Analiza zmogljivosti oblačnih in strežniških storitev

Uredil prof. dr. Miha Mraz

Maj 2019

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 2018/2019 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 2019

Poglavje 1

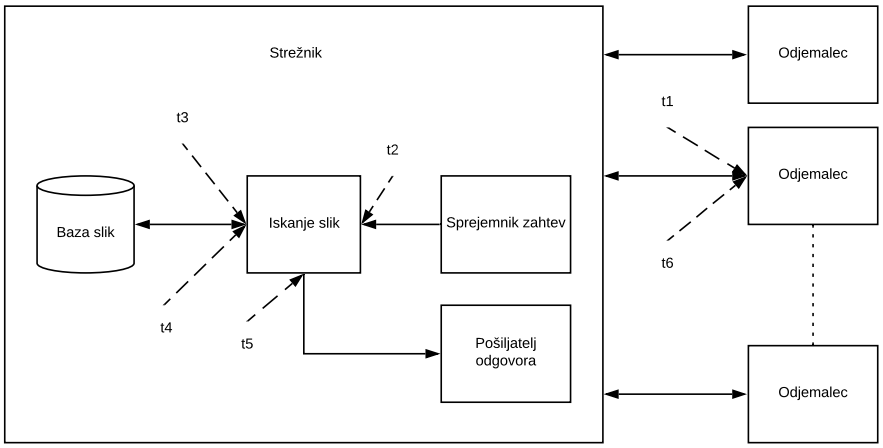
Testiranje zmogljivosti Amazon EC2 platforme

Peter Matičič, Jan Pelicon, Blaž Rojc

1.1 Opis problema

Z dneva v dan proizvedemo čedalje več slik. Predstavljajo znaten delež podatkov, shranjenih v raznih storitvah v oblaku. Ampak ko želimo najti določen predmet ali osebo, ki smo jo slikali, je ročno brskanje po digitalnih zbirkah zamudno.

S tem problemom v mislih bomo stestirali oblačno platformo Amazon EC2. Ustvarili bomo enostavno storitev, ki bo uporabniku omogočala iskanje vzorcev v večjem naboru slik, shranjenih v oblaku. Poglobili se bomo v zahtevnost uporabe za programerja, fleksibilnost pri programiranju in morebitnem prenašanju storitve na druge platforme, odzivnost in izkušnjo za uporabnika ter zmogljivost in skalabilnost virov na platformi.



Slika 1.1: Shema aplikacije.

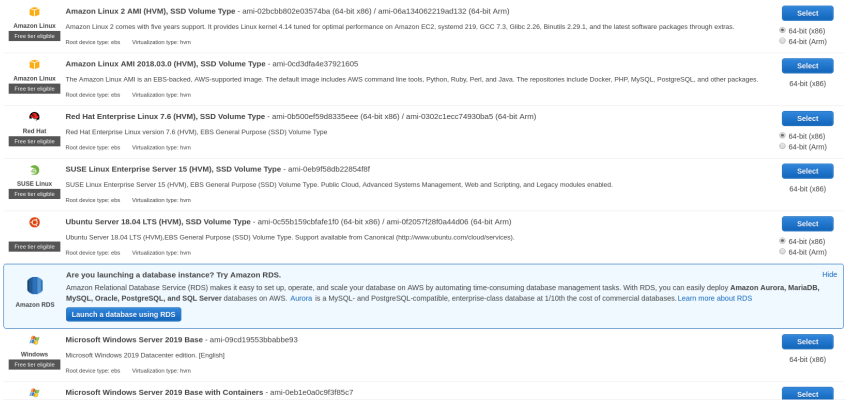
1.2 Realizacija

1.2.1 Način iskanja vzorca

- Iskanje ene slikovne točke v množici slik ki bo ustrezala zahtevam.
- Iskanje vzorca podanega z masko v mnžici slik.
- Iskanje slike, ki vsebuje podan slikovni izsek.

1.2.2 Amazon Web Services (AWS)

Za delo z Amazon Web Services si mora uporabnik ustvariti račun v njihovem sistemu. Po uspešni registraciji si lahko ustvarimo svojo instanco EC2 storitve, ki nam jo Amazon ponuja zastonj za eno leto pri čemer na mesec porabimo največ 750 ur delovanja. Obsežnejša navodila so navoljo tudi na Amaznovi spletni strani [**?**], mi pa smo to naredili tako, da se postavimo v AWS Management Console, kjer lahko izberemo opcijo Launc a virtual machine with EC2. Takoj nam konzola ponudi izbiro slike, katero bi uporabljali na novi virtualki kot prikazano na sliki 1.2. Za naš projekt smo izbrali sliko Amazon Linux 2 AMI c.



Slika 1.2: Slike za izdelavo virtualke

V naslednjem koraku izberemo tip instance slike, to je Amazonov način izbire paketov, ki vključujejo različne funkcionalnosti. V našem primeru ker izbiramo zastonjsko različico nam ponujajo tip t2.micro, ki vsebuje 1 jedro, 1GB pomnilnika, samo začasno hranjenje podatkov na disku in nižjo hitrost povezave. Za tem lahko nadaljujemo z nastavljanjem različnih konfiguracij naše virtualke ali pa preprosto kliknemo Review and Launch, ki nam ponudi še en pregled čez izbrane nastavitve in zažene virtualko. Po zagonu virtualke nam sistem ponudi opcijo generiranja para ključev za varno SSH povezavo nanjo. Ko zaključimo z ustvarjanjem, se premaknemo v EC2 management console kjer kliknemo na instances in od tam lahko opazujemo status naše storitve in pridobimo tudi naslov na katerem se nahaja. Za povezavo uporabimo javni naslov storitve, uporabnika $ec2 - suser$ za varnost pa uporabimo *.pem* datoteko, ki smo jo v prejšnjem koraku prenesli. Ko se uspešno povežemo na storitev lahko pričnemo z razvojem naše aplikacije.

1.2.3 Aplikacija

Aplikacija je napisana v programskem jeziku Java. Sestavljata jo odjemalska in strežniška komponenta, ki uporabljata skupne enumeratorje za določanje tipa zahteve. Zahteve sestavlja odjemalec in jih pošlje strežniku, ki to zahtevo obdela - začne iskanje v slikah in pripravi prvi najden rezultat. Povezava med strežnikom in odjemalcem je trajna, dokler je eden od njiju ne prekine, kar nam omogoči, da pri meritvah ne upoštevamo časa vzpostavitve povezave. Podatki se prenašajo v obliki JSON, ki vsebuje sekvenčno številko zahteve, tip zahteve in morebitne dodatne podatke, ki so potrebni za obdelavo. Strežnik ob prejemu podatkov začne z delom na ustrezni zahtevi ter nato pošlje odgovor klientu z številko zahteve in rezultatom. Strežniški del je napisan tako, da lahko paralelno obdeluje več zahtev.

1.3 Predvidevanje metrik

Kot glavno metriko bomo opazovali skupni čas zahteve in odgovora. Podrobneje ga bomo razdelili na čas potovanja zahteve od nas do strežnika ($t_2 - t_1$), čas obdelave na strežniku ($t_3 - t_2$) in čas potovanja odgovora od strežnika do nas ($t_4 - t_3$) (časi označeni na sliki 1.1.

1.4 Rezultati meritev

1.5 Plan dela

- implementacija iskalnih algoritmov (0 / 3 narejenih)
- določitev bremen
- določitev metrik
- določitev orodij

1.6 Literatura

Literatura

[1] "Digitalocean: Simple cloud computing for developers." https://www.digitalocean.com/, Marec 2017.

[2] G. D. Greenwade, "The Comprehensive Tex Archive Network (CTAN)," *TUGBoat*, vol. 14, no. 3, pp. 342–351, 1993.