

# Analiza zmogljivosti oblačnih in strežniških storitev

Uredil prof. dr. Miha Mraz

Maj 2017



# Kazalo

## Predgovor

iii

<b>1 Analiza zmogljivosti oblačne storitve Heroku (A. Dežman, E.Ljubljankic, M.Vrščaj, A Markežič)</b>	<b>1</b>
1.1 Opis problema . . . . .	1
1.2 Namen . . . . .	2
1.3 Izbira ponudnikov . . . . .	2
1.3.1 Digital Ocean . . . . .	2
1.3.2 Amazon Web Services . . . . .	2
1.3.3 Google Cloud Platform . . . . .	2
1.4 Izbira tehnologij . . . . .	3
1.4.1 PerfKit Benchmarker . . . . .	3
1.4.2 Python . . . . .	3
1.4.3 Odjemalec . . . . .	3
1.5 Izbrane metrike . . . . .	5
1.5.1 Prepustnost omrežja . . . . .	5
1.5.2 Hitrost procesiranja . . . . .	5
1.5.3 Zmogljivost datotečnega sistema . . . . .	5
1.6 Opis bremena . . . . .	5
1.6.1 Iperf . . . . .	5
1.6.2 CoreMark . . . . .	6
1.6.3 FIO . . . . .	6
1.7 Rezultati meritev . . . . .	7
1.7.1 Testiranje prepustnosti omrežja . . . . .	7
1.7.2 Testiranje hitrosti procesiranja . . . . .	9
1.7.3 Testiranje zmogljivosti datotečnega sistema . . . . .	10
1.8 Komentarji rezultatov . . . . .	15



# Predgovor

Pričajoče delo je razdeljeno v deset poglavij, ki predstavljajo analize zmogljenosti nekaterih tipičnih strežniških in oblačnih izvedenih računalniških sistemov in njihovih storitev. Avtorji posameznih poglavij so slušatelji predmeta *Zanesljivost in zmogljenost 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*



# Poglavlje 1

## Analiza zmogljivosti oblačne storitve Heroku

Anže Dežman, Edo Ljubljankic, Maks Vrščaj, Aljaž Markežič

### 1.1 Opis problema

V današnjem času imamo na voljo ogromno oblačnih storitev, ki nam ponujajo okolje za razvijanje in zaganjanje aplikacij, ki so lahko napsiane v različnih jezikih (javascript, ruby, python,...). Oblačnim storitvam takega tipa se reče "okolje kot storitev", oziroma v angleškem izvirniku "Platform as a system- PaaS. Namen seminarske naloge je testirati in analizirati zmogljivosti ene izmed PaaS oblačnih storitev po imenu Heroku.

Heroku se od konkurenco razlikuje po tem da podpira velik izbor programskih jezikov (java, python, php, node.js, scala,...), med katerimi so tudi manj popularni jeziki kot scala ali pa groovy, ki ga večina oblačnih storitev ne podpira. Zanima nas koliko nam oblačna storitev ponuja na področju zmogljivosti, kolikšna je razlika med ogleševano zmogljivostjo in dejansko, ali se bolj splača razvijati aplikacijo lokano ali je ponujeno brezplačna storitev dovolj zmogljiva za naše standarde.

Na vsa ta in mnoga druga vprašnja, ki se nam bodo zastavila ki se nam bodo pojavila tokom testiranje ter kakšne metode testiranja bomo uporabili, bodo opisana v naslednjih poglavjih.

**POGLAVJE 1. ANALIZA ZMOGLJIVOSTI OBLAČNE STORITVE  
2 HEROKU (A. DEŽMAN, E. LJUBLJANKIC, M. VRŠČAJ, A. MARKEŽIČ)**

## **1.2 Namen**

Namen pričajočega poglavja je analizirati zmogljivost in zanesljivost določenih oblačnih storitev na podlagi različnih metrik kot so čas in hitrost dostopa do podatkov, hitrost prenosa datotek, hitrost bralno/pisalnih operacij, hitrost procesiranja zahtev itd.

Na podlagi opravljenih meritev želimo pokazati prednosti in slabosti posameznega ponudnika oblačnih storitev, ki bi končnemu uporabniku pomagali pri izbiri, planirjanju in iskanju anomalij v delovanju storitev.

## **1.3 Izbira ponudnikov**

V pričajočem razdelku so na kratko opisani ponudniki oblačnih storitev, katere smo želeli testirati in primerjati med seboj.

### **1.3.1 Digital Ocean**

Digital Ocean je bila naša prva izbira za testiranje [2]. Ponuja oblačno infrastrukturo, ki bazira na Linuxovih distribucijah kot sta Ubuntu in CentOS. Izkoristili smo naročnino za študente, pri kateri smo za 5\$ dobili 100\$ kredita, ki ga lahko porabimo na storitvah DigitalOcean. Naročnina vsebuje uporabo Linux operacijskih sistemov (Ubuntu, FreeBSD, Fedora, Debian, CoreOS, CentOS), 40GB pomnilnega prostora na SSD disku za hranjenje podatkov, 2GB RAM, 2 centralno procesni enoti na Intel Xeon procesorju ter omejitev prenosa podatkov na 3TB.

### **1.3.2 Amazon Web Services**

Za drugega ponudnika smo izbrali Amazon Web Services [3]. Amazon omogoča enoletno brezplačno uporabo njihovih oblačnih storitev. To vključuje uporabo Linux ali Windows platforme, uporabo shrambnih storitev, podatkovne baze, itd. Pri Amazonu smo izkoristili ponudbo AWS Free Tier, kjer lahko v določenih mejah 12 mesecev uporabljamo naslednje storitve: 750 ur mesečne uporabe Linux operacijskega sistema ali 750 ur mesečne uporabe Windows operacijskega sistema, pri čemer lahko uporabljamo eno instanco cel mesec ali dve instanci, vsako pol meseca. Strojna oprema sestoji iz Intel Xeon procesorja s taktom 3.3 GHz in 1GB RAMa. Za shrambo dobimo 5GB prostora v oblaku, kjer smo omejeni na 20000 GET (zahteve, s katerimi pridobimo določen objekt iz oblačne shrambe) in 2000 PUT zahtev (zahteve, s katerimi določen objekt dodamo v oblačno shrambo).

### **1.3.3 Google Cloud Platform**

Naša zadnja izbira je bila Google Cloud Platform [4]. Omogočena nam je 60 dnevna brezplačna uporaba, poleg tega pa dobimo še 300\$ kredita, ki ga lahko

poljubno porabimo na Google Cloud storitvah. Prav tako ponuja široko ponudbo uporabe programskih jezikov in podatkovnih baz. V naročnini je prav tako vsebovana uporaba Windows in Linux operacijskih sistemov. V preizkusni dobi smo imeli na voljo 8 računalnikov, pri čemer je imel vsak eno centralno procesno enoto na procesorju Intel Xeon serije E5 (2.6 GHz Intel Xeon E5 (Sandy Bridge), 2.5 GHz Intel Xeon E5 v2 (Ivy Bridge) ali 2.3 GHz Intel Xeon E5 v3 (Haswell)) z 3.75GB RAMa.

## 1.4 Izberite tehnologije

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

### 1.4.1 PerfKit Benchmarker

PerfKit Benchmarker je odprtokodno orodje, ki je namenjeno merjenju zmogljivosti in primerjanju oblačnih storitev med seboj [5]. Podpira testiranje množice ponudnikov, med njimi vse naše izbrane kandidate kot tudi Microsoft Azure, Rackspace, OpenStack, itd. Omogoča merjenje hitrosti dostopa do storitev, merjenje latence, prepustnosti, ponuja pa tudi teste procesiranja in meritve bralno pisalnih operacij.

### 1.4.2 Python

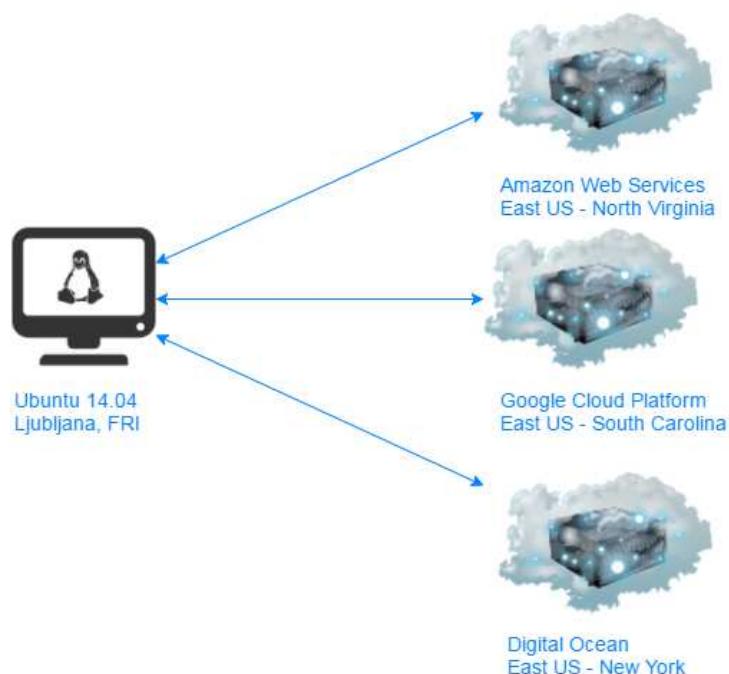
Testno ogrodje, ki sestavlja PerfKit Benchmarker, je spisano v programskem jeziku Python [6]. Ogorodje ponuja veliko število v naprej spisanih standardiziranih testov, kot tudi testov prilagojenih za določene ponudnike. Poleg tega omogoča hitro in enostavno razširljivost, saj je možno preprosto spremenjanje že obstoječih, kot tudi kreiranje lastnih testnih skript.

### 1.4.3 Odjemalec

Odjemaleca predstavlja Ubuntu Server, ki velja za enega najbolj razširjenih Linux distribucij. Testni strežnik je priklopljen na širokopasovno povezano v internet in predstavlja enakopravnega odjemalca za vse tri ponudnike oblačnih storitev. Na njem bo teklo orodje PerfKit Benchmarker, kjer bodo nastavljene povezave do vseh treh ponudnikov, kar zadeva tudi avtentifikacijo in sinhronizacijo s samimi strežniškimi napravami v oblaku.

Na sliki 1.1 je prikazana shema oblačnega sistema. Sestavljena je iz uporabnika, ki dostopa do oblaka, ki nudi storitve glede na njegove zahteve. Na strani uporabnika namestimo PerfKit Benchmarker, s katerim bomo nato izvajali in simulirali zmogljivostne teste za oblak.

**POGLAVJE 1. ANALIZA ZMOGLJIVOSTI OBLAČNE STORITVE  
4 HEROKU (A. DEŽMAN, E. LJUBLJANKIC, M. VRŠČAJ, A. MARKEŽIČ)**



Slika 1.1: Shema testnega sistema.

## 1.5 Izbrane metrike

V pričajočem poglavju so opisane izbrane metrike za merjenje zmogljivosti oblačnega sistema.

### 1.5.1 Prepustnost omrežja

Prepustnost omrežja je eden ključnih dejavnikov današnjega komunikacijskega sveta. Vse več podatkov se prenaša preko omrežja. Potreba po pomnilnih prenosnih medijih, s katerimi bi prenašali podatke, praktično izginja. Tako kot v vsakdanjem življenju, je tudi pri oblačnih storitvah seveda prenos podatkov preko omrežja skorajda edina možnost. Posledično gre za zelo zanimivo metriko s stališča končnih uporabnikov. Vsakogar zanima kako hitro bo lahko prenesel svojega podatke iz točke A do točke B v omrežju.

### 1.5.2 Hitrost procesiranja

Tudi hitrost procesiranja ima velik pomen, saj danes na računalniku ne teče več samo en proces oziroma aplikacija, ampak je teh aplikacij mnogo in se morajo zato boriti za procesorski čas. Končni uporabnik želi vedno imeti občutek tekočega delovanja sistema, kljub temu da hkrati uporablja več aplikacij ali servisov. Zmogljivost procesne enote (CPE) je tako zelo pomembna, saj omogoča hitro procesiranje zahtev in tako možnost delovanja tudi ostalih aplikacij na sistemu. Če bi recimo ena operacija na procesorju porabila preveč časa, bi posledično ostali procesi obstali in bi uporabnik dobil občutek, da sistem ne deluje najbolje.

### 1.5.3 Zmogljivost datotečnega sistema

Zmogljivost datotečnega sistema je izrednega pomena pri prenosu podatkov, bodisi da gre za prenos preko omrežja ali lokalni prenos med različnimi shranjevalnimi mediji. To ne zajema zgolj ročnega kopiranja datotek, ampak tudi vse interne prenose podatkov, ki jih opravljamjo aplikacije. Gre za enega izmed kriterijev, ki je tudi zelo zanimiv za končnega uporabnika, saj predstavlja operacije s katerimi se uporabniki srečujejo dnevno.

## 1.6 Opis bremena

V pričajočem razdelku so na kratko opisana vsa bremena, ki smo jih uporabljali pri testiranju.

### 1.6.1 Iperf

Prepustnost omrežja za promet po protokolarnem skladu TCP/IP lahko enostavno izmerimo z orodjem *Iperf* [7]. *Iperf* deluje kot običajna TCP storitev, in sicer na eni strani povezave je strežnik, na drugi pa odjemalec. Odjemalec

## POGLAVJE 1. ANALIZA ZMOGLJIVOSTI OBLAČNE STORITVE 6 HEROKU (A. DEŽMAN, E. LJUBLJANKIC, M. VRŠČAJ, A. MARKEŽIČ)

generira in pošilja zahteve, ki predstavljajo breme, na drugi strani pa je strežnik v vlogi prejemnika zahtev. *Iperf* izmeri količino prenesenih podatkov in na podlagi znanega trajanja meritve izračuna hitrost prenosa podatkov. Rezultat torej predstavlja hitrost prenosa podatkov v bitih na sekundo (bps). Pri velikih hitrostih povezav in daljših časih potovanja paketov med strežnikom in odjemalcem je zelo pomembna velikost okna v TCP/IP skladu (angl. *TCP buffer size*). V lokalnem ethernet omrežju so časi potovanja paketov zelo kratki, zato običajna velikost okna za TCP povsem zadošča. Priporočena velikost okna je enaka produktu hitrosti povezave in časa potovanja paketa (angl. *round trip time*). Prizeta velikost je 32 KB. Na primer v lokalnem 100 Mb/s omrežju z zakasnitvijo 1 ms zadošča že okno veliko 16 KB.

### 1.6.2 CoreMark

*CoreMark* [8] je preprosto testno orodje za testiranje centralno procesnih enot (CPE). Njegova uporaba je zelo razširjena in ima možnost, da postane del standardne specifikacije, vsaj kar se tiče osnovnega testiranja hitrosti procesiranja. Je neodvisen od procesorske platforme in tako omogoča primerjavo različnih platform med sabo. Rezultat testa predstavlja desetiška vrednost, ki je sestavljena iz rezultatov vseh internih testov, ki jih uporablja *CoreMark*, kar omogoča hitro in učinkovito primerjavo rezultatov. Program je napisan v programskem jeziku C in breme gradi na implementacijah splošno znanih algoritmov: iskanje in urejanje povezanih seznamov (uporaba kazalcev v pomnilniku), osnovne matrične matematične operacije, 'state machine' (za simulacijo različnih protokolov) in CRC (angl. *cyclic redundancy check*, ki se uporablja v množici protokolov predvsem za odkrivanje in popravljanje napak v podatkih). S pomočjo omenjenih bremen nam test prikaže zmogljivost procesorskega sistema iz perspektive zmogljivosti podatkovnih vodil v sistemu, hitrosti pomnilnika in obstoječega predpomnilnika ter hitrosti procesiranja računskih operacij.

### 1.6.3 FIO

*Fio* (angl. *Flexible IO*) predstavlja orodje, ki generira breme, nato pa s pomočjo tega bremena testira in analizira delovanje datotečnega sistema [9]. Gre za lokalni transport podatkov in ne za transport podatkov preko omrežja. *Fio* sestavlja dva dela: prvi del je definiranje bremena, kjer uporabnik definira ali se bo testiralo branje ali pisanje, število poslov, velikost posameznega posla ter velikost posameznega bloka za vhodno/izhodno operacijo. Drugi del predstavlja testiranje. Po uspešnem zaključku testa uporabnik pridobi informacije o hitrosti branja in pisanja, minimalno, maksimalno in povprečno latenco pri branju ali pisanju posameznega bloka, število iops (angl. *input/output operations per second*), itd.

## 1.7 Rezultati meritev

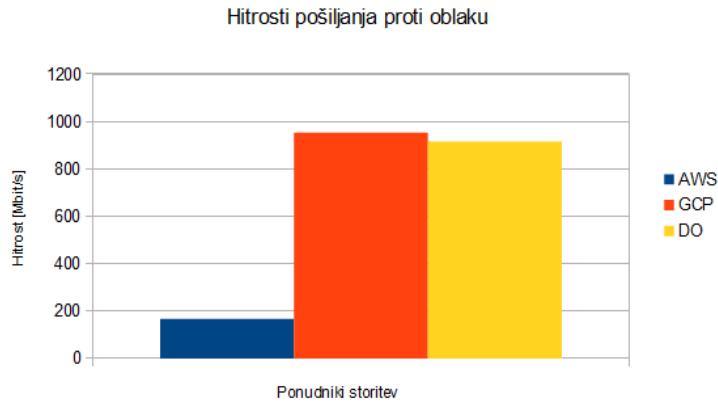
V tem razdelku so opisani postopki izvajanja testov in analize rezultatov, ki so bili pridobljeni po izvedenih testih. Vse teste smo izvajali na operacijskem sistemu Ubuntu 14.04. Teste smo pognali sredi tedna, torej v sredo, izvedli pa smo jih 24-krat - na vsako polno uro. Pri vsakem tipu meritev je tako podana tudi tabela z osnovnimi statističnimi postavkami - minimum, maksimum, povprečje ter standardna deviacija meritev. Pri orodju *Fio* je testov razmeroma veliko - skupno več kot 700 - zatorej smo se odločili, da v statistični analizi predstavimo le dve, za uporabnika morda najpomembnejši metriki - to sta bralna in pisalna hitrost.

### 1.7.1 Testiranje prepustnosti omrežja

Za testiranje prepustnosti omrežja smo uporabili orodje *Iperf*. Breme pri testiranju prepustnosti omrežja predstavljajo zahteve za prenos datotek. Datoteke se pošiljajo od oblaka proti uporabniku in od uporabnika proti oblaku. Na podlagi prenesene količine datotek (v MB) in pretečenega časa izračunamo povprečno hitrost prenosa po omrežju. Virtualne računalnike smo locirali na vzhodni obali ZDA (East US). Pri izvajaju testa smo merili internetno povezavo med entitetami v omrežju. Ena izmed entitet je služila kot oblačna storitev, druga pa kot uporabnik, ki do storitve dostopa. Test smo izvedli obojesmerno (torej od oblaka proti uporabniku in obratno). *Iperf* test se je v vsakem primeru izvajal 60 sekund.

Hitrosti pošiljanja podatkov proti oblaku posameznega ponudnika oblačne storitve so vidne na sliki 1.2. Občutno najmanjšo zmogljivost prenosa smo odkrili pri ponudniku Amazon (AWS), kjer je bila povprečna hitrost prenosa pod 200 Mbit/s. Največjo povprečno hitrost prenosa smo izmerili pri ponudniku Google (GCP), ki je le za odtenek premagal rešitev Digital Ocean (DO). Obe vrednosti presegata hitrost 900 MBit/s in GCP se zelo približa meji 1 Gbit/s. Pri GCP, kot tudi pri DO se je tako izkazalo, da imata več kot 4x večjo povprečno hitrost nalaganja datotek v oblak, v primerjavi s ponudnikom AWS.

POGLAVJE 1. ANALIZA ZMOGLJIVOSTI OBLAČNE STORITVE  
8 HEROKU (A. DEŽMAN, E. LJUBLJANKIC, M. VRŠČAJ, A. MARKEŽIČ)



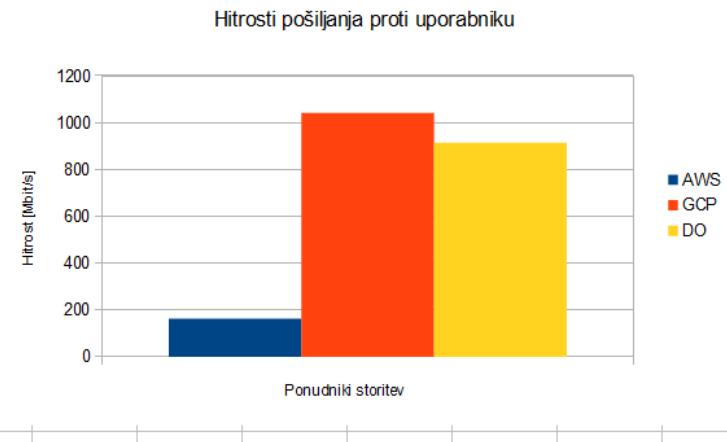
Slika 1.2: Povprečna hitrost pri prenosu podatkov proti oblaku (angl. *upload speed*).

Na sliki 1.3 so razvidni osnovni statistični podatki za meritve *Iperf - upload*. Do največjih odstopanj med minimalno in maksimalno vrednostjo je prišlo pri ponudniku AWS, čeprav ima daleč najslabši povprečni rezultat. Najbolj stabilno in povprečno tudi najboljše se je odrezal GCP.

<i>Enota: MB/s</i>	<b>AWS</b>	<b>DO</b>	<b>GCP</b>
<b>MIN</b>	118	440	926
<b>MAX</b>	914	938	1201
<b>AVG</b>	255	869	1035
<b>STDEV</b>	226	125	79

Slika 1.3: Statistična analiza rezultatov *Iperf - upload*.

Hitrosti pošiljanja podatkov proti uporabniku so vidne na sliki 1.4. Zgodba je skoraj da identična, kot pri hitrostih prenosa v obratni smeri. Občutno najhitrejša sta zopet GCP in DO, daleč zadaj pa je ponovno AWS. Pri primerjavi hitrosti se je izkazalo, da je razmerje zmogljivosti praktično enako, kot v prejšnjem primeru.



Slika 1.4: Povprečna hitrost pri prenosu podatkov proti uporabniku (angl. *download speed*).

Tudi statistična analiza *Iperf - download* testa, vidna na sliki 1.5, prikazuje podobne rezultate. Do največje standardne deviacije rezultatov je prišlo pri ponudniku AWS. DO in GCP pa sta pokazala precej boljše in bolj stabilne rezultate.

<i>Enota: MB/s</i>	<b>AWS</b>	<b>DO</b>	<b>GCP</b>
<b>MIN</b>	<b>113</b>	<b>473</b>	<b>598</b>
<b>MAX</b>	<b>913</b>	<b>938</b>	<b>1236</b>
<b>AVG</b>	<b>281</b>	<b>888</b>	<b>972</b>
<b>STDEV</b>	<b>229</b>	<b>95</b>	<b>122</b>

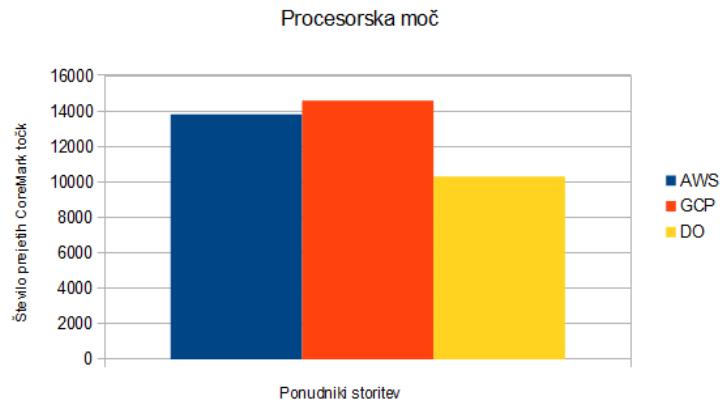
Slika 1.5: Statistična analiza rezultatov *Iperf - download*.

### 1.7.2 Testiranje hitrosti procesiranja

Procesorsko moč smo merili z orodjem *CoreMark*. Breme pri testiranju hitrosti procesiranja predstavljajo splošni algoritmi kot so npr. iskanje in urejanje po seznamih, osnovne matrične operacije ter izvajanje CRC. Ker pri nobenem ponudniku nismo imeli izbiro, kateri procesor bomo imeli na razpolago, je ta test zanimiv predvsem iz stališča procesorsko bolj zahtevnih aplikacij oziroma uporabnikov. Kot je razvidno iz slike 1.6, je najbolj zmogljiv procesor na voljo pri ponudniku Google Cloud, pri katerem rezultat testa presega 14.000 točk. Tesno mu sledi Amazon Web Services, z le nekaj manj kot 14.000 točkami, medtem ko je procesor pri DigitalOcean-u nekoliko manj zmogljiv, in se s skoraj 4000 tockami manj uvršča v *CoreMark* rang 10.000 točk. Pri vseh ponudnikih smo pazili, da smo izbrali takšne storitve, ki niso trenutno deljene med druge

## POGLAVJE 1. ANALIZA ZMOGLJIVOSTI OBLAČNE STORITVE 10 HEROKU (A. DEŽMAN, E. LJUBLJANKIC, M. VRŠČAJ, A. MARKEŽIČ)

uporabnike (torej imamo pri testiranju na voljo vso procesorsko moč; to nam zagotavlja ponudnik). Obstajajo namreč tudi rešitve, kjer je procesorska moč deljena med mnoge uporabnike hkrati, imamo pa le zagotovilo, da imamo v nekem trenutku vedno na razpolago nek odstotek skupne procesorske moči (npr. 10%).



Slika 1.6: Število prejetih *CoreMark* točk.

Statistični podatki, ki izhajajo iz *CoreMark* testa zmogljivosti procesorskega sistema so vidni na sliki 1.7. Povprečna rezultata AWS in GCP sta podobna in tudi standardna deviacija je skoraj enaka. Pri testu DO je prišlo do anomalije in sicer se je pri eni izmed ponovitev testa zgodila napaka, ki je privedla do rezultata 0. Ta rezultat je delno pokvaril vrednosti ostalih statističnih kazalcev.

Enota: / (coremark)	AWS	DO	GCP
MIN	13065	0	13549
MAX	14479	12516	15105
AVG	13728	8953	14521
STDEV	376	3904	374

Slika 1.7: Statistična analiza rezultatov *CoreMark*.

### 1.7.3 Testiranje zmogljivosti datotečnega sistema

Pri testiranju datotečnega sistema smo se poslužili orodja *Fio*. Preden smo teste izvedli, se je bilo potrebno odločiti za posamezne parametre (definiranje bremena). Breme je definirano v *.job* datoteki, kjer se definirajo parametri kot so velikost bremena samega, velikost posameznega bloka, IO tip (ali gre za bralno ali pisalno operacijo), število datotek (čez koliko datotek je breme razporejeno),

## POGLAVJE 1. ANALIZA ZMOGLJIVOSTI OBLAČNE STORITVE HEROKU (A. DEŽMAN, E.LJUBLJANKIC, M.VRŠČAJ, A MARKEŽIČ) 11

itd. Velikost bloka smo nastavili na 512KB. Pri testiranju smo zahtevali izvedbo enega posla, ki smo mu nastavili različne velikosti. Kot opombo bi vnaprej omenili, da platformi Amazon Web Services (AWS) in Google Cloud (GCP) uporablja standardne trde diske, pri platformi Digital Ocean (DO) pa smo imeli na voljo SSD diske.

Na slikah 1.8 in 1.9 so prikazani rezultati za povprečno hitrost pri branju ter pisalnih operacijah. Platforma Digital Ocean ima pričakovano največjo hitrost branja in pisanja, saj teče na SSD diskih. Bolj zanimiva je primerjava platform AWS in GCP. Povprečni hitrosti branja sta približno enaki. Razlika se pojavi pri hitrosti pisanja, kjer pa ima GCP več kot 2x večjo hitrost kot AWS. DO kot že rečeno tukaj ni primerljiv, saj rešitev teče na novejši pomnilniški tehnologiji, katere bralno pisalne hitrosti daleč presegajo klasični diskovni sistem.



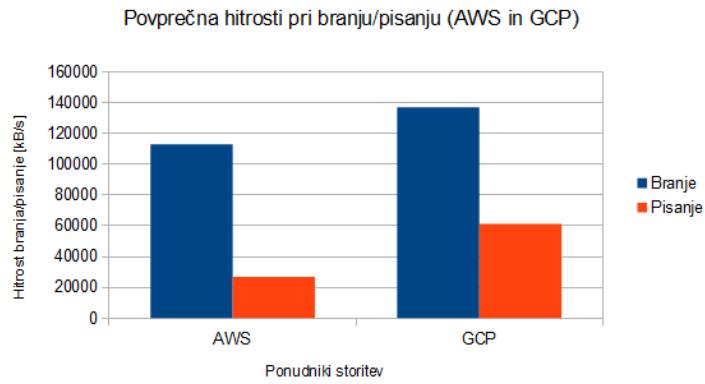
Slika 1.8: Povprečna hitrost pri branju.



Slika 1.9: Povprečna hitrost pri pisanju.

## POGLAVJE 1. ANALIZA ZMOGLJIVOSTI OBLAČNE STORITVE 12 HEROKU (A. DEŽMAN, E. LJUBLJANKIC, M. VRŠČAJ, A. MARKEŽIČ)

Glede na to, da platforma DigitalOcean uporablja SSD diske in ima s tem veliko prednost pred ostalima dneva ponudnikoma, je na sliki 1.10 prikazana še primerjava bralne in pisalne hitrosti samo za platformi AWS in GCP. Kot smo že omenili v prejšnjem odstavku sta bralni hitrosti približno enaki, pisalna hitrost pa je pri GCP več kot 2x večja od AWS.



Slika 1.10: Povprečna hitrost pri branju in pisanju za platformi AWS in GCP.

Na slikah 1.11 in 1.12 je prikazana statistična analiza rezultatov *Fio - branje* in *Fio - pisanje*. Rezultati AWS in GCP so občutno slabši, zaradi že prej omejenih razlogov. Presenetljivo pa je velika standardna deviacija pri rezultatih DO, kljub temu da je povprečni rezultat občutno boljši kot pri konkurentih, so odstopanja velika. Vzroke gre iskati v SSD tehnologiji sami, ki res da ponuja višje maksimalne hitrosti, a je tudi sama hitrost bolj nepredvidljiva in odvisna od več dejavnikov.

<i>Enota: kb/s</i>	<b>AWS</b>	<b>DO</b>	<b>GCP</b>
<b>MIN</b>	35853	661333	102342
<b>MAX</b>	173730	2506105	184785
<b>AVG</b>	110777	1671112	142746
<b>STDEV</b>	28259	525425	26046

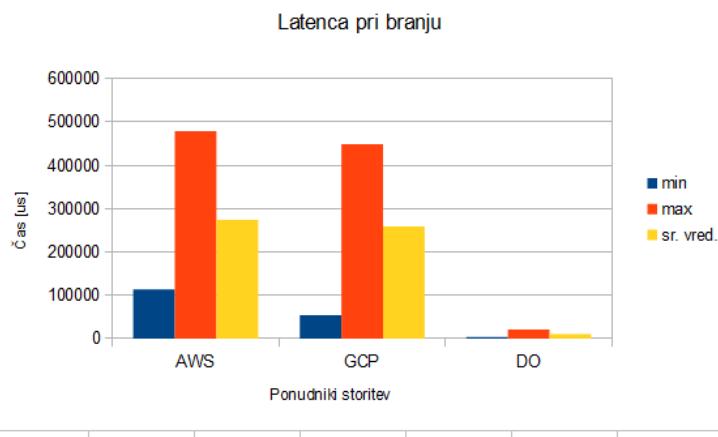
Slika 1.11: Statistična analiza rezultatov *Fio - branje*.

**POGLAVJE 1. ANALIZA ZMOGLJIVOSTI OBLAČNE STORITVE  
HEROKU (A. DEŽMAN, E.LJUBLJANKIC, M.VRŠČAJ, A MARKEŽIČ) 13**

<i>Enota: kb/s</i>	<b>AWS</b>	<b>DO</b>	<b>GCP</b>
<b>MIN</b>	<b>10546</b>	<b>203487</b>	<b>32817</b>
<b>MAX</b>	<b>69268</b>	<b>835368</b>	<b>89026</b>
<b>AVG</b>	<b>28450</b>	<b>475430</b>	<b>57928</b>
<b>STDEV</b>	<b>14418</b>	<b>169920</b>	<b>17309</b>

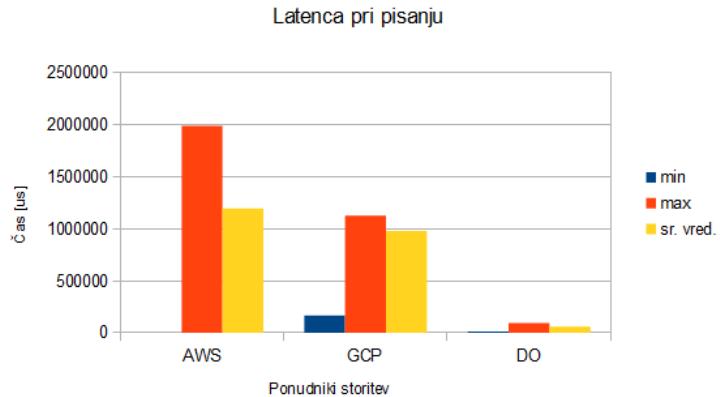
Slika 1.12: Statistična analiza rezultatov *Fio - pisanje*.

Naslednja metrika, ki smo si jo ogledali pri testiranju podatkovnega sistema je latenca ene vhodno/izhodne operacije za posamezen blok podatkov. Rezultati so vidni na slikah 1.13 (branje) in 1.14 (pisanje). Platforma DO ima najmanjše latence, zaradi že prej omenjenega razloga (SSD disk), zanimivejši pa so rezultati latenc branja in pisanja pri platformah AWS in GCP. GCP ima sicer minimalno latenco branja precej manjšo, kot je minimalna latenca pri AWS, in tudi povprečna vrednost je nekoliko nižja. Drugače so pa rezultati latenc pri branju dokaj primerljivi. Večja razlika se pojavi pri latenci pisanja, kjer so razlike precej večje. GCP ima precej manjšo povprečno latenco branja, kar je seveda ugodnejše. To je bilo pravzaprav za pričakovati, saj smo opazili tudi precej večji razkorak v samih hitrostih pisanja pri teh dveh platformah.



Slika 1.13: Minimalna, maksimalna in povprečna latenca pri branju enega bloka.

**POGLAVJE 1. ANALIZA ZMOGLJIVOSTI OBLAČNE STORITVE  
14 HEROKU (A. DEŽMAN, E. LJUBLJANKIC, M. VRŠČAJ, A. MARKEŽIČ)**

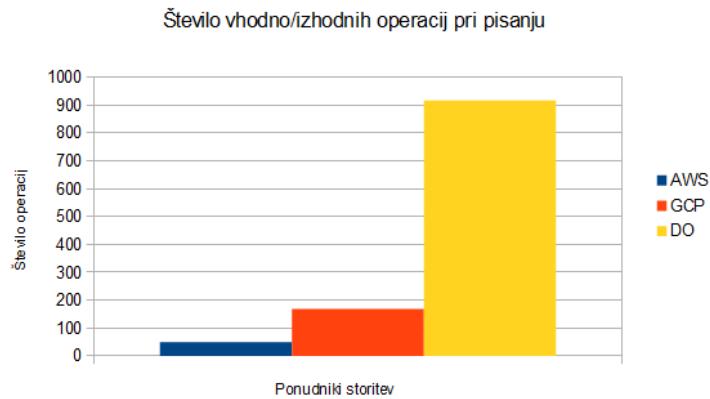


Slika 1.14: Minimalna, maksimalna in povprečna latenca pri pisanju enega bloka.

Za zadnjo metriko smo si izbrali število vhodno/izhodnih operacij na sekundo. Rezultati so vidni na slikah 1.15 in 1.16. Iz prejšnjih dveh analiz (hitrost in latenca) je moč pričakovati podobno razmerje moči tudi pri številu vhodno/izhodnih operacij. Platforma DO ima pričakovano največje število operacij, tako pri branju kot pri pisanju. Pri platformah AWS in GCP pa je ponovno opazna podobnost pri bralnih operacijah, pri številu pisalnih operacij pa je zopet večja razlika, in sicer GCP je v primerjavi z AWS pri pisalnih operacijah ponovno več kot 4x zmogljivejši.



Slika 1.15: Število vhodno/izhodnih operacij pri branju.



Slika 1.16: Število vhodno/izhodnih operacij pri pisanju.

## 1.8 Komentarji rezultatov

Katerega ponudnika torej izbrati? Kot lahko razberemo iz rezultatov, ima vsak ponudnik svoje prednosti in slabosti. Za uporabnike, ki potrebujejo veliko hitrih I/O operacij, hkrati pa niso preveč zahtevni glede zmogljivost procesorja, je DigitalOcean zagotovo prva izbira, saj novejša pomnilniška tehnologija omogoča neprimerljivo višje hitrosti bralno pisalnih operacij. Google Cloud Platform ponuja izjemno hitrost prenosa po njihovih optičnih povezavah, hkrati pa ima tudi rahlo boljšo procesorsko moč ter hitrost I/O operacij od njihovega rivala Amazon Web Services. Splošno gledano, se je od vseh najslabše odrezal prav AWS, saj v nobeni izmed kategorij ni bil najboljši. Kot že omenjeno, pa je izbira v veliki meri odvisna od samega uporabnika ter namena uporabe storitve, deloma pa tudi od drugih zunanjih dejavnikov, kot so npr. lokacija, cena, osebne preference in ostale omejitve.

POGLAVJE 1. ANALIZA ZMOGLJIVOSTI OBLAČNE STORITVE  
16 HEROKU (A. DEŽMAN, E. LJUBLJANKIC, M. VRŠČAJ, A. MARKEŽIČ)

# Literatura

- [1] A. Iosup, S. Ostermann, M. N. Yigitbasi, R. Prodan, T. Fahringer, and D. H. Epema, “Performance analysis of cloud computing services for many-tasks scientific computing,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 6, pp. 931–945, 2011.
- [2] “Digitalocean: Simple cloud computing for developers.” <https://www.digitalocean.com/>, Marec 2016.
- [3] “Amazon web services (aws) - cloud computing services.” <https://aws.amazon.com/>, Marec 2016.
- [4] “Google cloud computing, hosting services & apis.” <https://cloud.google.com/>, April 2016.
- [5] “Perfkit benchmarker.” <http://googlecloudplatform.github.io/PerfKitBenchmark/>, Marec 2016.
- [6] “Welcome to python.org.” <https://www.python.org/>, Marec 2016.
- [7] “iperf - the network bandwidth measurement tool.” <https://iperf.fr/iperf-doc.php>, April 2016.
- [8] “Industry-standard benchmarks for embedded systems.” <http://www.eembc.org/coremark/faq.php>, April 2016.
- [9] “Fio - flexible i/o tester synthetic benchmark.” [http://www.storagereview.com/fio\\_flexible\\_i\\_o\\_tester\\_synthetic\\_benchmark](http://www.storagereview.com/fio_flexible_i_o_tester_synthetic_benchmark), April 2016.