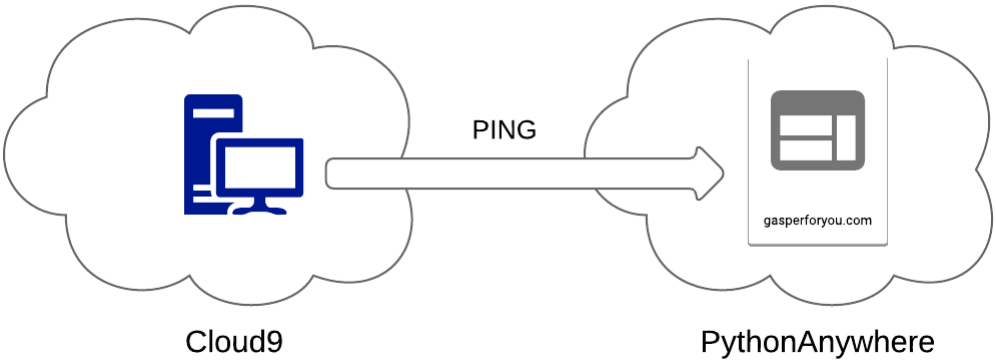
S pomočjo izmerjenih časov T_1, T_2 ter T_3 lahko izračunamo čas potovanja med odjemalcem ter oblačno storitvijo, čas procesiranja besedila ter čas shranjevanja v bazo, in sicer po sledečih formulah:

- Celotni čas: $T = T_1$,
- Čas delovanja analizatorja: $T_p = T_2 - T_3$,
- Čas delovanja podatkovne baze: $T_{db} = T_3$,
- Čas potovanja med klienti in oblačno storitvijo: $T_t = T_1 - T_2$.

Vsi navedeni časi so označeni na sliki 4.1.

4.5 Opis klientov

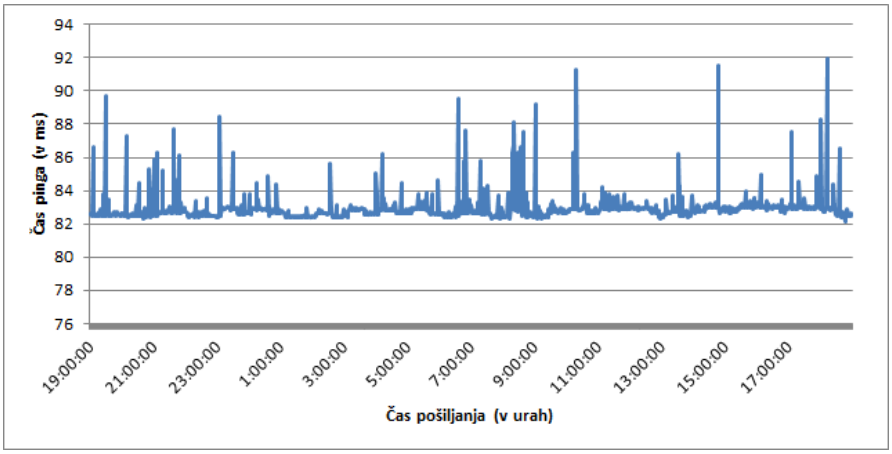
4.5.1 Klienti za testiranje analizatorja



Slika 4.5: Slika ponazarja potek testiranja povezave.

Klienti, ki testirajo delovanje in zmogljivost oblačne aplikacije, podrobneje teste povezane z analizatorjem, tečejo na oblačni storitvi Cloud9 [3], ki je del amazonovega oblaka AWS (Amazon Web Services). Ta ponuja virtualko z operacijskim sistemom **Linux**, kot rešitev programiranja v oblaku. Specifikacije virtualke na kateri tečejo klienti, ki jih je možno dobiti, so sledeče:

- velikost pomnilnika (RAM) 1 GB,
- velikost diska (HDD) 5 GB,
- število procesorjev 1.



Slika 4.6: Graf pingov oblačne storitve Cloud9 na našo oblačno storitev.

Slika 4.6 prikazuje rezultate testiranja spletne povezave med klienti stacioniranimi na oblačni storitvi Cloud9 in našo oblačno storitvijo, testiranje je potekalo kot to prikazuje slika 4.5. Samo pošiljanje pingov je potekalo med ponedeljkom 16.4.2018 od 19:00:00 (+0), do torka 17.4.2018 19:00:00 (+0), ping pa se je vršil vsako minuto. Tako dobimo rezultate, ki nakazujejo, da je povezava med klienti in oblačno storitvijo, sicer ne zelo hitra (v povprečju je čas malce nad 82ms) ampak dokaj konstantna, kar je razvidno tudi iz grafa. Prav tako ne zaznamo nobenih globalnih trendov iz rezultatov, ki bi nakazovali da se tekom nekega dela dneva hitrost povezave spremeni.

Želimo izvedeti tudi kaj več o sami poti med klienti, torej oblačno storitvijo Cloud9 in našo oblačno storitvijo, vendar je nažalost možnost uporabe ukaza **traceroute** na storitvi Cloud9 izklopljena. Zato analiziramo pot med domačim računalnikom, ki se nahaja v območju mesta Kamnik v omrežju ponudnika **Amis**, ter obema oblačnima storitvima.

```
Tracing route to ois-v1-zbabnik.c9users.io [35.195.221.140]
over a maximum of 30 hops:

  1    72 ms    99 ms    99 ms  192.168.1.254
  2    36 ms    38 ms    41 ms  asr-lj.amis.net [212.18.32.174]
  3    42 ms    35 ms    35 ms  mx-lj1-te-0-3-0.amis.net [212.18.44.161]
  4    18 ms    18 ms    17 ms  109.239.185.9
  5    17 ms    22 ms    17 ms  195.3.102.58
  6     *       56 ms    27 ms  195.3.102.57
  7     *        *        *     Request timed out.
  8     *        *       74 ms  lg1-1171.as8447.a1.net [195.3.64.2]
  9    47 ms   117 ms    65 ms  72.14.202.36
 10   101 ms   104 ms    98 ms  108.170.252.83
 11    63 ms    71 ms    81 ms  209.85.240.113
 12    77 ms    74 ms    84 ms  66.249.95.226
 13    92 ms    90 ms    89 ms  216.239.56.179
 14    43 ms    42 ms    43 ms  216.239.50.153
 15     *        *        *     Request timed out.
 16     *        *        *     Request timed out.
 17     *        *        *     Request timed out.
 18     *        *        *     Request timed out.
 19     *        *        *     Request timed out.
 20     *        *        *     Request timed out.
 21     *        *        *     Request timed out.
 22     *        *        *     Request timed out.
 23     *        *        *     Request timed out.
 24    42 ms    41 ms    41 ms  140.221.195.35.bc.googleusercontent.com [35.195.221.140]

Trace complete.
```

Slika 4.7: Rezultat analize omrežne poti med domačim računalnikom in oblačno storitvijo Cloud9.

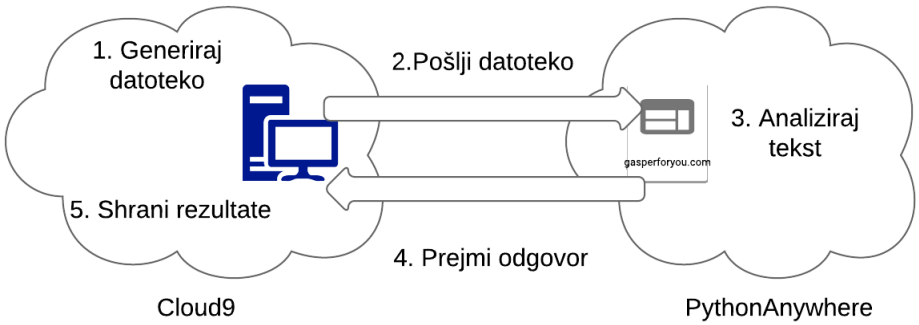
```
Tracing route to gasperforyou.pythonanywhere.com [34.206.101.184]
over a maximum of 30 hops:

  1     4 ms    99 ms    99 ms  192.168.1.254
  2    17 ms    17 ms    18 ms  asr-lj.amis.net [212.18.32.174]
  3    17 ms    35 ms    18 ms  mx-lj1-te-4-1-1.amis.net [212.18.44.209]
  4    18 ms    17 ms    17 ms  109.239.185.9
  5    18 ms    18 ms    17 ms  195.3.102.58
  6    30 ms    23 ms    23 ms  195.3.102.57
  7     *        *        *     Request timed out.
  8    24 ms    24 ms    24 ms  lg4-9072.as8447.a1.net [195.3.64.142]
  9    24 ms    25 ms    24 ms  ae52.bar2.vienna1.level3.net [212.73.202.225]
 10     *        *        *     Request timed out.
 11   121 ms   122 ms   122 ms  amazon.com.edge1.washington1.level3.net [4.28.125.110]
 12     *        *        *     Request timed out.
 13     *        *        *     Request timed out.
 14     *        *        *     Request timed out.
 15     *        *        *     Request timed out.
 16     *        *        *     Request timed out.
 17     *        *        *     Request timed out.
 18     *        *        *     Request timed out.
 19     *        *        *     Request timed out.
 20     *        *        *     Request timed out.
 21     *        *        *     Request timed out.
 22     *        *        *     Request timed out.
 23   124 ms   122 ms   122 ms  ec2-34-206-101-184.compute-1.amazonaws.com [34.206.101.184]

Trace complete.
```

Slika 4.8: Rezultat analize omrežne poti med domačim računalnikom in našo oblačno storitvijo.

Sliki 4.7 in 4.8 prikazujeta rezultate pridobljene, z ukazom **tracert**. Ta v korakih vrne vse skoke, ki jih paketi naredijo na omrežni poti, samo hitrost povezave na posameznem skoku ter samo lokacijo oziroma njeno IP številko. Podatki, ki so za nas zanimivi se začnejo po osmem skoku, ko paketi ne potujejo več po omrežju do ponudnika storitev **Amis** in nadrejenega ponudnika **A1**. Na

sliki 4.7 lahko opazimo, da je vseh skokov štiriindvajset, zadnji se nahaja na **Googlovem** Proxy strežniku, ki se verjetno nahaja v Evropi. Na sliki 4.8 lahko opazimo, da je vseh skokov triindvajset, kar je sicer primerljivo s številom skokov v primeru analize storitve Cloud9, vendar je tu zadnji skok pozicioniran na **Amazonovih** strežnikih, postaljvenih na vzhodni obali Združenih Držav Amerike.

4.6 Rezultati meritev - analizator

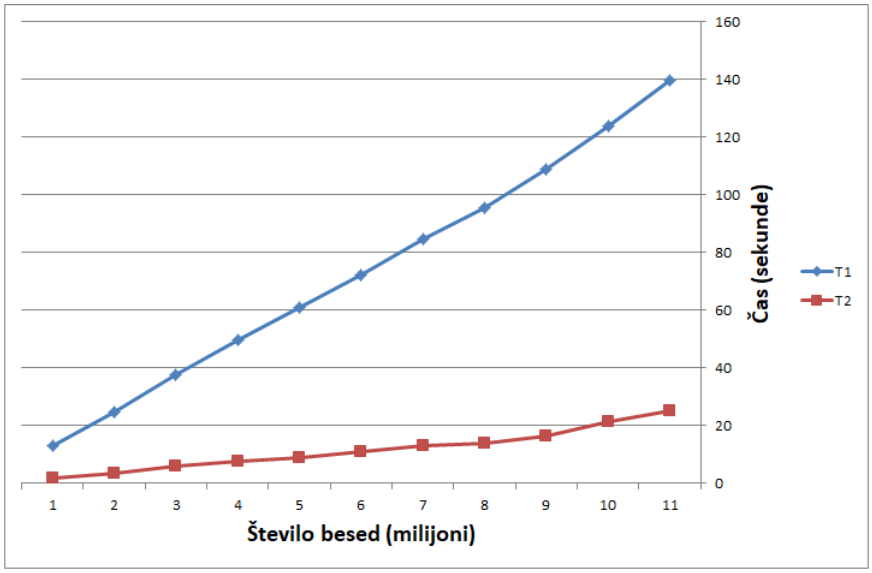


Slika 4.9: Slika ponazarja potek testiranja aplikacije z vidika analizatorja.

Vse meritve v tem razdelku potekajo po principu nakazanem na sliki 4.9.

4.6.1 Odvisnost med številom besed v datoteki in merjenim časom

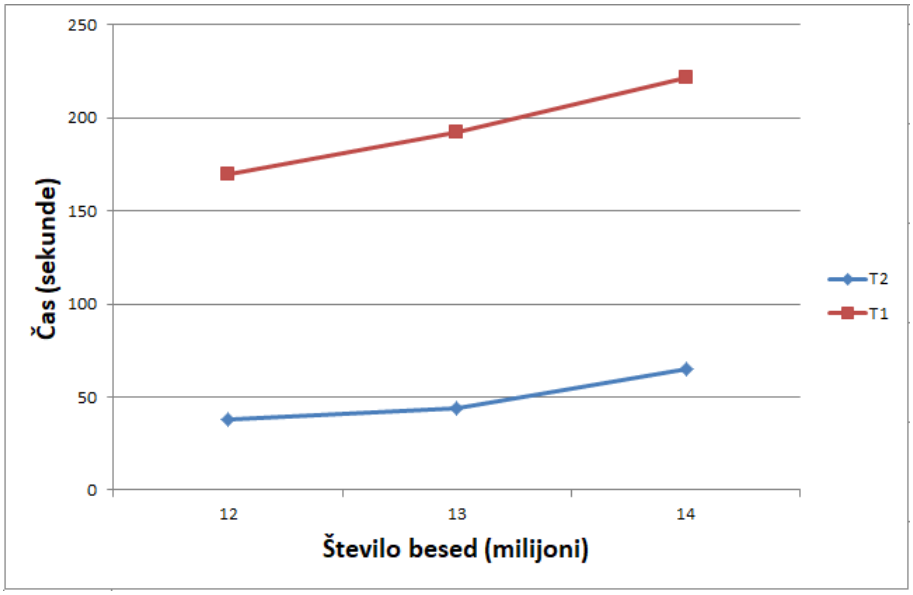
Oblačno storitev želimo preizkusiti z vidika strojne opreme. Konkretno nas v tem primeru zanima, kako se bo storitev odzvala, če pošiljamo vse večje tekstovne datoteke proti oblačni storitvi. Tako testiramo pomnilniške zmožnosti in v manjši meri tudi procesne zmožnosti storitve. Konkretno uporabljamo tekstovne datoteke, ki vsebujejo med enim in enajstimi milijoni besed, pri čemer je dolžina vseh besed enaka pet znakov, kar nam omogoča generiranje ravno malo več kot enajst milijonov unikatnih besed. Upamo, da bo zgornja meja enajst milijonov besed že dovolj za težjo obremenitev storitve. V nasprotnem primeru potrebujemo nov sistem generiranja besed.



Slika 4.10: Graf odvisnosti časov T_1 in T_2 od števila besed.

Vse vrednosti na sliki 4.10 so izračunane kot povprečje tridesetih meritev. Meritve se vršijo na klientu pozicioniranem na oblačni storitvi Cloud9, 9.4.2018 od 18:25:00 (+0), do istega dne 18:57:00 (+0).

Slika 4.10 prikazuje odvisnost celotnega časa pošiljanja T_1 in časa analize T_2 od števila besed v poslanih datotekah. Na žalost nismo opazili hudega porasta pri trenutnem povečevanju besed. To nakazuje, da moramo to število še drastično povečati, da bomo oblačno storitev dovolj obremenili do točke, ko bo začela odpovedovati. Ker smo v razdelku 4.6.2 ugotovili, da število unikatnih besed na čas pošiljanja T_1 ali analize T_2 nima vpliva, bomo to ugotovitev uporabili pri generiranju večjih datotek. Tako generiramo niz sestavljen z enajstimi milijoni unikatnih besed, nakoncu pa do potrebnega števila besed v niz dodajamo besede *aaaaa*. Ta pristop nam v praksi omogoča generiranje datotek s poljubno velikim številom besed.



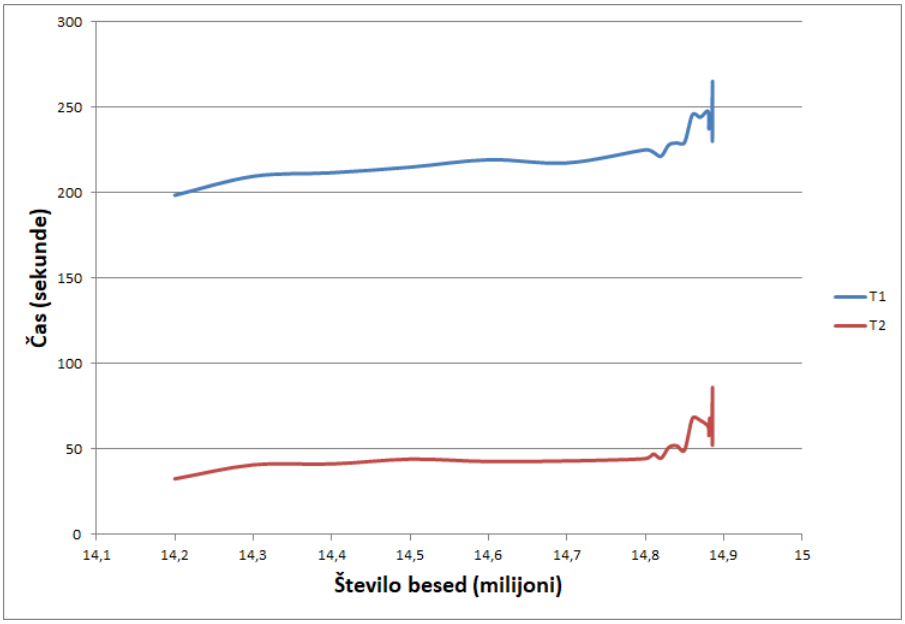
Slika 4.11: Graf odvisnosti časov T_1 in T_2 od števila besed.

Vse vrednosti na sliki 4.11 so izračunane kot povprečje desetih meritev. Meritve se vršijo na klientu pozicioniranem na oblačni storitvi Cloud9, 20.4.2018 od 13:51:00 (+0), do istega dne 15:11:00 (+0).

Slika 4.11 prikazuje gibanje časov T_1 in T_2, v odvisnosti od števila besed v datoteki, ko število besed še nadaljno povečujemo na vsakem koraku za milijon besed. Pri petnajstih milijonih besed v poslani datoteki aplikacija ne vrne več odgovora. Velika deviacija rezultatov je vidna tudi pri trinajstih in štirinajstih milijonih besed, kjer postane čas analize zelo šumen. Za boljšo predstavo velikosti omenjenih datotek lahko povemo, da so posamezne datoteke velike skoraj 100 MB.

Podrobno nas zanima stanje med štirinajstimi in petnajstimi milijoni besed, zato imamo pripravljeno strategijo pošiljanja, ki nam omogoča natančno analizo točke odpovedi. Predpostavimo, da je število besed v datoteki štirinajst milijonov in da se v vsakem koraku to število poveča za milijon besed. Torej v naslednjem koraku pošiljamo datoteko, ki vsebuje petnajst milijonov besed. Recimo, da v tej točki (petnajst milijonov besed v datoteki) aplikacija odpove. Ker klient od aplikacije nebo dobil odgovora, se vrne na prejšnjo točko, kjer je pošiljal štirinajst milijonov besed v datoteki, to stori tako, da od števila besed pri kateri je aplikacija odpovedala, odšteje število besed za katero povečujemo število besed v datoteki. Nato klient zmanjša samo število za katerega povečujemo število besed v datoteki, v našem primeru smo omenili, da je to število enako milijonu besed, in sicer to število zmanjša za faktor deset. Tako bo število besed s katerim povečujemo datoteke enako stotisočim besedam. Sedaj lahko pošiljanje nadaljujemo, tako da štirinajstim milijonom besedam prištejemo novo

število povečevanja in tako dobimo datoteko s štirinajstimi milijoni besed. Meritve lahko zaključimo, ko bo število povečevanja besed v datoteki enak nič, saj imamo takrat že do besede natančno določeno točko odpovedi.



Slika 4.12: Graf odvisnosti časov T_1 in T_2 od števila besed.

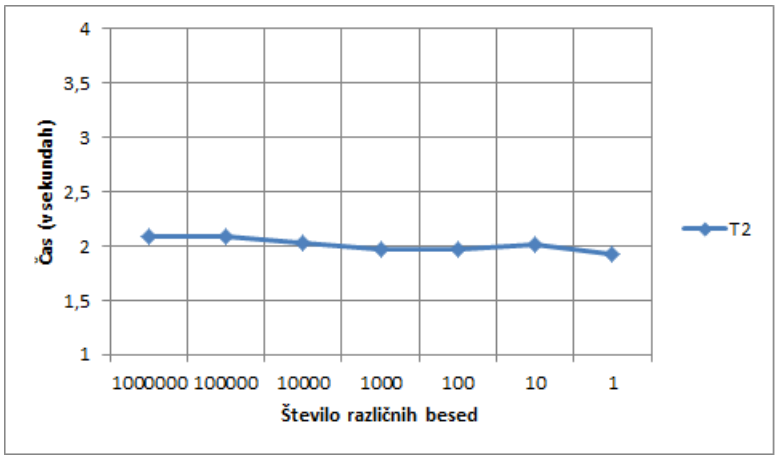
Vse vrednosti na sliki 4.12 so izračunane kot povprečje dvajsteih meritev. Meritve se vršijo na klientu pozicioniranem na oblačni storitvi Cloud9. Meritve so bile izvedene 26.4.2018 med 6:27:00 (+0) in 19:13:00 (+0), 27.4.2018 med 5:23:00 (+0) in 21:40:00 (+0), 28.4.2018 med 7:10:00 in 23:07:00 ter 29.4.2018 med 9:50:00 (+0) in 17:04:00 (+0).

Slika 4.12 prikazuje gibanje časov T_1 in T_2, med štirinajstimi in petnajstimi milijoni besed, kar je za našo analizo najbolj zanimivo področje saj se v tem območju nahaja točka odpovedi. S pomočjo daljšega pošiljanja smo uspeli to točko natančno definirati in sicer aplikacija odpove pri natanko štirinajst milijonov osemstopetdeset tisoč sedemsto devetinpetdeset besedah. Zanimivo je tudi gibanje časov v območju od štirinajst milijonov osemsto tisoč besed dalje. Tu lahko opazimo zelo velik porast časa T_2. To se odraža tudi na porastu časa T_1.

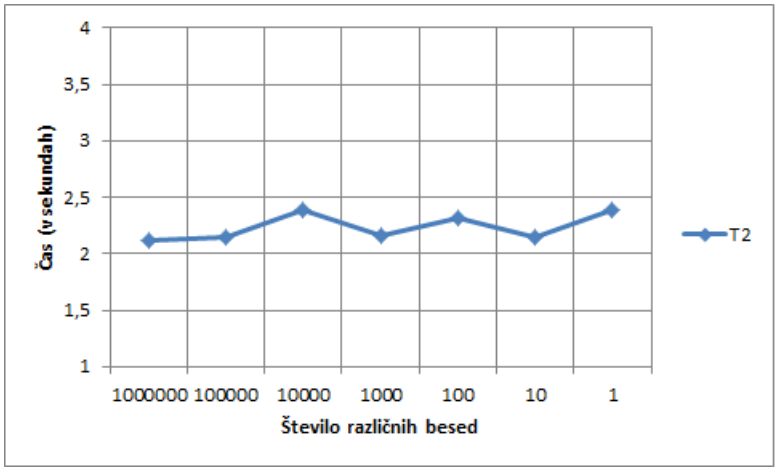
4.6.2 Odvisnost med številom unikatnih besed v datoteki in merjenimi časi

Zanima nas, kako se oblačna storitev odziva na datoteke, ki sicer vsebujejo enako število besed, vendar različno število duplikatov, zato imamo pripravljena dva tipa tekstovnih generatorjev, ki sta si sicer zelo podobna, razlikujeta pa se samo v tem, da prvi generator zapisuje vse enake besede v datoteki skupaj eno za

drugo; tako dobimo dolge verige enakih besed dolžin med eno in milijon besedami, medtem ko drugi generator zapisuje v verige tako rekoč unikatne besede, kjer se vsaka pojavi le enkrat; tako dobimo dolgo zaporedje verig, ki vsebujejo med eno in milijon unikatnih besed. Končne datoteke, ki jih generiramo, vse vsebujejo milijon besed, imamo pa za vsak generator posebej šest različnih datotek, ki vsebujejo med eno, deset, tisoč in tako dalje unikatnih besed.



Slika 4.13: Graf odvisnosti časa od števila unikatnih besed, za prvi generator



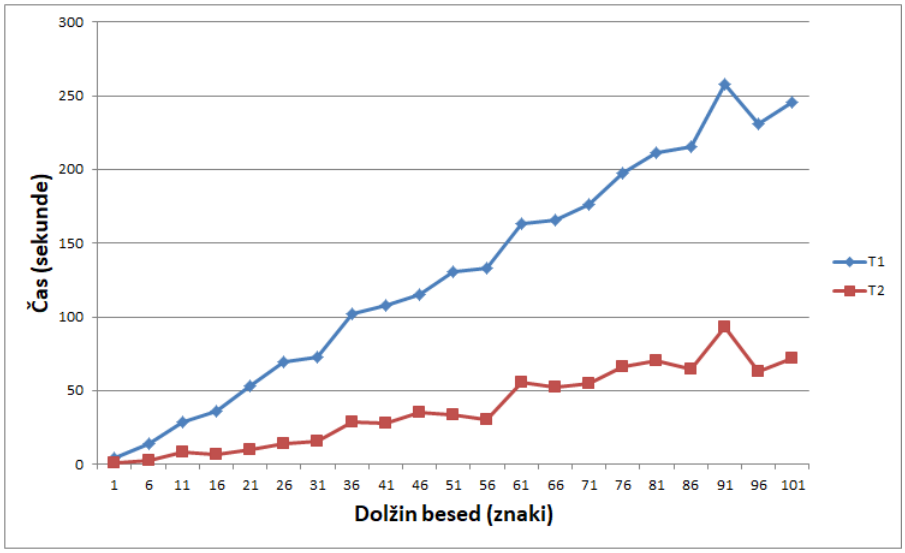
Slika 4.14: Graf odvisnosti časa od števila unikatnih besed, za drugi generator

Vse vrednosti na slikah 4.13 in 4.14 so izračunane kot povprečje šestdesetih meritev. Meritve se vršijo na klientu pozicioniranem na oblačni storitvi Cloud9, za prvi generator 11.4.2018 od 16:12:00 (+0), do istega dne 17:38:00 (+0), za drugi generator pa 13.4.2018 od 13:17:00 (+0), do istega dne 14:42:00 (+0).

Iz slik 4.13 in 4.14 sklepamo, da število unikatnih besed v tekstu ne vpliva, oziroma vpliva minimalno na čas analize. Večino manjših sprememb na obeh grafih je možno pripisati zunanjim vplivom. Tako rekoč je edini pravilen sklep, da čas procesiranja ni odvisen od števila unikatnih besed v tekstu.

4.6.3 Odvisnost med dolžino besed v datoteki in merjenimi časi

Zanima nas, kako se oblačna storitev odziva na datoteke, ki sicer vsebujejo enako število besed, vendar je dolžina besed v datotekah različna. Vnaprej smo določili, da pošiljamo datoteke z natanko milijonom besed, spreminjamo pa njihovo dolžino šteto v znakih (beseda "aaaaa"vsebuje pet zankov). Tako začnemo s pošiljanjem datoteke, ki vsebuje besede dolge natanko en znak, vsaka nadaljno poslana datoteka pa vsebuje besede, ki so za pet znakov daljše od besed, iz datoteke v prejšnji iteraciji pošiljanja. Torej v drugi iteraciji generiramo datoteko, ki vsebuje besede dolge šest znakov, za pet znakov daljše od besed v prvi iteraciji. V tretji iteraciji generiramo datoteko, ki vsebuje besede dolge enajst znakov, s tem postopkom nadaljujemo dokler ne odpove aplikacija.

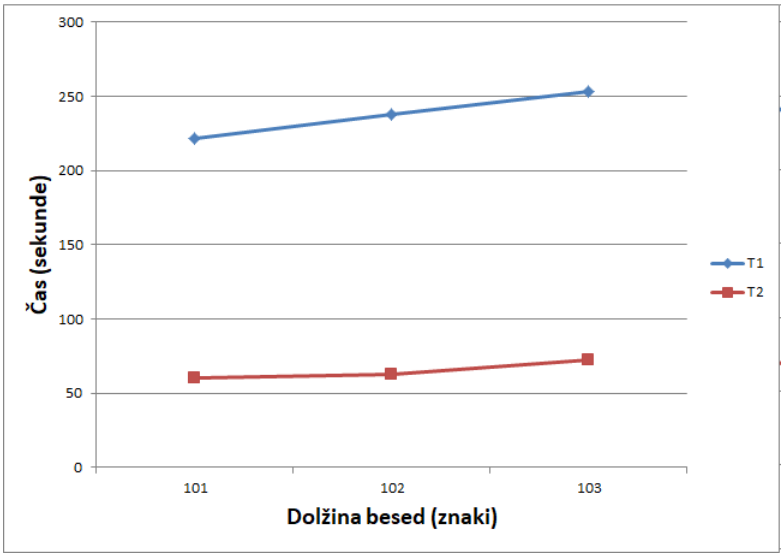


Slika 4.15: Graf odvisnosti časa od dolžine posameznih besed.

Vse vrednosti na sliki 4.15 so izračunane kot povprečje desetih meritev, meritve se vršijo na klientu pozicioniranem na oblačni storitvi Cloud9. Meritve so bile izvedene 29.4.2018, med 19:31:00 (+0) in 22:08:00 (+0) ter 30.4.2018, med 6:26:00 (+0) in 11:40:00 (+0).

Slika 4.15 prikazuje odvisnost med časom pošiljanja T_1 in časom analize T_2 ter dolžino besed v datotekah. Rezultati so občasno dokaj šumni. Kot primer lahko vzamemo rezultat za dolžino besede enaindevetdeset znakov. Šum lahko

pripišemo dokaj majhnemu številu ponovitev meritve za isto dolžino besed, ki je v tem primeru enak desetim ponovitvam. Navkljub šumnim podatkov lahko globalno vidimo dokaj lepo linearno rast časov T_1 in T_2. Aplikacija se ne odziva več pri dolžini besed stošest znakov, naš cilj je podrobneje raziskati kaj se dogaja v intervalu od stoenega do stošestih znakov.



Slika 4.16: Graf odvisnosti časa od dolžine posameznih besed.

Vse vrednosti na sliki 4.16 so izračunane kot povprečje desetih meritev. Meritve se vršijo na klientu pozicioniranem na oblačni storitvi Cloud9. Meritve so bile izvedene 30.4.2018, od 9:43:00 (+0) do 12:45:00 (+0).

Slika 4.16 prikazuje odvisnost časa pošiljanja T_1 in časa analize T_2 v odvisnosti od dolžine besed v poslani datoteki, natančneje za interval dolžine besed od stoenega do stošestih znakov, pri kateri je pri prejšnjih meritvah aplikacija že odpovedala. Iz dobljenih razultatov ugotovimo, da aplikacija odpove še pred mejo stošestih znakov in sicer pri dolžini besede dolge stoštiri znake. Tako smo dobili dolžino besede pri kateri aplikacija odpove, vendar ker posamezna datoteka vsebuje kar milijon besed, ta meja ni natančno določena. Za natančno določitev bi bilo potrebno preplesti merive iz podpoglavja 4.6.1, kjer je govora o vplivu števila besed na čas T_1 in T_2 in v meritve vnesti še različno število besed v poslanih datotekah.

4.7 Načrt dela za naslednji teden

Izvedba meritev na realnih datotekah (knjigah s spletne strani Project Gutenberg). Analiza meritev podatkovne baze.

4.8 Zaključek

V tem poglavju smo izvedli analizo oblačne storitve Pythonanywhere, ki ponuja gostovanje Pythonovih aplikacij v oblaku. Storitev teče v ozadju na Amazaon EC2 brezplačno. Za analizo smo si izbrali aplikacijo analiziranja besed, ki prešteje frekvence na oblak poslanih besedil. Za breme smo si izbrali knjige iz spletne strani Project Gutenberg. Za poglobitev analize pa smo uporabili studi sintetična bremena. Glavne meritve, ki smo jih opravljali so bili časi, ki so potrebni za določeno opravilo. Merili smo čas ki ga potrebuje analizator na strežniku za analizo besedil. In pa čas od pošiljanja zahtevka do prejemanja odgovora. S pomočjo teh dveh časov smo izračunali še čas, ki je potreben, da zahteva in odgovor prepotujeta skozi omrežje. Poleg analiziranja besedil smo rezultate in besedila tudi shranili na strežniku. Izmerili smo tudi čas potreben za shranjevanje in poizvedbo rezultatov. Zradi praktičnosti smo bremena pošiljali iz storitve cloud9, ki teče na oblačni storitvi AWS. Preverili smo omrežje iz pošiljanja paketov na cloud9 in potem še iz clod9 na Pythonanywhere. Nato pa smo analizirali še analizator z različnih vidikov. Najprej smo preverili, kako se storitev obnaša če povečujemo število poslanih besed. Sprva smo testirali velikost boesedil do 11 miljonov besed, ki pa žal niso obremenile storitve. Kasneje smo točko odpovedi bolj natančno določili na med štirinajt in petnajstimi miljoni besed. Nato smo testirali točke odpovedi od števila unikatnih besed. Zaključili smo da število unikatnih besed ne vpliva pomembno na velikost merjenih časov. Ugotovili smo, da se časi povečujejo linearno, kar je posebej zanimivo. Točke odpovedi nismo natančno določili, ker bi bilo potrebno rezultate primerjati s prvim aspektom analize.

Literatura

[1] PythonAnywhere. https://www.pythonanywhere.com/

[2] Project Gutenberg. http://www.gutenberg.org/

[3] AWS - Cloud9. https://aws.amazon.com/cloud9/?origin=c9io

[4] https://www.researchgate.net/profile/Steven_Stemler/publication/269037805_An_Overview_of_Cont

[5] https://www.sciencedirect.com/science/article/pii/S001999586790201X

[6] http://www.textarc.org/appearances/InfoVis02/InfoVis02_TextArc.pdf