Analiza zmogljivosti oblačnih in strežniških storitev

Uredil prof. dr. Miha Mraz

Maj 2016

Kazalo

Predgovor

Pričujoče delo je razdeljeno v devet poglavij, ki predstavljajo različne analize zmogljivosti nekaterih oblačnih izvedenk računalniških sistemov in njihovih storitev. Avtorji posameznih poglavij so slušatelji predmeta *Zanesljivost in zmogljivost računalniških sistemov*, ki se je v štud.letu 2015/2016 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 2016

Poglavje 1

Analiza zmogljivosti odprtokodne storitve Icecast2

Matic Tkalec, Klavdij Starman

1.1 Opis problema

Cilj poglavja je testiranje zmogljivosti 'open source' storitve Icecast2, ki omogoča 'streaming' glasbe ali videa iz oblaka na skoraj katero koli napravo. Najin namen je bil izvedeti, kje je meja pri obremenjenosti aplikacije glede na dane zmogljivosti strežnika, kjer je oblačna storitev nameščena.

1.2 Rešitev problema

Testni sistem je nameščen v Frankfurtu. Storitev sva testirala z različnimi bremeni ter na ta način prišla do ustreznih zaključkov.

Za gostitelja sva si izbrala ponudnika DigitalOcean, specifikacije strežnika pa so sledeče:

- Ubuntu 14.04,
- Intel(R) Xeon(R) CPU E5-2650L v3 @ 1.80GHz,
- 1GB RAM,
- 30GB SSD.

Za poganjanje aplikacije oz. 'streama' na gostitelju uprabljava sledečo programsko opremo:

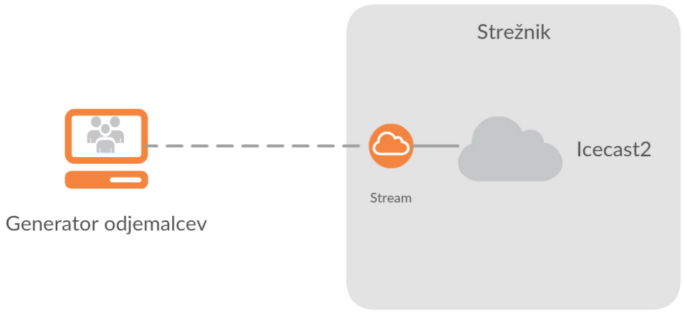
- Apache 2.4.7,
- PHP 5.5.9,
- MySQL 5.5.47.

1.3 Breme

Definirala sva več različnih bremen. Vsa bremena sva definirala z bash skripto, ki uporablja ukaz curl in s tem definira odjemalca, ki posluša naš 'stream'.

1.3.1 Prvo breme

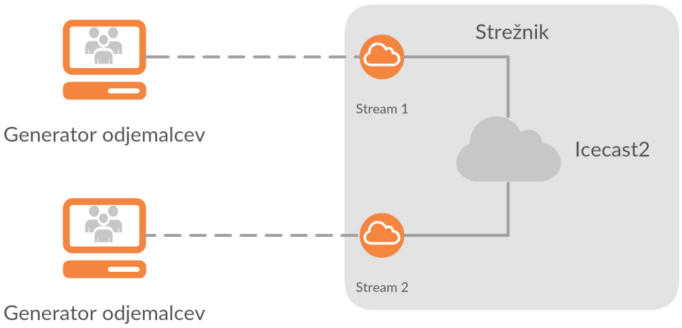
Pri prvem bremenu sva pognala samo eno skripto, ki je vse odjemalce usmerila na isti 'stream'. Vsakih 10 sekund je skripta ustvarila 10 novih odjemalcev in jih povezala na 'stream'. Število odjemalcev sva povečevala, dokler ni zmanjkalo pomnilnika (8GB), in prišla do številke 6000. Čas izvajanja je bil 1h 40min. Ko je zmanjkalo pomnilnika je sistem preprosto opozoril, da je temu tako in naprej deloval nemoteno, če pa je skripta še naprej ustvarjala nove procese, pa je prišlo do popolne zrušitve strežnika, tako da se do njega ni bilo več moč povezati - potreben je bil 'reboot' preko vmesnika ki ga ponuja DigitalOcean.



Slika 1.1: Struktura testiranja pri prvem bremenu.

1.3.2 Drugo breme

Pri drugem bremenu sva pognala 2 različni skripti iz dveh različnih računalnikov, ki sta se povezala na različna streama, ki pa jih je serviral isti strežnik. Skupno število odjemalcev je bilo seveda enako kot pri prvem bremenu, na vsak 'stream' se je povezalo 3000 odjemalcev. Obe skripti sta število odjemalcev povečevali za 10 vsakih 10 sekund, tako kot pri prvem bremenu, zato se je čas izvajanja prepolovil.



Slika 1.2: Struktura testiranja pri drugem bremenu.

1.3.3 Tretje breme

Pri tretjem bremenu sva ustvarila 100 'streamov', pri čemer je bilo na vsak 'stream' povezanih 90 odjemalcev, torej je bilo skupno število odjemalcev 9000. Odjemalcev nisva povezovala postopno, vendar se je vseh 9000 odjemalcev na svoje pripadajoče 'streame' povezalo naenkrat.

1.3.4 Četrto breme

Pri tem bremenu sva pognala skripto, ki je takoj ustvarila veliko število odjemalcev(10.000 in 12.000) in jih povezalo v enem primeru na dva 'streama', v drugem primeru pa na isti 'stream'.

1.4 Metrike

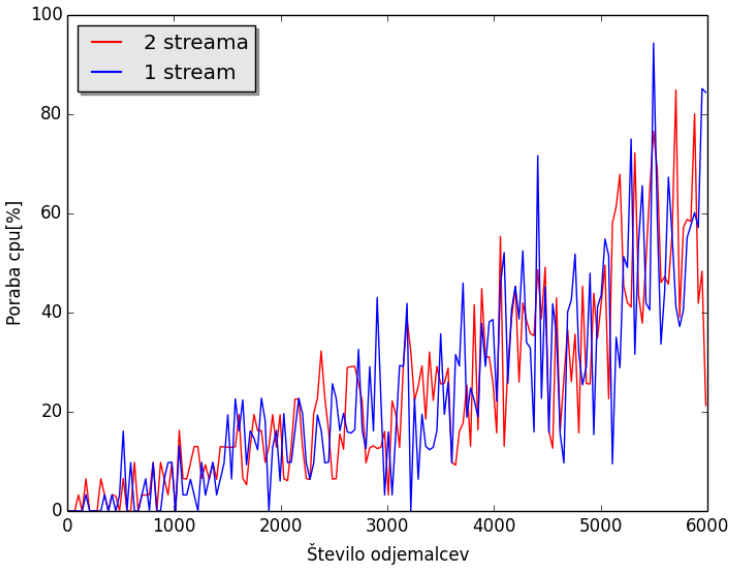
Merila sva obremenjenost centralno procesne enote v odvisnosti od časa oz. števila odjemalcev, merila sva tudi zasedenost pomnilnika RAM, ter s poslušanjem 'streamov' poizkušala preveriti, kako dobra je uporabniška izkušnja (preverjala sva, če je 'stream' neprekinjen).

1.5 Meritve

Za meritve sva uporabila skripto, ki je vsakih 5 sekund zajemala podatke o obremenjenosti procesorja. Podatke sva nato obdelala in predstavila na ustreznih grafih.

1.5.1 Postopno povečevanje odjemalcev

Ugotovila sva, da se obremenjenost procesorja približno linearno povečuje s številom odjemalcev. Merila sva torej obremenjenost centralno procesne enote če imamo samo en 'stream' in če imamo dva streama. Rezultati so združeni in jih lahko vidimo na sliki 1.3. Eksperiment je bil ponovljen dvakrat, rezultati na grafu so povprečeni. Prvi eksperiment se je izvajal v soboto, 30.4.2016 od 14.00 naprej, drugi pa je bil izveden v petek, 13.5.2016 od 22.00 naprej. Iz rezultatov je vidno, da je pri dveh 'streamih' CPU malce bolj obremenjen kot zgolj pri enem. Zasedenost pomnilnika RAM je tako pri prvem kot pri drugem poizkusu skorajda ostala enaka, zasedenost je sicer naraščala, vendar za zanemarljive vrednosti. Uporabniška izkušnja je bila pri obeh eksperimentih ves čas nemotena.

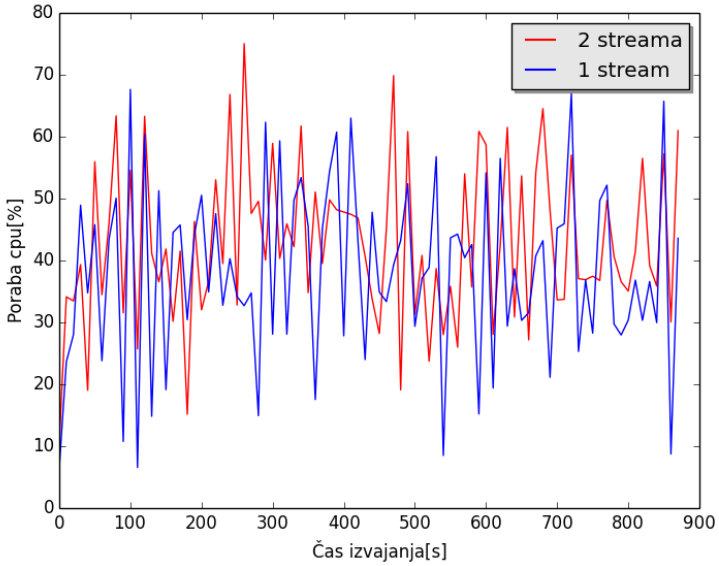


Slika 1.3: Primerjava obremenjenosti CPU pri enem 'streamu' ter pri dveh 'streamih'.

1.5.2 Obremenjenost CPU pri konstantnem številu odjemalcev

Merjenje pri enem in dveh 'streamih' in 6000 odjemalcih

Za to meritev sva vzela primer, ko na začetku povežemo veliko odjemalcev naenkrat (v prvem primeru 6000) na en 'stream' ter potem še po 3.000 odjemalcev na dva različna 'streama', da vidimo kakšna je razlika v obremenjenosti CPU-ja v teh primerih. Pognala sva skripto ter 15 minut izvajala meritev obremenjenosti CPU-ja. Rezultati so vidni na sliki 1.4.

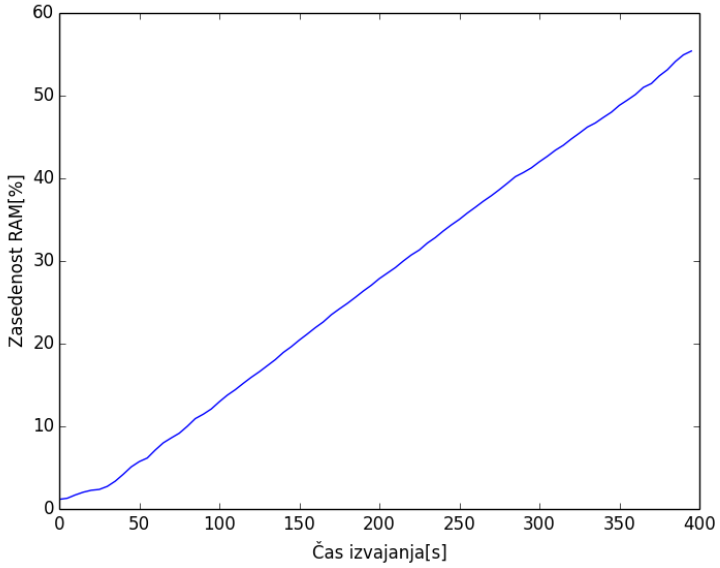


Slika 1.4: Primerjava obremenjenosti CPU med enim 'streamom' in dvema 'streamoma' ter konstantnem številu odjemalcev (6000).

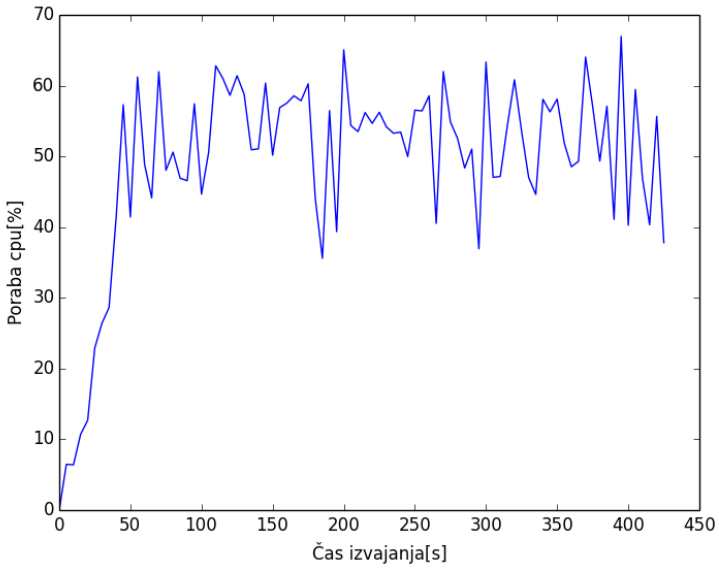
Vidimo, da se rezultati pri enem ali dveh 'streamih' ne razlikujejo preveč. Pri večjem številu odjemalcev na enem 'streamu' je mogoče malce več nihanj, kot pri dveh 'streamih'. Pri povprečni porabi CPU-ja razlike skorajda ni. Pri enem 'streamu' je namreč povprečna poraba v času 900 sekund : 31%, pri dveh 'streamih' pa 33%. Iz tega sledi, da bi pri več streamih in odjemalcih, porazdeljenih na različne streame bila ta razlika tudi večja, kar je potrjeno pri naslednji meritvi. Uporabniška izkušnja je bila pri obeh eksperimentih nemotena, zasedenost pomnilnika RAM je ostala na zelo majhni vrednosti. Eksperimenti so se izvajali v petek, 13.5.2016 od 7.00 naprej, izvedeni so bili trikrat. Rezultati na sliki 1.4 so povprečeni.

Merjenje pri 100 'streamih' z 9000 odjemalci

Za to meritev sva ustvarila 100 različnih 'streamov', ki so servirali različne datoteke, na vsak 'stream' pa se je povezalo 90 odjemalcev. Ravno zaradi ogromnega števila 'streamov' je bila ta meritev zelo zanimiva s pomnilniškega vidika, saj je zasedenost pomnilnika RAM naraščala hitro in linearno, kar je razvidno iz slike 1.5. Ko je zasedenost pomnilnika dosegla v povprečju 54,3%, je operacijski sistem preprosto ubil proces, ki je serviral 'streame', ker je zasedal preveč pomnilniškega prostora. Strežnik je naprej potem deloval nemoteno. Zasedenost centralno procesne enote je bila dokaj konstantna in je v povprečju znašala 49,1%. Gibanje zasedenosti CPU lahko vidimo na sliki 1.6. Eksperiment se je izvajal v četrtek, 12.5.2016 od 10.00 naprej, s trikratno ponovitvijo. Rezultati na slikah 1.5 in 1.6 so povprečeni. Uporabniška izkušnja je bila pri tej meritvi nezadostna, saj je bil 'stream' konstantno prekinjan.



Slika 1.5: Zasedenost pomnilnika RAM pri 100 'streamih' in 9000 odjemalcih.

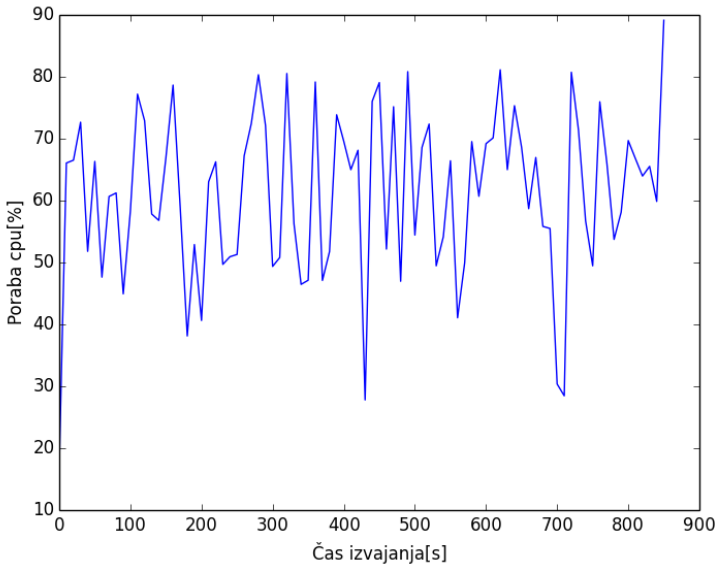


Slika 1.6: Obremenjenost CPU pri 100 'streamih' in 9000 odjemalcih.

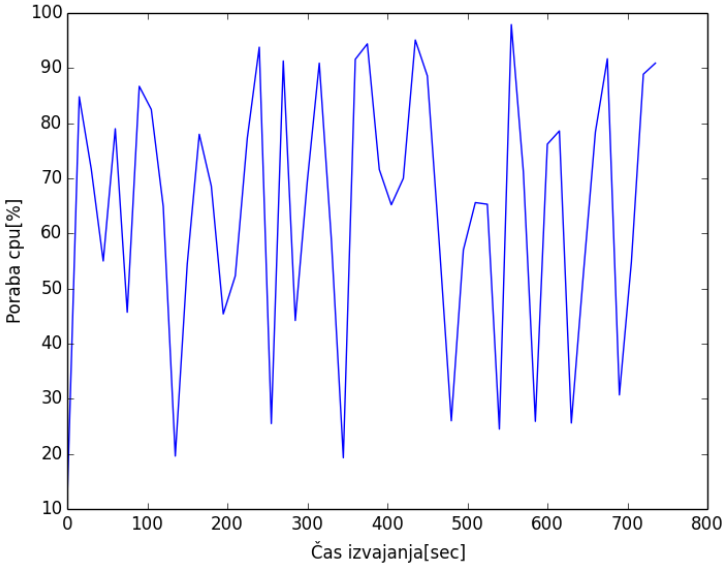
Merjenje pri enem 'streamu' z 10.000 ter 12.000 odjemalci

V nadaljevanju sva naredila še eno meritev in spremenila breme ter pognala skripto za merjenje obremenjenosti, da bi videla pri koliko odjemalcih strežnik dejansko počepne. Meritev sva izvedla z 10 tisoč odjemalci ter ponovila z 12 tisoč odjemalci, ki sva jih poganjala 15 minut ter poslušala 'stream'. Po nekaj sekundah je 'stream' že začel prekinjati, posamezne prekinitve so se pojavljale občasno. Rezultati so vidni na slikah 1.7 in 1.8.

Povprečna poraba je zaradi načina delovanja procesorja in nihanj sicer 48,5% v obdobju petnajstih minut za deset tisoč odjemalcev (slika 1.7). Potem pa sva meritev ponovila še za 12000 odjemalcev. Stream je začel prav tako kot pri prejšnji meritvi prekinjati, vendar so bile prekinitve bolj pogoste. Vidimo, da je obremenjenost CPU-ja na sliki 1.8 večja kot na sliki 1.7. Popvrečna obremenjenost je narasla na 65%. Tudi pri teh dveh meritvah se zasedenost pomnilnika RAM ni veliko spreminjala in je ostala na zelo majhni vrednosti. Zaznala pa sva motnje pri uporabniški izkušnji, ki je bila nezadostna že pri 10 tisoč odjemalcih. Eksperimenti so bili za meritev 1.7 ponovljeni trikrat in sicer prvič 9.5.2016 od 11.00 naprej, naslednjič pa 13.5.2016 od 9.00 naprej. Rezultati na grafu so povprečeni. Meritev na sliki 1.8 sva izvedla zgolj enkrat.



Slika 1.7: Meritev obremenjenost CPU pri enem streamu in 10000 odjemalcih.



Slika 1.8: Meritev obremenjenost CPU pri enem 'streamu' in 12000 odjemalcih.

1.6 Zaključek

Uspelo nama je postaviti sistem, ki deluje zanesljivo, dokler ima na voljo dovolj računalniških virov. Pri velikem številu odjemalcev je problem predvsem zasedenost centralno procesne enote, medtem ko je pri povečevanju števila 'streamov' problem predvsem zasedenost pomnilnika RAM. Zasedenost CPU torej raste približno linearno s povečevanjem števila odjemalcev, zasedenost pomnilnika RAM pa raste linearno s povečevanjem števila 'streamov'. Pri tem je treba poudariti, da morajo 'streami' servirati različne in nikakor ne iste datoteke, če želimo, da ta teza velja. Če želimo uporabnikom zagotoviti dobro uporabniško izkušnjo, lahko pri specifikacijah, kot jih je imel ta strežnik, en 'stream' serviramo največ 9500 odjemalcem.