

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Matej Čuček Gerbec

**Analiza ustreznosti naročniških razmerij v
oblačnih storitvah**

DIPLOMSKO DELO
NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

prof. dr. Miha Mraz
MENTOR

Ljubljana, 2016

© 2016, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko



Tematika naloge:

Kandidat naj v svojem delu opravi analizo uvedbe oblačnega interakcijskega sistema na specifičnem primeru podatkovne baze, do katere se dostopa iz celega sveta. Pri tem naj kandidat opravi vzorčno analizo bremen in meritve zasedenosti oblačnih virov.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani izjavljam, da sem avtor dela, da slednje ne vsebuje materiala, ki bi ga kdorkoli predhodno že objavil ali oddal v obravnavo za pridobitev naziva na univerzi ali drugem visokošolskem zavodu, razen v primerih kjer so navedeni viri.

S svojim podpisom zagotavljam, da:

- sem delo izdelal samostojno pod mentorstvom prof. dr. Mihe Mraza,
- so elektronska oblika dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko in
- soglašam z javno objavo elektronske oblike dela v zbirki "Dela FRF".

— Matej Čuček Gerbec, Ljubljana, junij 2016.

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Matej Čuček Gerbec

Analiza ustreznosti naročniških razmerij v oblračnih storitvah

POVZETEK

Računalništvo v oblaku je ponudba strežniških virov, ki so trženi kot storitev. Dostopnost in cenovna ugodnost se skozi leta izboljšujeta, zaradi česar lahko veliki ponudniki nudijo svoje sisteme za gostovanje oblračnih storitev drugih ponudnikov. Take rešitve postajo zanimive za podjetja z manjšimi sredstvi. Pojavi se vprašanje, ali je ta model ustrezen tudi za mikro in majhna podjetja. Da bi to ugotovili, smo opravili meritve na primeru oblaka v razvoju, ki je v lasti mikro podjetja. Za izvedbo meritev je bilo potrebno določiti predviden poslovni model, na osnovi katerega smo oblikovali tipična bremena. Med simulacijo bremena so se izvajale meritve najetih oblračnih virov, na katerih je gostovala uporabljena oblračna rešitev. Tekom dela je predstavljena tehnika, kako lažje preverimo ustreznost uvedbe ponudbe oblračnih storitev. Na koncu dela pokažemo, da je vpeljava oblračne storitve lahko smotrna tudi za mikro podjetja.

Ključne besede: Računalništvo v oblaku, naročniško razmerje, mikro podjetje, simulacija bremen, meritve obremenitev

University of Ljubljana
Faculty of Computer and Information Science

Matej Čuček Gerbec

Analysis of suitability of cloud subscription packages

ABSTRACT

Cloud computing consists of server resources offered as services. Over the years, the availability and affordability have been improving, which allows large cloud providers to host cloud services of smaller providers. These solutions are becoming increasingly interesting to companies with smaller assets, which begets the question if the model is suitable for micro and small companies. In order to find the answer, we ran a set of measurements on a micro company's test cloud. Before performing the measurements, it was necessary to ascertain the expected business model, which was then used as a base to determine the typical load. Measurements of cloud resources were performed through simulated runs of the typical load. Throughout this work, we present a technique how to better ascertain whether it is financially sound to provide cloud services. Finally, we show that it may be suitable even for micro companies to become cloud providers.

Key words: Cloud computing, subscription model, micro company, load simulation, load measurement

ZAHVALA

Kot prvemu se zahvaljujem mentorju Mihi Mrazu za vse nasvete, pomoč in potrpljenje z večernimi e-maili. Hvala direktorju Juretu Jemcu, da mi je dovolil uporabo oblaka podjetja TMG-BMC d.o.o. kot primer v tem diplomskem delu. Njemu in sodelavcem v podjetju se iskreno zahvaljujem za pomoč in informacije. Na koncu se zahvaljujem še svoji življenjski partnerki Heleni Vidovič za potrpljenje, čustveno podporo in vse zelenjavne sokove.

— Matej Čuček Gerbec, Ljubljana, junij 2016.

KAZALO

Povzetek	i
Abstract	iii
Zahvala	v
1 Uvod	1
1.1 Pristop k problemu	2
1.2 Struktura diplomskega dela	2
2 Računalništvo v oblaku	5
2.1 Vrste oblakov	5
2.1.1 Javni oblak	6
2.1.2 Zasebni oblak	7
2.1.3 Oblak skupnosti	8
2.1.4 Hibridni oblak	9
2.2 Arhitekture oblakov	10
2.2.1 Infrastruktura kot storitev	11
2.2.2 Platforma kot storitev	11
2.2.3 Programska oprema kot storitev	12
2.3 Izbira storitve za potrebe diplomskega dela	14
3 Predstavitev naročnika oblačne storitve	15
3.1 Oblačne storitve podjetja TMG-BMC	17
3.2 Tipi uporabnikov	18
3.2.1 Zdravstvena in fizioterapevtska ambulanta	18
3.2.2 Kondicijski trener	18

3.2.3	Glavni trener	19
3.2.4	Merjenec	19
3.3	Podzvrsti uporabnikov	19
3.3.1	Uporabnik je vezan na klub	20
3.3.2	Uporabnik ni vezan na klub	20
3.4	Opisi tipičnih uporabnikov	20
3.4.1	Zdravstvena ambulanta	20
3.4.2	Samostojni športni svetovalec	21
3.4.3	Samostojni merjenec	22
3.4.4	Zdravnik v športnem klubu	22
3.4.5	Kondicijski trener v športnem klubu	23
3.4.6	Merjenec v športnem klubu	23
3.4.7	Glavni trener	23
3.5	Predvidene obremenitve tipičnih uporabnikov	23
3.6	Predvidena začetna zasedenost oblaka	25
3.7	Dostop do oblaka tekom dneva	25
4	Meritve obremenitev oblčnih virov	27
4.1	CloudWatch	28
4.2	Struktura oblčnih storitev podjetja TMG-BMC	28
4.2.1	Amazon EC2	29
4.2.2	Amazon RDS	31
4.3	Zastavljen model meritev	32
4.3.1	Meritve vmesnika API	34
4.3.2	Meritve spletnega vmesnika	37
4.4	Programska oprema za izvedbo meritev	37
4.4.1	Python	38
4.4.2	Requests	38
4.4.3	Selenium in ChromeDriver	39
4.4.4	AWS CLI	40
4.5	Rezultati	40
4.5.1	Spletni strežnik	41
4.5.2	Strežnik za vmesnik API	43

4.5.3	Strežnik RDS	45
5	Sklep	49
5.1	Ugotovitve	49
5.2	Priporočila za nadaljnje raziskave	50
	Literatura	51

1 Uvod

Računalništvo v oblaku je zanimiv in inovativen pristop k staremu računalniškemu konceptu strežnikov in odjemalcev. Pristop je inovativen v tem, da ponuja strežniške vire kot storitev. Naročniku se ni več potrebno spopadati z visokim začetnim stroškom vzpostavitve strežniškega sistema. Ta strošek in vlogo vzdrževalca namreč nase prevzame ponudnik oblčnih storitev. Tak pristop zveča krog podjetij, ki se lahko ukvarjajo z nudenjem svojih oblčnih storitev, saj si lahko v taki situaciji to privoščijo tudi podjetja s skromnejšimi sredstvi in z manjšimi oddelki IT. Podjetja lahko uporabljajo najete oblčne storitve kot polizdelek, ki ga nadgradijo s svojimi rešitvami in ponujajo dalje kot svoje oblčne storitve. Pri tem se pojavi zanimivo vprašanje, za kako majhna podjetja je ta pristop še finančno in tehnološko ustrezen.

V tem diplomskem delu bomo poskusili ugotoviti, ali so oblčne storitve dosegle točko, ko so primerne tudi za manjša podjetja.

1.1 Pristop k problemu

Problema smo se lotili z analizo že obstoječe oblačne storitve, ki je še vedno v fazi razvoja. Oblačna storitev, ki je uporabljena kot primer v diplomskem delu, je nova ponudba podjetja TMG-BMC in je še v pripravi. Analiza se osredotoča na dva vidika. Prvi vidik predstavljajo stroški, ki izvirajo iz naročnin za oblačne vire, na katerih gostuje omenjena storitev. Na srečo so informacije za ta vidik lahko dostopne pri ponudniku Amazon, ki ga uporablja podjetje TMG-BMC. Drugi vidik se ukvarja z ustreznostjo najetih oblačnih virov za pokrivanje obremenitev, ki bi jih povzročala predvidena uporaba storitve. Analiza tega vidika je bila težja, saj je bilo potrebno izdelati oceno predvidene uporabe, sestaviti tipična bremena, narediti orodja za simulacijo uporabe ter izvesti meritve obremenjenosti. Za izdelavo predvidene uporabe je bil nujen posvet z direktorjem in uslužbenci v podjetju TMG-BMC. Na podlagi te ocene smo definirali tipične uporabnike, ki so nato osnova za določitev bremena. Da bi predstavljala predvideno breme za oblačne vire, so orodja morala za simulacijo zadovoljivo oponašati delovanje tipičnega uporabnika. Programski jezik Python, ki je bogat z različnimi knjižnicami, je bil dober kandidat za izdelavo orodij za simulacije uporabe oblačnih storitev. Vsi ti koraki so nujni, da se lahko opravi verodostojne meritve zasedenosti oblačnih virov. Na osnovi teh meritev je bilo možno določiti, ali razpoložljive količine virov zadostujejo za predvideno uporabo. Te informacije bi morale skupaj z informacijami o stroških podjetju olajšati določitev finančne in tehnične ustreznosti oblačne storitve v razvoju.

V pričujočem diplomskem delu želimo določiti ustreznost najetih oblačnih virov za potrebe oblačnih storitev podjetja TMG-BMC ter ugotoviti stroške povezane z najemom.

1.2 Struktura diplomskega dela

Diplomsko delo sestavlja pet poglavij, ki se ukvarjajo s pomembnimi vidiki računalništva v oblaku in problemom ustreznosti oblačnih storitev kot ponudbe malih podjetij. Drugo poglavje se imenuje Računalništvo v oblaku in prikazuje osnovne pojme področja oblačnih storitev. Prikazani sta delitev na vrste oblakov glede na dostopnost in delitev glede na arhitekturo, katere srž je nivo abstrakcije oblačnih storitev. Poglavje služi kot osnova za ostala poglavja. V tretjem poglavju je predstavljen primer, ki je uporabljen za analizo problematike, in sicer je opisana izvedba oblačnih storitev inovativnega mikro podjetja TMG-BMC. Navedeni so predvideni tipi uporabnikov ter predvideni tipi uporabe oblačne

storitve podjetja. Ti služijo kot osnova za oblikovanje meritev. Meritve, njihovo načrtovanje in izvedba so tema četrtega poglavja. Poleg predstavitve samih meritev se v tem poglavju nahaja tudi opis arhitekture oblačne storitve podjetja TMG-BMC in predstavitev orodij, ki so uporabljena za izvedbo meritev. V petem poglavju so predstavljene ugotovitve, ki pripomorejo k boljšemu določanju, ali je ponudba oblačnih storitev primerna za manjše podjetje. Ugotovitve so podkrepjene z rezultati meritev, ki so že predstavljeni v četrtem poglavju.

2 Računalništvo v oblaku

Računalništvo v oblaku je kombinacija stare tehnološke osnove z novim poslovnim modelom. Tehnološka osnova za oblačne storitve je desetletja star princip strežnikov in odjemalcev. Novost poslovnega modela je v tem, da je *ponudnik* lastnik in upravljalec strežniškega sistema, *naročnik* pa najema ali kupuje informacijske vire. Ponudniku to omogoča optimizirano obremenitev svojih strežnikov in s tem boljši izkoristek osnovnih sredstev, naročnik pa ima ravno tako korist, ker plačuje le za porabljene vire in se izogne večjemu začetnem vložku.

V nadaljevanju tega poglavja si bomo ogledali vrste in arhitekture oblakov. Na koncu bomo v skladu s temi vidiki opredelili primer oblaka, ki je bil uporabljen v tem diplomskem delu.

2.1 Vrste oblakov

Na podlagi poslovnega modela lahko računalništvo v oblaku delimo na štiri vrste [24] in sicer na

- javni oblak,

- zasebni oblak,
- oblak skupnosti in
- hibridni oblak.

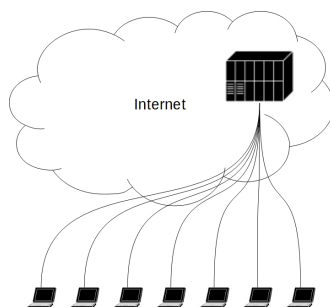
Delitev temelji predvsem na lokaciji, kjer je strežnik nameščen, in dostopnosti oblačnih virov. Različne vrste oblakov ponujajo različne stopnje zasebnosti in varnosti hranjenih podatkov ter različne nivoje nadzora nad infrastrukturo. Omogočajo tudi različno fleksibilnost najetih virov in stroškovnih modelov.

2.1.1 Javni oblak

Javni oblak je vrsta oblaka, ki je najbolj znana večini uporabnikov. Ponudniki, kot so Dropbox, Google (Gmail, Apps, ...) ipd., so vsi primeri javnega oblaka. To so storitve, s katerimi se uporabniki interneta srečujejo dnevno. Informacijski viri so javno dostopni in si jih delijo vsi naročniki, kakor je razvidno na sliki 2.1. Infrastruktura je v lasti ponudnika in se nahaja v njegovih prostorih. Tak pristop omogoča naročniku največji nivo elastičnosti oblačnih virov in stroškov.

Zaradi enotne infrastrukture in velike količine uporabnikov lahko ponudnik optimizira izkoriščenost svoje strojne opreme. Zato lahko ponuja izredno elastičen sistem, kjer naročnik lahko najame zeleno kapaciteto oblačnih virov. Naročnik s tem pridobi prednost, da lahko znatno prilagaja kapacitete najetih oblačnih virov in z njimi povezane stroške. S tem ima boljši nadzor nad svojimi izdatki in jih lahko bolj natančno prilagodi svoji dejanski porabi oblačnih virov, kakor bi to lahko recimo pri zasebnem oblaku.

Pristop z javnim oblakom ima tudi slabe lastnosti. Storitve, ki so na razpolago naročniku, so praviloma bolj toge, ker sta programska in strojna infrastruktura enaka za vse naročnike oz. uporabnike. Naročnik je namreč omejen le na komponente, ki jih ima ponudnik že pripravljene. Njegovi podatki se običajno nahajajo na strojni opremi ponudnika in se lahko nahajajo v drugi državi. S tem se odpre vprašanje, katerim različnim zakonodajam se morajo ti podatki podrežati [38]. Ker so podatki hranjeni na strojni in programski infrastrukturi ponudnika, nad katero naročnik nima popolnega nadzora, se pojavljajo tudi vprašanja glede varnosti podatkov [1].

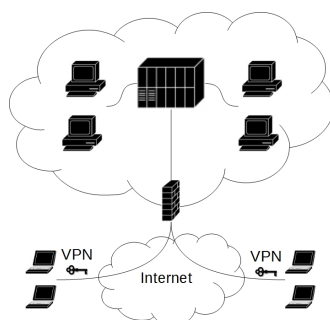


Slika 2.1: Javni oblak.

2.1.2 Zasebni oblak

Ta zvrst oblaka je še najbolj podobna klasičnemu nakupu lastnih strežnikov, zato tudi naročniku ponuja največji nivo fleksibilnosti. Ponudnik priskrbi informacijsko infrastrukturo za sistem, ki jo namesti na lokaciji naročnika. Tak oblak je namenjen le za uporabo znotraj informacijskega omrežja naročnika. Do oblčnih virov bodo dostopali samo uporabniki naročnika, zato je sama izvedba infrastrukture oblaka lahko dosti bolj fleksibilna. Ponudnik lahko celotni strojni in programski sistem prilagodi potrebam naročnika. Glavna vloga ponudnika pri zasebnem oblaku je izgradnja in vzdrževanje informacijske infrastrukture oblaka.

V zasebnem oblaku se uporablja požarne zidove, da se loči oblačne vire od javnih omrežij in da se zviša varnost in zaupnost podatkov v oblaku. Podatki so dostopni le



Slika 2.2: Zasebni oblak.

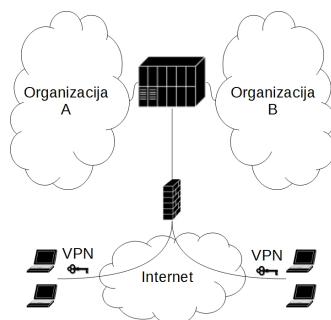
znotraj naročnikovega omrežja. Naročnik lahko poljubno omogoči dostop do podatkov preko javnih omrežij z uporabo navideznega zasebnega omrežja. S pomočjo šifriranja tak pristop omogoča navidezno ločitev javnega in zasebnega omrežja. Opisana struktura oblaka je prikazana na sliki 2.2.

Ker je zasebno omrežje ločeno od javnega in so strežniki s podatki locirani fizično v prostorih naročnika, to zagotavlja visok nivo tajnosti in zaupnosti podatkov. Istočasno pa to prinese nižjo elastičnost sistema in posledično so stroški manj prilagodljivi trenutni porabi naročnika. Če so oblačni viri redno zasedeni, nižja stroškovna elastičnost ne predstavlja hude pomanjkljivosti, saj jo odtehta nižja cena na enoto vira [25, 27].

Zasebni oblak je idealen za ustanove, ki potrebujejo izredno visok nivo varnosti glede tajnosti podatkov. Nivo varnosti je rahlo nižji pri navideznem zasebnem oblaku. Navidezni zasebni oblak je podskupina zasebnih oblakov, kjer se strojna oprema nahaja v prostorih ponudnika in ne v prostorih naročnika. V taki izvedbi je možnost, da bi prišlo do zlorab s strani uslužbencev ponudnika, nekoliko večja kot v primeru strojne opreme, ki se nahaja pri naročniku. Lahko pa zato naročnik sam prilagodi sistem svojim trenutnim potrebam in optimizira stroške glede to, kako se spreminjajo potrebe po oblačnih virih. Primer takega oblaka je Amazon Virtual private Cloud [7].

2.1.3 Oblak skupnosti

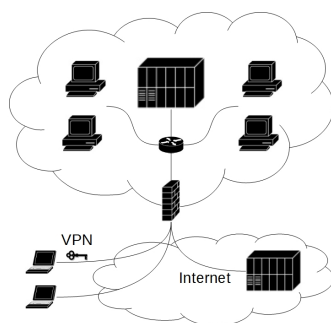
V tej vrsti oblaka je naročnik skupina ustanov, kot je razvidno na sliki 2.3. Podobno kot pri zasebnem oblaku so podatki dostopni le znotraj skupnega omrežja ustanov. Vloga ponudnika tudi tukaj obsega le postavitev infrastrukture in vzdrževanje podatkov. Oblak skupnosti ima nekatere prednosti v primerjavi z zasebnim oblakom. Ima namreč večjo možnost, da bodo zakupljeni viri uporabljeni v dovolj veliki meri in bo zato naložba stroškovno smotrna. Tak pristop je zelo primeren za javne ustanove, kjer mora do skupnih podatkov dostopati več ločenih ustanov, kot so denimo bolnice, dobavitelji elektrike, itd.



Slika 2.3: Oblak skupnosti.

2.1.4 Hibridni oblak

Hibridni oblak je kombinacija storitev javnega in zasebnega oblaka, kar je prikazano na sliki 2.4. Naročnik zasebnega oblaka lahko uporablja storitve javnega oblaka in s tem dopolni svoje storitve. Med drugim lahko uporablja javni oblak kot dodatne oblačne vire pri premoščanju obdobj, ko so strežniki zasebnega oblaka preobremenjeni. Poleg tega lahko naročnik uporablja strežnike zasebnega oblaka za nudenje storitev javnega oblaka, kot so npr. spletna pošta, ki je dostopna preko javnih omrežij, ali spletna hramba slik ali datotek. Tako lahko naročnik nudi svojim uslužbencem še neizkoriščene oblačne vire zasebnega oblaka, za katere ni potrebno vzpostaviti povezave v navidezno zasebno omrežje.

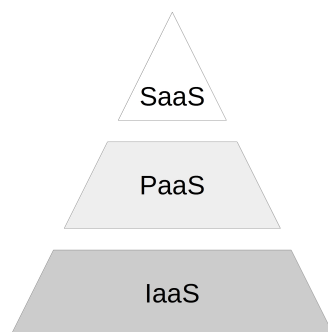


Slika 2.4: Hibridni oblak.

2.2 Arhitekture oblakov

Arhitekturo oblakov lahko delimo po slojih, kot kaže slika 2.5. Vsak sloj predstavlja nivo abstrakcije storitve, do katere dostopa naročnik. Nižje v arhitekturi naročnik dostopa do storitve, bolj neposredno lahko tudi upravlja s strojno opremo. Bolj direkten dostop do strojne opreme omogoča večjo prilagodljivost storitev. Žal pa to od naročnika zahteva več dela in več vzdrževanja. Višji nivo abstrakcije pomeni bolj preprost sistem za uporabo, a istočasno večja abstrakcija omejuje vrsto razpoložljivih storitev. Abstrakcijske nivoje delimo v tri skupine in sicer na

- IaaS - infrastrukturo kot storitev,
- PaaS – platformo kot storitev in
- SaaS – programsko opremo kot storitev.



Slika 2.5: Abstrakcijski nivoji arhitekture oblakov.

2.2.1 Infrastruktura kot storitev

To je arhitektura, pri kateri naročnik dobi dostop do najnižjega nivoja abstrakcije. Na tem nivoju imamo lahko fizične strežnike ali pa navidezne strežnike.

Fizični strežniki so bolj primerni za zasebne oblake ali oblake skupnosti, kjer je varovanje podatkov kritičnega pomena, fleksibilnost oblačnih virov pa ni toliko pomembna. Navidezni strežniki ponujajo večjo fleksibilnost. Ponudnik lahko omogoča dinamično prilagajanje strežniških virov, kot so pomnilnik, število procesorskih jeder, frekvenca procesorjev, velikost podatkovnih medijev in omrežnih kapacitet. V taki arhitekturi obstaja velika možnost, da se na isti strojni opremi, kjer se izvaja naročnikov navidezni strežnik, izvajajo hkrati tudi navidezni strežniki drugih naročnikov. To predstavlja dodatno tveganje za varnost in zasebnost podatkov, saj doda še eno šibko točko, ki se jo lahko izrabi za napad na uporabnikove podatke [15].

Naročnik lahko na strežnike namesti želene operacijske sisteme in programe. S tem pridobi zelo fleksibilen sistem, ki mu omogoča prilagoditev in izbiro storitev. Lahko se odloči za programsko opremo različnih ponudnikov in uporablja svoje programske rešitve. To prinese tudi dodatne stroške, ker mora naročnik sam priskrbeti vso potrebno programsko opremo in zagotoviti njeno vzdrževanje. Abstrakcija na tem nivoju omogoča naročniku dinamično prilagajanje strežniških virov. Zaradi tega je začetni vložek v infrastrukturo majhen v primerjavi z nakupom ali najemom fizičnih strežnikov. Infrastruktura in z njo povezani stroški rastejo skladno z naročnikovimi potrebami. Naročniku taka arhitektura olajša postavitev in izvedbo sistema, oteži pa pregled in zaračunavanje storitev. Primer take arhitekture je Amazon EC2.

2.2.2 Platforma kot storitev

Na tem nivoju arhitekturne abstrakcije naročnik ne dostopa več do infrastrukture. Ponudnik daje naročniku na razpolago dostop do programskega vmesnika, na katerem lahko nato naročnik zaganja svoje programe. Npr. Google s svojim App Enginom ponuja okolje, kjer lahko naročnik izvaja svoje programe, pisane v jezikih Node.js, Python, Ruby, GO, PHP ali Java [22]. Ta pristop naročniku omogoči veliko lažji in hitrejši razvoj oblačnih aplikacij. V primerjavi z arhitekturo IaaS se naročniku sedaj ni potrebno več ukvarjati z nastavljanjem in vzdrževanjem operacijskega sistema. Ni mu več potrebno namestiti in nastaviti vseh podpornih programov. Zato se lahko nemudoma začne ukvarjati z razvojem svoje programske rešitve. Takoj lahko namesti svojo programsko rešitev in

jo začne uporabljati. Ker ponudniki dajejo na razpolago standardne programske jezike, lahko naročnik hitro in enostavno zamenja ponudnika PaaS. Možnost poganjanja lastnih programskih rešitev omogoča naročniku tudi nalaganje standardnih programskih paketov, ki delujejo na ponudnikovih vmesnikih. Naročnik lahko tako na PaaS namesti tudi prilagojene odprtokodne rešitve, kot je recimo WordPress [32].

PaaS pristop je lahko veliko cenejši od pristopa IaaS, če se programi izvajajo le občasno, saj lahko zaračunavamo zasedenost oblačnih virov po urni postavki [9]. Tak pristop prinese s seboj tudi omejitve. Naročnik lahko namesti le programe v jezikih, ki jih ponudnik podpira, ne more pa namestiti zaprtokodnih rešitev in nima dostopa do strojnih virov in njihovih nastavitev. Ponudniku to omogoča večjo standardizacijo sistemov in s tem nižje stroške. S takim nivojem abstrakcije lahko namreč ponudnik bolje razporeja in razprši svoje vire, bolje pa lahko porazdeli tudi strojne vire glede na različne uporabnike. Izkoriščenost ponudnikovih sistemov je posledično boljša. Podobno kot je pri sistemu IaaS na virtualnih strežnikih, tudi sistem PaaS omogoča fleksibilnost glede na naročnikove potrebe. Ponudnik lahko prilagaja količine oblačnih virov, ki so namenjeni naročnikovi aplikaciji, v sorazmerju z nivojem obremenitve naročnikove aplikacije.

Arhitekturi IaaS in PaaS imata različne prednosti in slabosti. Vsaka naslavlja določeno skupino naročnikov in določen tip potreb.

Urna postavka omogoča večjo dodano vrednost na storitve za tiste uporabnike, ki imajo oblačne storitve aktivne le občasno, preostali čas so pa neuporabljene. Podobno je PaaS bolj zanimiv za uporabnike, ki želijo poganjati svoje rešitve. S tem se istočasno izognejo vezavi na programsko opremo, ki jo določa ponudnik.

2.2.3 Programska oprema kot storitev

To je arhitektura oblaka, s katero se večina ljudi srečuje najbolj pogosto. Rešitve na tem nivoju so namenjene predvsem končnemu uporabniku. Ponudnik tukaj naročniku daje dostop do končne aplikacije. Naročniku ni potrebno nameščati svojih programov ali pa razvijati novih. Ravno tako se ogne vzdrževanju infrastrukture in programske opreme. Ponujeno programsko opremo lahko začne uporabljati takoj in skoraj brez začetnih stroškov. Lep primer preproste in naročniku prijazne oblačne storitve je DropBox. Tukaj naročnik dobi sistem za strežniško hranjenje, sinhronizacijo in deljenje podatkov z drugimi naročniki. Ta rešitev je tako preprosta, da jo lahko naročnik prične uporabljati takoj po registraciji računa [16].

Pri bolj naprednih rešitvah SaaS ima naročnik praviloma opravka zgolj s tem, da programsko opremo prilagodi svojim potrebam. Primer take rešitve tipa SaaS predstavlja npr. Salesforce, ki ponuja svojim uporabnikom oblačno rešitev CRM (Customer relationship management, upravljanje odnosov s strankami) [41]. Naročnik mora sam vzpostaviti šifrate in ostale nastavitve, ki ustrezajo potrebam njegove organizacije, pri čemer je omejen na prilagoditve, ki so že del programske rešitve. Če potrebuje bolj kompleksne prilagoditve, lahko stopi korak nižje z abstrakcijo in poseže po Apexu, ki je rešitev tipa PaaS in ravno tako del ponudbe podjetja Salesforce [41].

V obeh primerih naročnik ne potrebuje lastnega strežnika, operacijskega sistema, podatkovnih baz in drugih vmesnih programov. Programska rešitev gostuje na ponudnikovi infrastrukturi. Ponudnik prevzame nase celotno vzdrževanje sistema, nadgrajevanje programske opreme in varnost podatkov. To omogoči naročniku, da se osredotoča na dejavnosti, ki so mu pomembne. Če je naročnik podjetje, se lahko osredotoči izključno na opravila, ki so vezana na jedrno dejavnost podjetja. Podjetje ne potrebuje več tako velikega oddelka IT, ki bi se ukvarjal z vzdrževanjem informacijskih sistemov. Če je naročnik posameznik, se mu ni potrebno učiti novih tehnologij, ampak lahko uporablja že ponujeno programsko rešitev. Se pa s tem naročnik odpove nadzoru nad varnostjo svojih podatkov. Ko sprejme ponudnikovo rešitev, naročnik sprejme tudi omejitve, ki pridejo s sistemom. Sprememba samega ponudnika je težja, ker govorimo o končnih programih. Naročnik postane takoj vezan na izbranega ponudnika, kar pomeni, da mora temu ponudniku in kakovosti njegovih storitev zaupati.

Ponudnik prevzame nase precejšnjo odgovornost, za kar pa odtehta veliko drugih prednosti. Vzdrževanje in razvoj oblačne programske opreme je za ponudnika lažje, kot bi bilo vzdrževanje programske opreme pri končnem uporabniku. Ponudnik točno ve, kakšna bo oblačna infrastruktura, na kateri se bo izvajala programska oprema. Ravno tako ponudnik določa, katera programska oprema bo še tekla vzporedno s ponujeno rešitvijo SaaS. To znatno olajša razvoj in nadgradnjo, saj se ponudniku ni več potrebno ukvarjati z raznolikostjo okolij, v katerih se bo njegova programska rešitev izvajala. Nadgradnjo v rešitvi SaaS lahko ponudnik izvede za vse uporabnike istočasno. Tehnično podporo izvaja na znani in za vse enaki različici programske opreme. Razlike, s katerimi se ponudnik srečuje, se pojavljajo predvsem v spletnih odjemalcih (Firefox, Chrome itd.), s katerimi uporabniki dostopajo do oblačne rešitve. Mora pa zato ponudnik prepričati svoje potencialne naročnike, da je dovolj stabilno in zaupanja vredno podjetje.

2.3 Izbira storitve za potrebe diplomskega dela

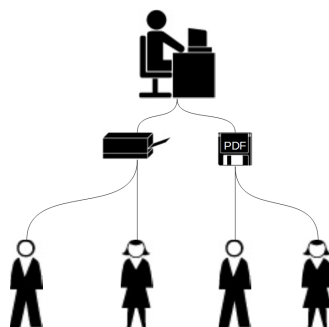
Arhitekture in vrste oblakov sta dva različna vidika oblačnih rešitev. Lahko jih poljubno kombiniramo med seboj. Morda bi za enega naročnika optimalna rešitev bila zasebni oblak z aplikacijo CRM, ki jo ponuja Salesforce, drugemu naročniku pa bi najbolj odgovarjal javni oblak z virtualnim strežnikom Amazon EC2. Spet za tretjega bi bila izdelana rešitev DropBox. V tej diplomi bomo pogledali na oblak tako skozi oči uporabnika kot ponudnika oblačnih storitev. Iz ponudniškega pogleda smo se v diplomskem delu osredotočili na oblačno rešitev SaaS, ki jo je razvilo podjetje TMG-BMC d.o.o. Ta gostuje na Amazonovih rešitvah IaaS, bolj specifično na EC2 in RDS, ter na rešitvi SaaS, ki jo ponuja YouTube.

3 Predstavitev naročnika oblačne storitve

Podjetje TMG-BMC d.o.o. (v nadaljevanju podjetje TMG-BMC) je specializirano na področju mišične diagnostike, ki je osnovana na lastnih metodologijah in tehnoloških sistemih podjetja.

Večina ponudbe temelji na ustaljenem sistemu TMG. Ta omogoča hitro in neinvazivno merjenje stanja posameznih mišic. Sistem temelji na tenziomiografiji, metodologiji diagnostike mišic, ki jo je razvilo podjetje samo. Tenziomiografija je namreč nov pristop za monitoriranje mišičnega stanja. Ukvarja se s funkcionalno diagnostiko mišice. V nasprotju z elektromiografijo (v nadaljevanju EMG), s katero se pridobiva informacije o mišici na podlagi njenega električnega odziva, metoda TMG uporablja fizični odziv mišice. Ime sistema in podjetja tako izvirata iz kratic za tenziomiografijo - TMG, podobno kot je EMG kratica za elektromiografijo.

S to tehnologijo podjetje TMG-BMC nudi svojim strankam edinstvena ter napredna orodja in storitve za pridobivanje informacij o stanju in o sestavi mišic. Ti podatki so neprecenljivi za vse, ki se ukvarjajo s področjem profesionalnih športov. Izrednega pomena so tudi v medicini. V športu tenziomiografija omogoča bolj natančno prilagajanje



Slika 3.1: Trenutna ponudba podjetja TMG-BMC.

treningov, saj so ti sestavljeni na podlagi meritev in v skladu s potrebami posameznega športnika. Na podlagi meritev lahko tudi sprejemamo bolj argumentirane odločitve pri izbiri članov v skupinskih športih. V medicini tenziomiografija omogoča lažje sledenje in prilagajanje fizioterapije po mišičnih poškodbah. Podjetje tako tesno sodeluje z znanimi športnimi klubi pri optimizaciji treningov, preprečevanju poškodb in rehabilitaciji športnikov. Nekaj bolj znanih športnih klubov, ki se poslužujejo storitev ali izdelkov podjetja TMG-BMC, so FC Manchester United, FC Barcelona, FC Liverpool, AFC Fiorentina itd. Med drugim podjetje TMG-BMC nudi svoje izdelke in storitve ustanovam, kot so npr. Manchester Metropolitan University, INEFC, Quiron Groupo Hospitalario [14]. Trenutno podjetje TMG-BMC ob svoji merilni opremi ponuja le programsko opremo za lokalno uporabo. Kot prikazuje slika 3.1 so stranki podatki meritev dostopni le na računalniku, kjer je nameščena programska oprema. Merjenci pa lahko prejmejo le statična poročila o meritvah v tabeli excel ali obliki pdf.

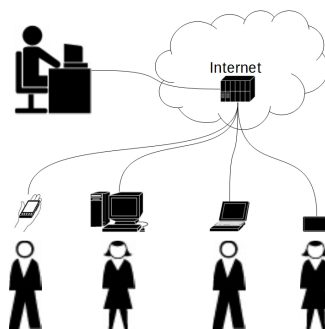
Oblačne storitve tipa SaaS bodo tako samo logična nadgradnja že obstoječe ponudbe podjetja TMG-BMC. S tem, ko bo podjetje nadgradilo že obstoječe programske rešitve na programsko opremo kot storitev, bo lahko ponudilo svojim uporabnikom nove načine za uporabo izdelkov podjetja TMG-BMC, kot prikazano na sliki 3.2. Pristop SaaS predstavlja veliko prednost za merjence, ki bodo imeli možnost dostopa do interaktivnih poročil in priporočil zdravnika ali trenerja. Z rešitvijo SaaS bo uporabnik dobil prost dostop do podatkov, saj ne bo več vezan na lokalno napravo za pregledovanje in interpretacijo podatkov. To bo tudi omogočilo istočasen dostop do podatkov več različnim uporabnikom, ker dostopnost podatkov ne bo več omejena le na eno fizično napravo.

Uporabniki bodo lahko vzdrževali stike z merjenci tudi na daljavo s pomočjo interaktivnih poročil.

3.1 Oblačne storitve podjetja TMG-BMC

Podjetje TMG-BMC bo v prihodnje v svojem oblaku ponujalo dva vmesnika. To sta spletni vmesnik za končne uporabnike in pa vmesnik API za integracijo s programi in drugimi oblačnimi storitvami. Vmesnik API omogoča prenos meritev v oblak, dostop do podatkov v oblaku in njihovo urejanje. Spletni vmesnik je namenjen končnim uporabnikom. Deli se na dva podsistema in sicer na vmesnik, ki je namenjen merjencem, in na vmesnik, ki je namenjen uporabnikom sistema TMG.

Vmesnik za merjence omogoča pregled poročil meritev merjenca ter pregled in potrjevanje vaj, ki jih je naročil uporabnik sistema TMG. Merjencem omogoča tudi pregled posnetkov vaj in komunikacijo z uporabnikom sistema TMG. Merjenec ima zgolj dostop do podatkov o svojih meritvah. Vmesnik za uporabnike sistema TMG omogoča pregled poročil meritev, predpisovanje vaj in komunikacijo z merjenci. Uporabnik sistema TMG ima v oblaku lahko dostop do meritev različnih merjencev, lahko pa si tudi več uporabnikov sistema TMG med seboj deli dostop do opravljenih meritev.



Slika 3.2: Oblačna ponudba podjetja TMG-BMC.

3.2 Tipi uporabnikov

Oblačne rešitve so raznolike in vsaka izmed njih pritegne drugačen tip uporabnikov. Slednje bi lahko razdelili v štiri osnovne skupine. To so zdravstvene ambulante, športni kondicijski trenerji, glavni trenerji in merjenci.

Zdravstvene in fizioterapevtske ambulante so nedvomno potencialen uporabnik, ker tenziomiografija v kombinaciji z drugimi diagnostičnimi orodji omogoča merjenje fizičnega stanja mišice in s tem bolj kakovostno diagnostiko živčno-mišičnih in mišično-skeletnih težav. Informacija o fizičnem stanju mišice je zelo koristna v športnih panogah, saj pomaga pri določanju fizične pripravljenosti posameznih mišic in mišičnih sklopov ter tako predstavlja neprecenljivo orodje pri sestavljanju režima treninga. To nas pripelje k drugi in tretji skupini uporabnikov, ki ju tvorijo kondicijski trenerji in glavni trenerji. Vse te tri skupine povezuje četrta skupina in sicer končni uporabniki, merjenci. Ti so lahko športniki ali pacienti.

3.2.1 Zdravstvena in fizioterapevtska ambulanta

Zdravstvenim ambulantam pristop SaaS omogoči boljšo izkoriščenost merilnih sistemov. Izurjeni asistenti lahko izvajajo meritve, medtem ko zdravnik obdeluje in pregleduje poročila o meritvah. Pri uporabi lokalnega programa bi bil zdravnik omejen bodisi na statičen izvid v obliki PDF bodisi na natisnjene izvide ali pa bi zasedel računalnik, na katerem je program nameščen, in v tem času onemogočal izdelavo meritev. S pristopom SaaS se lahko poveže na aplikacijo v oblaku in z lahkoto pregleduje in išče po vseh predhodno izdelanih meritvah. Uporaba interaktivnih poročil mu omogoči pregled nad tem, kako vestno merjenec/pacient sledi njegovi terapiji. Zdravnik oz. fizioterapevt bi v bazi najbrž imel večje število merjencev z manjšim številom meritev, saj bi do zdravnika verjetno prišlo več pacientov z zdravstvenimi težavami. Vsak pacient bi bil merjen le v sklopu diagnostike in fizioterapije, ne pa redno kot pri športnikih.

3.2.2 Kondicijski trener

Trenerju uporaba SaaS olajša sestavljanje režima treninga. Kjerkoli se trener nahaja, ima lahek in preprost dostop do podatkov v digitalni obliki, kar mu poenostavi pregled nad stanjem športnikov in zmanjša potrebo po delu s tiskanimi poročili. Uporaba oblačne storitve mu nudi tudi dodatno varnost, saj za varnostno kopiranje podatkov, ki so v

oblaku, skrbi ponudnik oblčnih storitev. Kondicijski trener bo najbrž imel manjše število merjencev, ker bo najverjetneje meril le člane kluba, kjer je zaposlen. Bo pa zato imel veliko večje število meritev na vsakega merjenca, saj bo želel slediti napredku, ki ga predpisani trening prinese.

3.2.3 Glavni trener

Za glavnega trenerja rešitev SaaS predstavlja večji nivo priročnosti in hitrejši ter bolj pregleden dostop do informacij o njegovi ekipi. Na hitro lahko pregleda stanja vseh svojih športnikov kar iz svojega mobilnega telefona ali tablice, kjerkoli in kadarkoli mu odgovarja. Ni mu potrebno več s seboj nositi celotne zbirke tiskanih poročil, saj mu je katerokoli poročilo dostopno že na mobilnem telefonu. Glavni trener je v takem pogledu le porabnik informacij v oblaku in sam ne dodaja novih podatkov v sistem. Lahko pa vpliva na pogostost izvajanja meritev.

3.2.4 Merjenec

Podobno kot glavni trener je tudi merjenec, bodisi športnik ali pacient, predvsem porabnik informacij v oblaku. Le v manjši meri prenaša podatke v oblak. To bi počel predvsem v komunikaciji z zdravnikom/trenerjem in s potrjevanjem opravljenih vaj. Merjenec s predpisanim režimom vaj bi predvidoma vsaj enkrat na dan dostopal do svojih podatkov. Do oblaka dostopa, da pregleda vaje, ki jih mora opraviti po predpisanem režimu. Oblačna aplikacija SaaS omogoča uporabniku tudi komunikacijo z zdravnikom/trenerjem, če mu kakšna od vaj ni razumljiva, lahko pa si predhodno ogleda video posnetek vaje, ki gostuje na YouTube-u.

3.3 Podzvrsti uporabnikov

Glede na različno poslovno politiko uporabnikov lahko ločimo dve pomembni variaciji pri uporabi programske opreme v športu. Večji športni klubi si lahko privoščijo zdravnike in kondicijske trenerje, ki delajo ekskluzivno za ta klub, klubi z manj sredstvi pa se lahko odločijo, da najamejo zdravnike in kondicijske trenerje po pogodbi. Pogodbenemu zdravniku oz. kondicijskemu trenerju, ki nudi svoje storitve več klubom hkrati (torej nudi svetovanja, diagnostiko in terapije), prinese uporaba sistema TMG prednost pred ostalo konkurenco. Omogoča mu še bolj individualno obravnavo vsakega športnika z dodatnimi informacijami o mišičnem stanju, ki so na razpolago le preko sistema TMG.

Seveda obstajajo primeri, ko lahko določen zdravnik, trener ali fizioterapevt tudi prosto nudi svoje storitve vsem. Glavna razlika med podzvrstmi uporabnikov je v tem, ali je uporabnik oblačnih storitev podjetja TMG-BMC vezan na nek športni klub ali ne. V odvisnosti od vezave in s tem poslovnega modela uporabnika se bodo tudi načini uporabe oblačnih storitev razlikovale. S tako delitvijo pridemo do sedmih tipov uporabnikov.

3.3.1 Uporabnik je vezan na klub

Uporabniki, ki so vezani na en sam klub, bodo dostopali do oblaka podjetja TMG-BMC drugače kot uporabniki, ki imajo dejavnost ločeno od kluba. Klubske zdravniki, fizioterapevti in trenerji bodo v bazi imeli omejeno količino merjencev, saj bodo najverjetneje omejeni le na člane znotraj kluba in morda še na druge merjence, ki so povezani s klubom. Zato pa lahko pričakujemo veliko meritev za vsakega posameznega merjenca, saj se lahko trenerji in zdravniki v takem okolju veliko bolj posvetijo vsakemu posamezniku. Izvajanje meritev je lahko dnevno ali vsaj tedensko. Ker imajo klubske trenerji, zdravniki in fizioterapevti več časa za vsakega merjenca, lahko pričakujemo, da bodo merjenci manj intenzivno uporabljali oblak podjetja TMG-BMC.

3.3.2 Uporabnik ni vezan na klub

Pri zdravnikih, fizioterapevtih in trenerjih z zasebno dejavnostjo lahko pričakujemo veliko večje število različnih merjencev. Meritve izvajajo po potrebi ali pogodbi. To pomeni, da bo število meritev na merjenca dokaj majhno. Skupno število meritev bo še vedno znatno, ker bo na to vplivalo večje število. Tak uporabnik bo v bazi imel veliko število merjencev. Pričakujemo, da bodo merjenci bolj intenzivno uporabljali oblak podjetja TMG-BMC. Trener, fizioterapevt ali zdravnik se namreč ne more tako podrobno posvetiti vsakemu merjencu posebej, zato bodo merjenci bolj odvisni od informacij, ki jih pridobijo iz svojih poročil in režimov treninga, ki so jim dostopni v oblaku.

3.4 Opisi tipičnih uporabnikov

V nadaljevanju sledijo podrobnejši opisi vseh tipov uporabnikov.

3.4.1 Zdravstvena ambulanta

Zdravnik v ambulanti, ki ni vezana na športni klub, bo imel večje število merjencev. V primeru preventivnih meritev inštruktorji iz podjetja TMG-BMC priporočajo eno

meritev na leto. Glede na tip preventive se v enem meritvenem sklopu izvaja meritev na desetih do dvajsetih parih mišic (dvajset do štirideset meritev mišice). V primeru meritev zaradi poškodbe je priporočeno izvajati meritve obeh mišic v paru (mišico na levi in desni strani telesa), po potrebi pa tudi ostalih mišic v funkcijskem sklopu (koleno, gleženj, komolec itd.). Meritve se izvajajo skozi obdobje obravnave poškodbe v sklopu terapije. Inštruktorji iz podjetja TMG-BMC v tem primeru priporočajo vsaj pet meritev in sicer eno začetno diagnostično meritev in štiri meritve za nadzorovanje napredka terapije. Razmak med meritvami je lahko zelo različen, saj lahko mišične poškodbe trajajo od nekaj tednov do več mesecev [23, 29]. Po vsakem paketu meritev lahko od zdravnika pričakujemo vsaj en ogled poročila stanja mišic [12]. Zdravnik bo za vsak paket meritev pregledal vaje, ki jih algoritem v oblaku priporoča, in jih po potrebi spremenil. Tako bo izdelal režim terapije za preventivo poškodb ali za rehabilitacijo po poškodbi. Pri posvetu z inštruktorji iz podjetja TMG-BMC je ocena števila pacientov pri takem zdravniku znašala nekaj sto pacientov. To bi se skladalo s hitrim izračunom pacientov na osnovi števila tedenskih obiskov pri zdravnikih, ki znaša nekje med 70 in 90 tedenskih obiskov za splošnega zdravnika [17]. Število tedenskih pregledov je lahko še manjše v primeru specialista, ker si bo morda moral vzeti še več časa na pacienta. Leto ima približno 50 tednov. To skupaj nanese 3.500-4.500 obiskov na leto, za katere pa iz zgornjih informacij že vemo, da so med njimi tudi ponovljeni obiski v primeru poškodb. Tako pridemo na do 900 pacientov na leto s približno petimi paketi meritev. Če predpostavimo, da prva meritev obsega 10 parov mišic in da ponovne meritve zajamejo mišice vsaj enega funkcijskega sklopa na obeh straneh telesa, pridemo skupno na približno 30.000 meritev mišic na leto. Te vrednosti so računane za primer ambulate, kjer meritve opravljajo asistenti zdravnika. Celotna meritev vseh dvajsetih parov mišic traja približno od dvajset minut do pol ure. To pomeni, da lahko v štirideseturnem delovnem tednu ena oseba opravi največ 80 celotnih meritev. Če zdravnik opravlja meritve sam, bodo številke seveda nižje. Sistem TMG predstavlja le eno izmed razpoložljivih diagnostičnih orodij, medtem ko mora zdravnik nameniti čas tudi dialogu s pacientom.

3.4.2 Samostojni športni svetovalec

Kondicijski trener, ki ni vezan na športni klub, bi predvidoma imel tudi večje število merjencev. Tak trener bi ponujal svoje storitve več klubom, pri čemer bi meritve najverjetneje izvajal mesečno ali pa sezonsko oz. letno in sicer na začetku, sredini in koncu

pripravljalnega obdobja. Če vzamemo za primer nogometne ekipe, je v klubu okoli 30 igralcev [36]. Pri privzeti celotni meritvi (20-30 minut na osebo) lahko trener izmeri približno 16 igralcev na dan, kar pomeni, da porabi za eno ekipo dva dni. V enem delovnem tednu lahko torej izmeri dve ekipi. V dveh tednih lahko torej trener izvede meritve za štiri športne klube, kar pomeni, da lahko člane kluba meri v istih fazah treninga znotraj ene sezone. Tak uporabnik bo imel okoli 100 merjenecv v bazi. Če upoštevamo še priložnostne meritve, lahko v bazi pričakujemo od 100 do 200 merjenecv. Predvideno letno število mišičnih meritev je skupno s priložnostnimi meritvami 20.000, od tega je 12.000 pogodbenih meritev.

3.4.3 Samostojni merjenec

Za merjenca, ki ga je izmeril zdravnik, ki ni vezan na športni klub, je uporaba oblačne storitve veliko bolj zanimiva. Tak merjenec namreč nima rednega dostopa do zdravnika tako kot športniki v klubu. Pričakujemo, da si bo tak merjenec vsak dan ponovno ogledal že dano priporočilo zdravnika. Video posnetki vaj so za takega uporabnika veliko bolj koristni. Športniki v klubu z internim uporabnikom sistema TMG in športniki z zunanjim svetovalcem bodo imeli podoben vzorec dostopanja do oblaka podjetja TMG-BMC. V klubih z zunanjim svetovalcem lahko pričakujemo, da bo glavni trener pregledoval poročila meritev in nato na osnovi teh sestavil primeren trening. Merjenec zunanjega svetovalca bo imel dnevno samo bralne dostope do oblačne baze podatkov. Na začetku in koncu terapije oz. treninga pričakujemo komunikacijo med merjencem in svetovalcem. Pričakujemo, da bo merjenec na koncu terapije potrdil opravljene vaje v terapiji.

3.4.4 Zdravnik v športnem klubu

Pri zdravniku, ki je vezan na športni klub, pričakujemo bistveno nižje število merjenecv v bazi. Zdravnik lahko za preventivo uporablja iste meritve kot jih trener za namene treninga. Tak zdravnik bi najverjetneje izvajal dodatne meritve le v primeru poškodbe, večinoma pa bi bil zgolj porabnik podatkov v oblaku. V večjih klubih z večjim številom moštev je možno, da bi zdravnik prevzel del merjenja. Meritve bi opravljal na moštvih, ki drugače niso deležni meritev s sistemom TMG. V tej raziskavi se bomo omejili na zdravnika kot čistega porabnika podatkov, ki izvaja zanemarljivo malo meritev s sistemom TMG. Za zdravnika predvidevamo, da bo za vsakega igralca v moštvu pregledal podatke vsaj enkrat mesečno, za poškodovane igralce pa vsaj enkrat tedensko.

3.4.5 Kondicijski trener v športnem klubu

Kondicijski trener, ki je vezan na športni klub, bo meritve opravljal na športnikih v klubu. V primeru nogometnega kluba lahko številko zaokrožimo na 30 igralcev [36]. Inštruktorji podjetja TMG-BMC predlagajo dve do štiri letne meritve vseh glavnih mišic (deset do dvajset mišičnih parov). Poleg tega predlagajo še tedensko meritev dveh do štirih mišičnih parov, ki so se na celotnih meritvah izkazali kot problematični. To pripelje na približno 15.000 meritev posameznih mišic na leto. Iz tega lahko sklepamo, da bo tak uporabnik v bazi hranil okoli 30 merjencev in 15.000 meritev za vsako leto uporabe. Poleg shranjevanja meritev pričakujemo pri takem uporabniku vsaj en ogled poročila za vsak paket meritev. To pomeni 30 ogledov tedensko, letno pa vsaj 1500 ogledov.

3.4.6 Merjenec v športnem klubu

Merjenci v športnem klubu z internim uporabnikom sistema TMG imajo manj razlogov, da bi redno spremljali podatke v oblaku. Trener ali zdravnik sta jim vedno na razpolago. Pripravita jim trening ali terapijo, pri kateri lahko prisostvujeta. Pričakujemo lahko le dostope z namenom potrditve priporočenih vaj ali ogleda zaradi radovednosti. Pri takem uporabniku ne pričakujemo več kot enega ogleda tedensko, kar letno nanese približno 50 ogledov.

3.4.7 Glavni trener

Glavni trener v športnem klubu predvidoma dostopa do oblčnih storitev le kot uporabnik informacij. Pred tekmami in sestanki pregleda poročilo o celotnem moštvu in po potrebi preveri tudi posamezne igralce. Kot primer vzemimo nogomet v Angliji, kjer vsako moštvo odigra 38 tekem [20], v slovenski prvi ligi pa je ta številka 33 [31]. Če predpostavimo še dvakrat toliko sestankov kot tekem, lahko zaokrožimo število dostopov na 100 ogledov letno.

3.5 Predvidene obremenitve tipičnih uporabnikov

Za lažji pregled so v tabeli 3.1 zbrane predvidene količine dostopov do spletnega vmesnika vseh šestih tipičnih uporabnikov, medtem ko so v tabeli 3.2 zbrane predvidene količine uvozov meritev TMG.

Tip uporabnika	Mesečni dostopi	Letni dostopi
Zdravstvena ambulanta	300	3500
Športni svetovalec	80	1000
Samostojni merjenec	30	365
Zdravnik v športnem klubu	40	500
Kondicijski trener v športnem klubu	150	2000
Merjenec v športnem klubu	4	50
Glavni trener	35	450

Tabela 3.1: Dostopi do spletnega vmesnika.

Tip uporabnika	Mesečni dostopi	Letni dostopi
Zdravstvena ambulanta	2500	30000
Športni svetovalec	1200	20000
Samostojni merjenec	0	0
Zdravnik v športnem klubu	0	0
Kondicijski trener v športnem klubu	960	15000
Merjenec v športnem klubu	0	0
Glavni trener	0	0

Tabela 3.2: Uvozi meritev (posamezna mišica).

3.6 Predvidena začetna zasedenost oblaka

Da bi določil začetno zasedenost oblčnih storitev podjetja TMG-BMC, sem se sestal z direktorjem podjetja. Njegova ocena za začetno uporabo predvideva kot naročnike 5-7 športnih klubov, 1-2 zdravstveni ambulantni in morda enega samostojnega kondicijskega trenerja. Izrazil je tudi namen, da se storitve zaračunavajo zdravniku in ne njegovim pacientom. Na osnovi teh podatkov lahko v prvem letu sklepamo na 350 merjencev in 100.000 meritev, ki jih bodo v bazo shranili športni klubi. V tem obdobju bi doprinos zdravstvenih ambulant znašal 1800 merjencev in 60.000 meritev. Kondicijski trener bi doprinesel še 100 merjencev in 20.000 meritev. Skupno bi tako v bazi dosegli okoli 2.200 merjencev in 180.000 meritev.

3.7 Dostop do oblaka tekom dneva

Težko je določiti, kakšen bi bil raspored dostopov skozi dan, saj so potencialni uporabniki lahko razpršeni po svetu (ZDA, EU, Japonska, Kitajska, ...). V tem smislu celotno število dostopov hipotetično razdelimo enakomerno preko celega dneva.

4 Meritve obremenitev oblačnih virov

Meritve so ključnega pomena pri izdelavi ocene obremenitve in stroškov, ki so povezani z vzdrževanjem oblačnih storitev. Za izdelavo zanesljivih meritev je potrebno združiti več različnih vidikov. Potrebno je sestaviti poslovni vidik uporabe oblaka, ki je predstavljen v prejšnjem poglavju. Na osnovi tega vidika lahko sestavimo oceno predvidene uporabe oblaka in s tem model tipičnega dostopa. Poleg tega je potrebno poznati tehnično izvedbo oblaka, na osnovi katere določimo, katera orodja lahko uporabimo in katera orodja je še potrebno izdelati za izvedbo meritev in simulacijo tipičnih bremen. Tehnična izvedba omogoča oceno stroškov, ki so povezani z oblačnimi viri.

Primer, kako tehnični vidik vpliva na izvedbo meritev, je izbira orodja za nadzor oblačnih virov. Oblačna storitev podjetja TMG-BMC se izvaja na oblačnih virih podjetja Amazon. Slednji ponuja naročnikom orodje za nadzor nad porabo virov najetih oblačnih storitev. To orodje je oblačna storitev z imenom CloudWatch, ki naročniku olajša pregled nad porabo virov vseh storitev, ki so naročene pri podjetju Amazon. Pri izdelavi diplomskega dela sem se za uporabo te storitve odločil zaradi enostavnosti uporabe in brezplačnega dostopa v osnovni ponudbi.

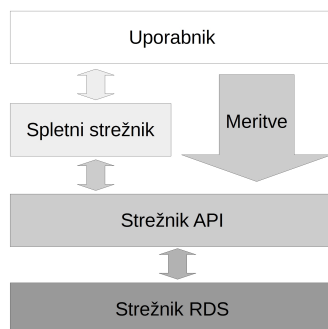
4.1 CloudWatch

Kot že omenjeno, podjetje Amazon v naboru svojih oblačnih storitev ponuja tudi storitev CloudWatch. Ta omogoča uporabniku nadzor nad porabo oblačnih virov [2]. CloudWatch izvaja meritve oblačnih virov z minutno resolucijo. Ponuja pregled nad podatki, kot so obremenjenost procesorja, uporaba pasovne širine omrežnih povezav itd. Storitve shranjuje statistiko uporabe za obdobje zadnjih dveh tednov. Pri pregledu podatkov je možno izbrati želeno obdobje, želena periodo za združevanje meritev, način združevanja meritev in oblačni vir, ki ga uporabnik želi opazovati. Uporabnik lahko tako recimo za obdobje enega tedna preveri dnevno maksimalno obremenitev procesorja in vsoto dnevnih dostopov do podatkovne baze. Poleg pregleda uporabe oblačnih storitev CloudWatch dopušča izdelavo alarmov in dogodkov, ki omogočajo napreden nadzor nad delovanjem najetih oblačnih storitev [40]. V okviru storitve CloudWatch je vključeno brezplačno pregledovanje osnovnih metrik z največ 5 minutno resolucijo. Poleg tega je na razpolago še 10 brezplačnih alarmov in 1 milijon dostopov do vmesnika API za storitev CloudWatch [11]. Za potrebe diplomskega dela se bomo omejili na brezplačno ponudbo in zajem podatkov s 5 minutno resolucijo. Meritve so bile izvedene za tri strežnike, ki so opisani v naslednjih podpoglavjih.

4.2 Struktura oblačnih storitev podjetja TMG-BMC

Oblačne storitve podjetja TMG-BMC sestavljajo trije strežniki in sicer dva strežnika tipa EC2 (Elastic Compute Cloud) in en strežnik tipa RDS (Relational Database Server). Strežnika tipa EC2 skrbita za vmesnik API in za spletni vmesnik, medtem ko strežnik tipa RDS skrbi za hranjenje podatkov. Struktura je prikazana na sliki 4.1.

Vmesnik API za oblak podjetja TMG-BMC deluje po principu REST [33] in gostuje na lastnem spletnem strežniku, ki je ločen od spletnega strežnika za uporabniški vmesnik. Uporabnik spletnega oblaka podjetja TMG-BMC ima direktni stik le s strežnikom za spletni vmesnik in z lokalnim programom za izvajanje meritev. Meritve iz lokalnega programa se prenašajo v oblak preko strežnika z vmesnikom API, ki izvaja preverjanje pristnosti in dovoljenj uporabnika ter nato zapisuje podatke na strežniku RDS. Spletni strežnik, podobno kot lokalni program za zajem meritev, uporablja strežnik API za preverjanje pristnosti in dovoljenja ter za dostop do podatkov na strežniku RDS. Uporabnik tako nima nikoli neposrednega dostopa do strežnika RDS, medtem ko strežnik API vedno



Slika 4.1: Struktura oblačnih storitev podjetja TMG-BMC.

skrbi za zasebnost in varnost podatkov.

Ker je oblak podjetja TMG-BMC še vedno v fazi testiranja in ga podjetje še ne ponuja uporabnikom, so trenutno vsi strežniki v najnižjem cenovnem rangu. EC2 strežniki so tipa t2.micro, medtem ko je RDS strežnik tipa db.t1.micro.

4.2.1 Amazon EC2

EC2 je okrajšava za Elastic Compute Cloud, kar lahko prevedemo kot prilagodljivo računalništvo v oblaku, pri čemer je »Compute« mišljen predvsem v smislu izračuna in obdelave podatkov. Tako ne preseneča, da je Amazon svojo ponudbo oblačnih navideznih strežnikov poimenoval EC2 [19]. Na voljo je širok spekter nastavitvev, med katerimi lahko uporabnik dinamično prehaja [35] in tako prilagaja stroške trenutnim potrebam. Amazon v svoji ponudbi EC2 nudi strežnike, ki so optimizirani za različne potrebe [18]; lahko so optimizirani za spominsko intenzivne aplikacije, za procesorsko intenzivne aplikacije, za strežnike z grafičnimi pospeševalniki podjetja NVIDIA, za strežnike, ki so optimizirani za shranjevanje podatkov itd. Strežnike je možno najeti po potrebi ali za določeno obdobje. Tip naročnine vpliva na začetne in sprotne stroške. Naročnik lahko izbere operacijski sistem, ki ga želi uporabljati na strežniku EC2. Podjetje Amazon ponuja različne distribucije Linux in rešitve na osnovi operacijskih sistemov Microsoft Windows.

V času pisanja diplomskega dela sta oba strežnika EC2, ki ju podjetje TMG-BMC uporablja za svoj oblak, tipa t2.micro. Najeta sta po modelu po potrebi. Na strežnikih teče distribucija SUSE Linux Enterprise Server. V tem tipu strežnika ima naročnik na razpolago 1 procesor in 1GB rama. Strežnik prilagaja moč procesorja trenutni porabi,

kar pomeni, da v situacijah z večjo obremenitvijo strežnik lahko začasno poveča moč procesorja. Amazon jih priporoča za strežnike z občasnimi dostopi, kot so npr. razvojni strežniki [18].

Uporaba strežniških virov za to konfiguracijo se zaračunava na podlagi aktivnih ur strežnika in znaša 1,5 centa USD na uro v času pisanja diplomskega dela. Strežnik se šteje kot aktiven, tudi če trenutno do njega ni povezav, dovolj je, da na strežniku teče operacijski sistem. Kot neaktiven se šteje le v primeru, če je ugasnjen. Ker morata biti strežnik za API in spletni strežnik ves čas aktivna in pripravljena, če slučajno pride do zahtevka s strani uporabnika, to pripelje do mesečnega stroška okoli 10 USD na posamezni strežnik.

Poleg omenjenih stroškov so prisotni še stroški za storitev EBS (Elastic Block Store), ki se uporablja v kombinaciji s strežniki EC2. EBS nudi hrambo podatkov za strežnike EC2 in omogoča lažjo migracijo med različnimi tipi strežnikov EC2 [3]. Večina podatkov je shranjena na strežniku RDS. V primeru oblaka podjetja TMG-BMC so stroški, ki so povezani s storitvijo EBS, zanemarljivo majhni.

Za strežnike tipa EC2 t2.micro storitev CloudWatch omogoča spremljanje naslednjih metrik [4]

- CPUCreditBalance – razpoložljivi žetoni za povečanje moči procesorja,
- CPUCreditUsage – uporabljeni žetoni za povečanje moči procesorja,
- CPUUtilization – zasedenost procesorja,
- DiskReadBytes – količina bajtov, prebranih iz diska,
- DiskReadOps – količina bralnih operacij na disku,
- DiskWriteBytes – količina bajtov, zapisanih na disk,
- DiskWriteOps – količina pisnih operacij na disku,
- NetworkIn – hitrost prenosa vhodnih podatkov na mreži,
- NetworkOut – hitrost prenosa izhodnih podatkov na mreži,
- NetworkPacketsIn – število prejetih paketov preko mreže,
- NetworkPacketsOut – število poslanih paketov preko mreže,

- `StatusCheckFailed` – neuspešne preverbe delovanja,
- `StatusCheckFailed_Instance` – neuspešne preverbe delovanja strežnika in
- `StatusCheckFailed_System` – neuspešne preverbe delovanja Amazonovega sistema.

4.2.2 Amazon RDS

Kratice RDS je okrajšava za izraz "Relational Database Server", kar v slovenščini pomeni strežnik relacijske baze podatkov. Podjetje Amazon ponuja raznolike tipe relacijskih baz podatkov v storitvi RDS. Podjetje TMG-BMC je za svoj oblak izbralo bazo MySQL. Podobno kot to počne za strežnike EC2, Amazon ponuja tudi za strežnike RDS različne nastavitve, ki omogočajo naročniku hitro prilagajanje trenutnim potrebam [10, 28].

Amazon ponuja štiri osnovne skupine strežnikov RDS

- `Standard – Latest Generation` – tretja generacija strežnikov RDS, z izboljšano računsko močjo,
- `Standard – Previous Generation` – druga generacija strežnikov RDS, z uravnoteženo ponudbo računske moči, pomnilnika in mrežne infrastrukture,
- `Memory Optimized` – strežniki druge generacije, z več računske moči in pomnilniškimi izboljšavami,
- `Burst Capable` – strežniki RDS, ki omogočajo povečanje zmogljivosti za kratka časovna obdobja.

Za razvojno obdobje so se v podjetju TMG-BMC odločili za uporabo strežnika `db.t1.micro`, ki je namenjen le za razvojno testiranje. Ta konfiguracija vsebuje en procesor in 0.613 GB pomnilnika.

Podobno kot pri strežniku EC2 v prejšnjem poglavju, se tudi pri tem tipu strežnika zaračunava poraba na osnovi števila aktivnih ur. Aktivnost strežnika se upošteva kot pri EC2 v prejšnjem poglavju. V času pisanja diplomskega dela je bila cena urne postavke za strežnik RDS podjetja TMG-BMC 2,5 centa USD [6]. Mesečni strošek za strežnik RDS znaša torej okoli 20 USD.

Storitev `CloudWatch` omogoča spremljanje naslednjih metrik za strežnike tipa `db.t1.micro` [5]

- `BinLogDiskUsage` – količina prostora, ki ga zasedajo dnevniki,

- CPUUtilization – zasedenost procesorja,
- DatabaseConnections – število povezav do podatkovne baze,
- DiskQueueDepth – dolžina sklada diskovnih operacij,
- FreeStorageSpace – razpoložljiva količina prostora v podatkovni shrambi,
- FreeableMemory – količina pomnilnika, ki se lahko sprosti,
- NetworkReceiveThroughput – vhodni pretok na mrežnih vmesnikih,
- NetworkTransmitThroughput – izhodni pretok na mrežnih vmesnikih,
- ReadIOPS – število bralnih operacij,
- ReadLatency – čas zakasnitve pri branju,
- ReadThroughput – bralni pretok,
- SwapUsage – količina zasedenega izmenjalnega prostora,
- WriteIOPS – povprečno število diskovnih operacij na sekundo,
- WriteLatency – čas zakasnitve pri pisanju in
- WriteThroughput – pisalni pretok.

4.3 Zastavljen model meritev

Arhitektura oblaka in stroškovni model naročnin predstavljajo osnovo, na podlagi katere se lahko odločimo, kako bomo zastavili izvedbo meritev. V primeru oblaka podjetja TMG-BMC je meritve potrebno izvesti na treh strežnikih

- na strežniku za spletni vmesnik,
- na strežniku za vmesnik API,
- na strežniku relacijske baze.

Spletni strežnik za svoje delovanje uporablja strežnik za vmesnik API in slednji za svoje delovanje uporablja strežnik relacijske baze. Obremenitev strežnika višje v arhitekturi vpliva na obremenitev strežnikov nižje v arhitekturi. V oblaku podjetja TMG-BMC sta uporabljena dva osnovna tipa dostopa

- uvoz meritev TMG preko vmesnika API in
- pregled in urejanje meritev preko spleta.

Pri vnosu meritev TMG preko vmesnika API prihaja do obremenitve tako strežnika za vmesnik API kot tudi strežnika relacijske baze. Pri pregledovanju in urejanju meritev preko spleta so obremenjeni vsi trije strežniki.

Da bi lahko bolje ocenili vpliv specifičnega dostopa na obremenitve strežnikov, moramo ločiti izvajanje meritev. Tako moramo izvesti ločene meritve, kjer se simulirajo zapisi meritev TMG v bazo, in meritve, kjer se simulira dostop uporabnikov do spletnega vmesnika.

Na sestavo protokola meritev je vplivala tudi struktura naročniških razmerij za oblačne storitve podjetja Amazon. Ker je cena za najem strežnikov za oblak podjetja TMG-BMC pretežno neodvisna od porabe virov in se plačuje samo za aktivne ure delovanja, postane glavno vprašanje glede meritev, ali zakupljeni viri lahko zadovoljijo predvidene obremenitve. Vse to v kombinaciji s predvidenim manjšim številom naročnikov omogoči poenostavitev strukture meritev. Da preverimo, ali strežniki zadovoljijo predvidene obremenitve, se lahko omejimo na analizo maksimalnih obremenitev. Zato za meritve izberemo enega tipičnega uporabnika, ki bo povzročil največjo obremenitev sistema. V ta namen sem za reprezentativnega uporabnika izbral zdravstveno ambulanto. Kot tipično količino opravljenih mišičnih meritev znotraj enega sklopa lahko privzamemo začetno meritev, ki zajema 10 mišičnih parov. Z uporabo največjega števila predvidenih mišičnih meritev na meritveni sklop zagotovimo največjo obremenitev oblačnih resursov. V tabeli 4.1 so prikazane predvidene frekvence dostopov za osemurni delovni dan.

Dostopi do spletnega vmesnika	15x
Uvoz sklopov meritev TMG	15x
Število meritev na sklop	40x
Trajanje sklopa meritev TMG	30 min
Razmak med dostopi do spletnega vmesnika	30 min
Razmak med uvozi sklopov meritev TMG	30 min

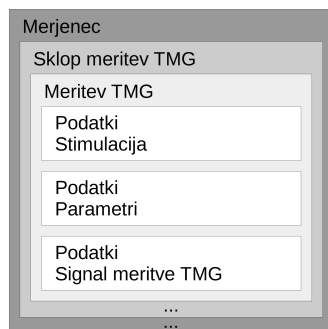
Tabela 4.1: Razčlenitev tipičnih dostopov do oblaka podjetja TMG-BMC znotraj osemurnega delovnika.

4.3.1 Meritve vmesnika API

Pri meritvah vmesnika API se izvajajo meritve obremenitev resursov strežnika EC2, kjer gostuje API, in strežnika RDS, kjer gostuje podatkovna baza. Meritve se izvajajo za simulirano uvažanje podatkov opravljenih meritev TMG. Uporabljeno orodje za poganjanje simuliranih bremen mora omogočati komunikacijo s strežnikom, ki uporablja tehnologijo REST preko protokola https. V času pisanja diplomskega dela oblachna ponudba podjetja TMG-BMC uporablja samopodpisano potrdilo za protokol https. Zato je koristno, če orodje omogoča komunikacijo tudi z nepreverjenimi potrdili. Podatki se med strežnikom za vmesnik API in odjemalcem prenašajo v obliki datotek JSON. JSON je tekstovni format zapisa podatkov, ki je osnovan na jeziku JavaScript in je podprt s strani mnogih programskih jezikov. S tem struktura JSON omogoča enostaven prenos podatkov [26].

Kot smo omenili že v prejšnjem poglavju, je za tipično breme izbrana zdravstvena ambulanta. Simulirano breme za tega uporabnika je sestavljeno iz uvoza merjenja in pripadajočega sklopa meritev. Sklop meritev zaobsega meritve dvajsetih mišic. Tabela 4.2 prikazuje število potrebnih klicev do vmesnika API za uvoz tipičnega sklopa meritev. Slika 4.2 prikazuje strukturo podatkov meritev TMG v oblaku podjetja TMG-BMC.

Pri uvozu merjenja se prenašajo podatki *ime*, *priimek*, *šifra spola*, *šifra dominantne strani zgornjega dela telesa*, *šifra dominantne strani spodnjega dela telesa*, *šifra države*, *naslov* (neobvezno), *elektronska pošta* (neobvezno), *telefonska številka* (neobvezno), *datum rojstva* (neobvezno), *datum vnosa*, *uporabniško ime lastnika podatkov*, *enolika identifikacijska koda* (UUID4), *datum spremembe podatkov* (neobvezno), *uporabniško ime osebe*,



Slika 4.2: Struktura meritvenih podatkov v oblaku podjetja TMG-BMC.

Klic vmesnika API	Število klicev
Uvoz merjenca	1
Uvoz sklopa meritev	1
Uvoz meritev mišic	20
Uvoz podatkov meritev mišic (amplituda stimulacije)	20
Uvoz podatkov meritev mišic (izračunani parametri)	20
Uvoz podatkov meritev mišic (signal meritve)	20

Tabela 4.2: Število klicev do vmesnika API, razčlenjeno po tipih klicev.

ki je nazadnje spremenila podatke (neobvezno), datum uvoza podatkov.

Pri uvozu sklopa meritev se prenašajo podatki *šifra merjenca, šifra diagnoze* (poškodovan, zdrav, . . .), *šifra športa, teža, višina, starost merjenca* (če pri podatkih merjenca manjka datum rojstva), *opombe osebe, ki je izvajala meritev, datum vnosa, uporabniško ime lastnika podatkov, enolika identifikacijska koda (UUID4), datum spremembe podatkov* (neobvezno), *uporabniško ime osebe, ki je nazadnje spremenila podatke* (neobvezno), *datum uvoza podatkov.*

Pri uvozu meritve mišice se prenašajo podatki *šifra sklopa meritve, šifra uporabljenega tipa senzorja, šifra mišice, šifra strani mišice, datum vnosa, uporabniško ime lastnika podatkov, enolika identifikacijska koda (UUID4), datum spremembe podatkov* (neobvezno), *uporabniško ime osebe, ki je nazadnje spremenila podatke* (neobvezno), *datum uvoza podatkov.*

Pri uvozu podatkov meritve mišice se prenašajo podatki *šifra meritve mišice, šifra tipa podatkov, podatki* (decimalna števila, ločena z znakom »|«), *datum vnosa, enolika identifikacijska koda (UUID4), datum uvoza podatkov.*

Predvidena velikost paketov za uvoz novega merjenca, sklopa meritve in meritve mišice je nekaj 100 bajtov, saj vsebujejo pretežno šifre in krajša besedila. Pri uvozu podatkov amplitude meritve in izračunanih parametrov lahko prav tako pričakujemo manjše pakete v redu velikosti nekaj 100 bajtov, ker podatki vsebujejo do največ 5 realnih števičnih vrednosti z natančnostjo na dve decimalni mesti. Največji so paketi s signalom meritve, za katere je predvidena velikost nekaj kilobajtov in jih sestavlja 1000 števil s plavajočo vejico.

Glavni delež obremenitve predstavljajo uvozi meritev mišic. Za uvoz meritev mišic je potrebnih kar 80 od predvidenih 82 klicev vmesnika API. Ravno tako signali meritve TMG predstavljajo veliko večino prenešenih podatkov. Zato uvoz uporabnika ob vsakem uvozu sklopa meritev TMG predvidoma ne predstavlja znatne povečave obremenitve oblračnih virov.

Potek meritve za oceno obremenitve, ki jo povzroči uvoz meritev TMG, je sestavljen iz 16ih simulacij bremena, ki se izvajajo vsakih 30 minut. Vsaka simulacija vsebuje 82 klicev do vmesnika API s simuliranimi podatki.

Korak simulacije	Čas
Prijava v spletni vmesnik	5 s
Izdelava novega uporabniškega računa za merjenca	60 s
Izbira merjenca	5 s
Izdelava in pregled poročila	120 s
Določitev vaj	10 s/mišični par

Tabela 4.3: Predvidena količina časa, ki jo uporabnik potrebuje za izvedbo koraka.

4.3.2 Meritve spletnega vmesnika

Pri meritvah spletnega vmesnika se izvajajo meritve obremenitve strežnika EC2, kjer gostuje spletni vmesnik, strežnika EC2, kjer gostuje vmesnik API, in meritve strežnika RDS, kjer gostuje podatkovna baza. Meritve se izvajajo za simulirane dostope do spletnega vmesnika. Spletni vmesnik je grajen na ogrodju AngularJS, ki v veliki meri uporablja JavaScript in AJAX [8]. Zaradi uporabe tehnologij JavaScript in AJAX mora orodje za izvajanje simulacij v čim večji meri oponašati dostope preko spletnega brskalnika. Ker vmesnik API in spletna stran uporabljata samopodpisana potrdila, mora izbrano orodje za izvajanje simulacij omogočati dostope do strani z nepreverjenimi potrdili.

Enako kot meritve API vmesnika, so meritve spletnega vmesnika tudi sestavljene iz 16-ih simulacij, ki so se izvajale vsakih 30 minut.

Vsaka simulacija je bila sestavljena iz predvidenih tipičnih korakov, ki bi jih zdravnik izvedel pri obdelavi novega merjenca. V tabeli 4.3 so predstavljeni vsi koraki. Podani so tudi predvideni časi, ki bi jih uporabnik potreboval za vsak korak.

4.4 Programska oprema za izvedbo meritev

Za izvedbo meritev obremenjenosti so potrebna tri orodja

- orodje za zajem obremenjenosti oblačnih resursov,
- orodje za simulacijo dostopov do vmesnika API in
- orodje za simulacijo dostopov do spletnega vmesnika.

Kot omenjeno, oblačne storitve podjetja Amazon že ponujajo orodje za zajem obremenjenosti oblačnih resursov. To je storitev CloudWatch, do katere uporabnik dostopa

preko programa AWS CLI (AWS Command Line Interface). AWS CLI je program za interakcijo z oblračnimi storitvami podjetja Amazon preko ukazne vrstice. Uporablja se lahko v raznih operacijskih sistemih.

Orodja za simulacijo dostopov sem spisal samostojno. V pomoč so mi bile razne že obstoječe knjižnice, ki so poenostavile izdelavo orodij.

Da bi izdelal orodja, sem se odločil, da bom uporabil programski jezik Python v kombinaciji s knjižnicama `requests` in `selenium` in programom `ChromeDriver`. Python je programski jezik, ki omogoča hitro izdelavo programskih orodij in se poleg tega poslužuje uporabniku prijazne sintakse [21]. `Requests` je knjižnica za Python, ki je namenjena enostavni komunikaciji s spletnimi strežniki. Poenostavi namreč simulacijo dostopov do vmesnika API [34]. Knjižnica `selenium` v kombinaciji s programom `ChromeDriver` služi avtomatizaciji rabe brskalnika Google Chrome [13, 39].

4.4.1 Python

Python je objektno usmerjen tolmačen programski jezik. Struktura jezika je preprosta za uporabo in na razpolago je veliko knjižnic, ki poenostavijo razvoj aplikacij [37]. Te knjižnice na primer poenostavijo delo z datotekami JSON, spletnimi strežniki, več-nitnimi opravili, generatorji osebnih podatkov itd. Jezik je zelo lahek za uporabo in omogoča hitro in preprosto zaganjanje programov, ki so spisani v tem jeziku. Glavna hiba Pythona je počasnost, saj je tolmačen programski jezik [30].

Zaradi vseh naštetih razlogov je Python idealen za pisanje raznih orodij, ki ne bodo pogosto uporabljana. Dejstvo, da gre za tolmačen in zato počasen programski jezik, ne predstavlja težave za simulacijo dostopov do vmesnika API in simulacijo uporabe spletnega brskalnika, saj sta to relativno počasna in nezahtevna procesa, ki ne potrebujeta visokega nivoja optimizacije. Obstoj knjižnic `requests` in `selenium` za Python sta vsekakor pripomogla k moji izbiri Pythona za izdelavo orodij za simulacije dostopov.

4.4.2 Requests

`Requests` je knjižnica za Python, namenjena za hitro in enostavno pisanje programov, ki komunicirajo s spletnimi strežniki. Velika prednost te knjižnice je zelo preprost sistem za vzpostavitev seje s spletnim strežnikom. Izredno enostavno je tudi izvajanje zahtevkov do spletnega strežnika in rokovanje z odgovori.

Izvedba simulacije uvažanja sklopa meritev TMG s knjižnico `requests` deluje tako,

da se na začetku uvažanja ustvari nova seja s strežnikom API. Na tej točki se izvede prijava z uporabniškim imenom in geslom. Nato sledi serija 82 spletnih zahtevkov tipa POST po komunikacijskem modelu REST. Pri vsakem zahtevku se generira primerno besedilo JSON z vsemi potrebnimi podatki za izvedbo vnosa v bazo. Polja, ki so potrebna za generiranje besedila JSON, so predstavljena v poglavju Meritve vmesnika API. Za generiranje podatkov so uporabljene knjižnice `faker`, `names`, `numpy`, `random` in `uuid`. Z njimi se lahko enostavno in hitro generira vse podatke, ki so potrebni za simulacijo uvažanja novega merjenca, sklopa meritev in meritev mišic.

Uporaba samopodpisanega potrdila ne predstavlja težav za knjižnico `request`. V primeru, ko potrdilo ni preverjeno, knjižnica samo javi opozorilo, ki ga lahko hitro izključimo z enim samim ukazom.

Če spletni zahtevek ni sprejet in strežnik ne pošlje odgovora, je potrebno ujeti izjemo, ki jo generira knjižnica `request`. Program za simulacijo bremena API obravnava take primere tako, da ponavlja zahtevek, dokler ne dobi odgovora.

4.4.3 Selenium in ChromeDriver

Selenium je knjižnica za več programskih jezikov, vključno s Pythonom, in omogoča avtomatizacijo izvajanja spletnih brskalnikov. Za avtomatizacijo delovanja brskalnika mora knjižnica imeti dostop do prilagojenega brskalnika, ki zna interpretirati klice iz knjižnice. Google ponuja program `ChromeDriver`, ki je različica brskalnika `Chrome` in deluje s knjižnico `selenium`.

Selenium omogoča akcije, kot so iskanje elementov na spletni strani, premikanje strani k zelenemu elementu, vnašanje besedila v polja za vnos, klikanje miškinih gumbov itd. Omogoča tudi preziranje napak v potrdilih, kar je zelo koristno v primeru uporabe samopodpisanih potrdil.

Knjižnica `selenium` je odlična pri avtomatizaciji delovanja brskalnika na navadnih spletnih straneh, a stvari se malo zakomplicirajo na dinamičnih straneh, ki v veliki meri uporabljajo JavaScript in AJAX. Take so npr. strani, narejene na ogrodju `AngularJS`. Pri teh je potrebno biti bolj kreativen pri iskanju in klikanju na elemente.

Tak primer je recimo izbiranje vaje za mišice. Spletna stran uporablja element tipa `combobox`, ki je nastavljen kot neviden. Obkljuka se preko JavaScripta s klikom na drugi element, ki je prazen in težje ločljiv od ostalih elementov na strani. Selenium za izvedbo klika na element zahteva, da je element viden. Preprosta rešitev za to težavo je, da

začasno naredimo element *combobox* viden in ga nato kliknemo. Problem nastane, če je element izven vidnega področja. V tem primeru selenium omogoča, da vidno področje prestavimo, da bo tudi izbrani element viden. ChromeDriver to izvede tako, da se premika po strani, dokler pozicija izbranega elementa ni v vidnem polju. Praviloma bi to moralo biti možno tudi z določenim odmikom od pozicije elementa, kar pa žal ne deluje pravilno na straneh, ki uporabljajo AngularJS. Posledično se lahko zgodi, da je izbrani element prekrit z drsnikom okvirja in je preprečeno pošiljanje klika na element. Možen trik, ki odpravi to težavo, je, da na strani začasno spremenimo pozicijo elementa in nato izvedemo klik.

Drugi problem, na katerega sem naletel pri uporabljanju knjižnice selenium, je nastal pri dekodiranju vsebine elementov. Pogosto se je namreč dogajalo, da besedilo ni bilo zaznано pri elementih, ki so vsebovali še druge elemente poleg besedila. Rešitev je bila postavljena z branjem vsebine elementa (`innerHTML`) in iskanjem besedila znotraj vsebine. To rešitev je nekoliko olajšalo dejstvo, da se je v strukturi spletne strani oblaka podjetja TMG-BMC pri vseh takih elementih besedilo nahajalo na začetku vsebine elementa.

4.4.4 AWS CLI

AWS CLI je program za upravljanje z oblaknimi storitvami podjetja Amazon preko ukazne vrstice. To je razvidno tudi iz samega imena AWS CLI, Amazon Web Services Command Line Interface, kar bi prevedeno pomenilo vmesnik spletnih storitev Amazon za uporabo v ukazni vrstici. Program je na razpolago za operacijske sisteme Microsoft Windows, Linux, Mac OSX in Unix.

V okviru diplomskega dela je program koristen za dostopanje do storitve CloudWatch. S programom AWS CLI lahko izvedemo zahtevek do storitve CloudWatch. Odgovor vrne v obliki JSON na standardni izhod, ki ga lahko nato preusmerimo v datoteko. Ta datoteka vsebuje serijo meritev izbranega resursa, ki jo je v Pythonu zelo lahko prevesti v vrsto slovarjev in nadalje v preprosto datoteko CSV (Comma Separated Values). Te rezultate se lahko nato v Pythonu oblikuje v grafe s pomočjo knjižnice matplotlib.

4.5 Rezultati

V simulacijah tipičnih bremen smo zajeli meritve vseh metrik, ki jih storitev CloudWatch daje na razpolago. To nam omogoči lažje sklepanje na vzorce in prepoznavo pomembnih

metrik.

Zajete meritve sem nato združil po strežnikih in opravljenih simulacijah. To sem storil za lažji pregled nad meritvami in lažjo primerjavo med simuliranimi bremenami. Tako zbrani podatki so serija podatkovnih točk, ki so še vedno težko pregledni v tekstovni obliki, so pa že primerni za prikaz z grafi. Za razumljiv tekstovni prikaz je bilo potrebno še eno združevanje in sicer na podatke, ki so bili zajeti v času aktivnosti in v času mirovanja. Med temi podatki sem nato poiskal povprečno vrednost ter, kjer je bilo to smiselno, še minimalno in maksimalno vrednost.

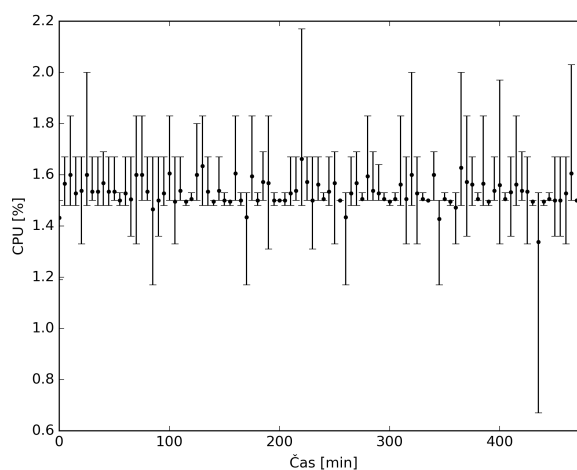
4.5.1 Spletni strežnik

Meritve na spletnem strežniku so bile opravljene le za simulacijo uporabe spletnega vmesnika. Pri simulaciji uvoza meritev TMG meritev obremenjenosti resursov spletnega strežnika nisem opravljal, saj strežnik ni bil v uporabi.

Kakor je razvidno iz metrik `DiskReadBytes`, `DiskWriteBytes`, `DiskReadOps` in `DiskWriteOps` v tabeli 4.4, je bil disk med testiranjem neobremenjen. Iz tega lahko sklepamo, da so resursi v pomnilniku zadoščali za delovanje spletnega strežnika. Iz tabele je razvidno, da dostopi do spletnega vmesnika nimajo izrazitega vpliva na obremenjenost procesorja. To je še bolj očitno na sliki 4.3, saj ni razvidnih vrhov tekom simulacije. Omrežni resursi jasno kažejo na veliko razliko v obremenjenosti med obdobjem mirovanja in aktivnim obdobjem simulacije.

Metrika	Mirovanje			Aktivnost		
	Min.	Povp.	Maks.	Min.	Povp.	Maks.
CPUUtilization [%]	1,17	1,52	2	0,67	1,55	2,17
NetworkIn [kB]		65,47			204,01	
NetworkOut [kB]		66,36			3620,47	
NetworkPacketsIn [no.]		238,14			1376,94	
NetworkPacketsOut [no.]		288,83			1762,81	
CPUCreditBalance [no.]		142,57			142,57	
CPUCreditUsage [no.]		0,08			0,08	
DiskReadBytes [kB]		0			0	
DiskWriteBytes [kB]		0			0	
DiskReadOps [no.]		0			0	
DiskWriteOps [no.]		0			0	

Tabela 4.4: Izluščeni rezultati meritev obremenitve spletnega strežnika pri simulaciji spletnih dostopov.



Slika 4.3: Obremenitev procesorja spletnega strežnika pri simulaciji spletnih dostopov. Točke na sliki prikazujejo povprečno vrednost. Stolpci pod piko prikazujejo minimalno vrednost, stolpci nad piko pa maksimalno.

4.5.2 Strežnik za vmesnik API

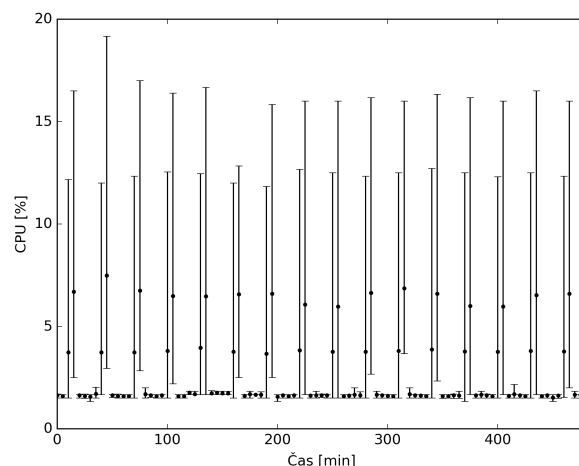
Meritve obremenitev resursov strežnika za vmesnik API so bile zajete tako ob simulaciji uvoza meritev TMG kakor tudi ob simulaciji dostopov do spletnega vmesnika. To je bilo potrebno, saj je strežnik za vmesnik API obremenjen pri obeh simulacijah. Iz tabel 4.5 in 4.6 je razvidno, da je procesor bolj obremenjen v obdobju aktivnega delovanja simulacij. Slika 4.4 jasno prikazuje vrhove, ki se ujemaajo z aktivnim časom simulacije. Podobno je v času aktivnosti obeh simulacij povečana obremenitev mrežnih virov. Med simulacijama lahko opazimo razliko med obremenitvijo mrežnih virov v vhodni in izhodni smeri. Razlog je najverjetneje več kompleksnih klicev vmesnika API v simulaciji spletnih dostopov, kot jih je bilo primeru simulacij uvažanja meritev TMG. Uvažanje meritev TMG sestavljajo preprosti pisalni klici do vmesnika API za vnašanje v podatkovno bazo. Pri uporabi spletnega vmesnika je več bralnih klicev do vmesnika API, ki so namenjeni prikazovanju rezultatov meritev TMG in izbiranju med različnimi ponujenimi možnostmi. Enako kot pri spletnem strežniku je bil v času meritev disk neobremenjen.

Metrika	Mirovanje			Aktivnost		
	Min.	Povp.	Maks.	Min.	Povp.	Maks
CPUUtilization [%]	1,33	1,64	2,17	1,33	5,15	19,17
NetworkIn [kB]		66,21			864,18	
NetworkOut [kB]		73,04			401,93	
NetworkPacketsIn [no.]		245,97			1149,91	
NetworkPacketsOut [no.]		281,75			1082,72	
CPUCreditBalance [no.]		100,75			100,85	
CPUCreditUsage [no.]		0,08			0,26	
DiskReadBytes [kB]		0			0	
DiskWriteBytes [kB]		0			0	
DiskReadOps [no.]		0			0	
DiskWriteOps [no.]		0			0	

Tabela 4.5: Izluščeni rezultati meritev obremenitve strežnika za vmesnik API pri simulaciji spletnih dostopov.

Metrika	Mirovanje			Aktivnost		
	Min.	Povp.	Maks.	Min.	Povp.	Maks
CPUUtilization [%]	1,19	1,63	2,17	1,17	5,31	22
NetworkIn [kB]		61,9			310,87	
NetworkOut [kB]		61,96			449,39	
NetworkPacketsIn [no.]		201,66			682,88	
NetworkPacketsOut [no.]		243,19			1050,81	
CPUCreditBalance [no.]		103,89			103,9	
CPUCreditUsage [no.]		0,08			0,27	
DiskReadBytes [kB]		0			0	
DiskWriteBytes [kB]		0			0	
DiskReadOps [no.]		0			0	
DiskWriteOps [no.]		0			0	

Tabela 4.6: Izluščeni rezultati meritev obremenitve strežnika za vmesnik API pri simulaciji uvažanja meritev TMG.



Slika 4.4: Obremenitev procesorja spletnega strežnika pri simulaciji spletnih dostopov. Točke na sliki prikazujejo povprečno vrednost. Stolpci pod piko prikazujejo minimalno vrednost, stolpci nad piko pa maksimalno.

4.5.3 Strežnik RDS

Meritve obremenitve strežnika RDS so bile opravljene za obe simulaciji. Iz tabel 4.7, 4.8 in slike 4.5 je razvidna povezava med obremenjenostjo procesorja in aktivnostjo simulacije. Vzorec je manj izrazit kakor pri strežniku API, ampak bolj izrazit kot pri spletnem strežniku.

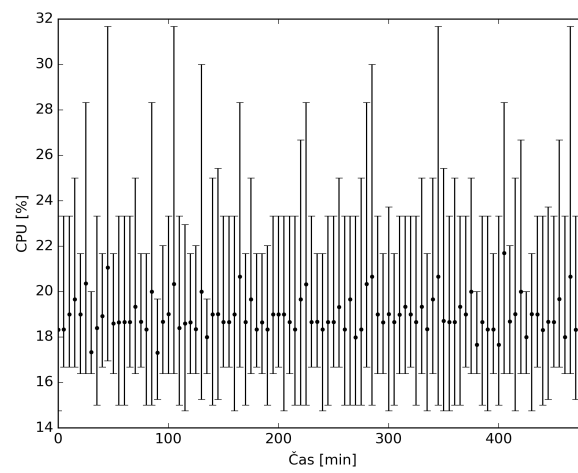
Obremenitve diska in mrežnih virov se v večji meri skladajo z obdobji aktivnosti obeh simulacij. So pa očitne razlike v obremenitvi vhodnih in izhodnih virov med simulacijo spletnega vmesnika in simulacijo uvoza meritev TMG. Pri simulaciji spletnega vmesnika je denimo obremenitev mrežnih virov v izhodni smeri veliko večja kot v vhodni. V nasprotju s tem je pri simulaciji uvoza meritev TMG razlika v obremenitvi med vhodno in izhodno smerjo veliko manjša in je vhodna smer bolj obremenjena. Podobno situacijo lahko opazimo tudi pri obremenitvi diska, pri čemer se pisalna obremenitev sklada z vhodno, bralna pa z izhodno obremenitvijo. To je najverjetneje povezano s tem, da ima uporaba spletnega vmesnika večje število bralnih klicev do vmesnika API. Uvoz meritev TMG je sestavljen le iz pisalnih klicev do vmesnika API.

Metrika	Mirovanje			Aktivnost		
	Min.	Povp.	Maks.	Min.	Povp.	Maks
CPUUtilization [%]	14,75	18,69	28,33	14,75	19,49	31,67
NetworkReceiveThroughput [kB/s]	0,05	0,08	0,25	0,05	0,21	0,93
NetworkTransmitThroughput [kB/s]	1,17	1,32	2,31	1,19	3,72	23,37
FreeableMemory [MB]	89,93	96,29	107,5	90,39	96,12	108,49
DiskQueueDepth [no.]	0	0	0,02	0	0	0,01
ReadIOPS [IOPS/s]	0	0,22	1,12	0	0,24	1,57
WriteIOPS [IOPS/s]	0	0,1	1,13	0	0,15	1,02
ReadThroughput kB/s]	0	0,14	1,71	0	0,15	1,23
WriteThroughput [kB/s]	0	0,93	22,66	0	1,49	12,77
ReadLatency [ms]	0	0,5	16	0	1,95	184
WriteLatency [ms]	0	1,25	13,51	0	1,69	12,11
DatabaseConnections [no.]		2,05			6,56	
FreeStorageSpace [MB]		4110,64			4110,64	
SwapUsage [MB]		197,39			197,39	

Tabela 4.7: Izluščeni rezultati meritev obremenitve strežnika RDS pri simulaciji spletnih dostopov.

Metrika	Mirovanje			Aktivnost		
	Min.	Povp.	Maks.	Min.	Povp.	Maks
CPUUtilization [%]	14,75	18,38	33,33	14,75	19,68	35
NetworkReceiveThroughput [kB/s]	0,06	0,07	0,21	0,05	1,28	7,26
NetworkTransmitThroughput [kB/s]	1,17	1,31	2,34	1,18	1,54	2,82
FreeableMemory [MB]	90,89	98,41	107,36	91,28	97,08	107,82
DiskQueueDepth [no.]	0	0	0,01	0	0,01	0,09
ReadIOPS [IOPS/s]	0	0,22	1,1	0	0,28	1,47
WriteIOPS [IOPS/s]	0	0,09	0,72	0	1,92	10,63
ReadThroughput [kB/s]	0	0,14	0,68	0	0,31	1,71
WriteThroughput [kB/s]	0	0,84	12,42	0	35,42	202,41
ReadLatency [ms]	0	0,5	12	0	2,23	26
WriteLatency [ms]	0	1,43	17,14	0	2,75	13,89
DatabaseConnections [no.]		5,56			8,31	
FreeStorageSpace [MB]		4110,64			4110,64	
SwapUsage [MB]		197,52			197,52	

Tabela 4.8: Izluščeni rezultati meritev obremenitve strežnika RDS pri simulaciji uvažanja meritev TMG.



Slika 4.5: Obremenitev procesorja spletnega strežnika pri simulaciji spletnih dostopov. Točke na sliki prikazujejo povprečno vrednost. Stolpci pod piko prikazujejo minimalno vrednost, stolpci nad piko pa maksimalno.

5 Sklep

5.1 Ugotovitve

Oblak je ponudba strežniških virov v obliki storitve. Na tržišču trenutno obstaja veliko različnih oblačnih ponudb, od ozko usmerjenih rešitev s programsko opremo kot storitvijo do širokonamenskih ponudb v obliki infrastrukture kot storitve. Dve vodilni podjetji na tem področju sta Google in Amazon, ki sta poenostavili uporabo oblačnih storitev in s tem zvečali njihovo dostopnost. Z novimi modeli lahko podjetja uporabljajo že obstoječe oblačne storitve kot osnovo, na kateri lahko zgradijo svojo lastno oblačno ponudbo. Na primeru oblaka podjetja TMG-BMC smo se ukvarjali z vprašanjem, ali je tak sistem primeren tudi za mikro podjetja, ki si drugače ne bi mogla privoščiti svojih lastnih oblačnih storitev. Oblak tega podjetja je nišno usmerjen in namenjen dopolnjevanju obstoječe ponudbe in ojačanju osrednje dejavnosti podjetja. Tekom diplomskega dela smo pokazali, da lahko mikro podjetje ustvari oblačne storitve z nižjim vložkom, kar ponudniki oblačnih storitev omogočajo preko svojih naročniških modelov. Tak oblak lahko raste skupaj z manjšim podjetjem, saj se stroški prilagajajo obremenjenosti storitve. Meritve, ki so bile opravljene v sklopu tega diplomskega dela, nakazujejo, da v primeru oblaka

podjetja TMG-BMC zadoščajo že najbolj osnovne izvedbe strežnikov podjetja Amazon. Meritve so pokazale nizko obremenjenost oblačnih virov pri simulacijah uvozov podatkov in dostopov do spletnega vmesnika. Bremena v simulacijah so bila oblikovana po najbolj zahtevnem od predvidenih tipičnih uporabnikov. Na osnovi meritev lahko sklepamo, da bi uporabljeni Amazonovi strežniki lahko brez težav zadovoljili maksimalno začetno predvideno število desetih uporabnikov. Sledi, da bi podjetje TMG-BMC imelo približno 4 USD mesečnih stroškov na uporabnika iz naziva naročnin na oblačne storitve podjetja Amazon. Podjetje lahko sestavi ceno naročnine na svoje oblačne storitve s pomočjo tega podatka v kombinaciji s predvidenimi stroški nudene tehnične podpore, zelene amortizacije vložka v izdelavo oblačne programske opreme in preostalih sprotnih stroškov. Ker je uporabljen primer dokaj generičen, menim, da bi pridobljene ugotovitve lahko koristile tudi drugim majhnim podjetjem, ki bi želela ponujati svoje oblačne storitve.

5.2 Priporočila za nadaljnje raziskave

Ena od bolj preprostih razširitev meritev bi bila poleg meritve posameznega bremena tudi meritev sočasnih bremenih. Tak tip meritev bi omogočil vpogled v součinkovanje različnih bremen. Možno je, da bi bremena tekmovala med seboj in s tem povečala obremenitev virov. Strežniške tehnike predpomnjenja in druge strežniške optimizacije bi lahko zmanjšale skupno obremenitev. Zanimivo bi bilo tudi izvesti meritve z bolj raznolikimi in realističnimi tipi bremen, kar bi prispevalo k boljši oceni razpoložljivih kapacitet strežnikov v dejanski rabi. Vsekakor bi k boljšemu razumevanju problematike pomagalo izvajanje meritev skozi daljše obdobje na sistemu, ki je v dejanski uporabi. Na takem sistemu bi bilo tudi možno izvesti analize dohodkov in stroškov, ki so vezani na oblačno storitev. To je idealna tema za interdisciplinarno raziskavo na področjih računalništva in ekonomije.

LITERATURA

- [1] 9 Worst Cloud Security Threats - InformationWeek, InformationWeek, 2016. [Online]. Dosegljivo:
<http://www.informationweek.com/cloud/infrastructure-as-a-service/9-worst-cloud-security-threats/d/d-id/1114085>. [Dostopano 14. 6. 2016].
- [2] Amazon CloudWatch - Cloud & Network Monitoring Services, Amazon Web Services, Inc., 2016. [Online]. Dosegljivo:
<https://aws.amazon.com/cloudwatch/>. [Dostopano 27. 6. 2016].
- [3] Amazon Elastic Block Store (EBS) - Block Storage, Amazon Web Services, Inc., 2016. [Online]. Dosegljivo:
<https://aws.amazon.com/ebs/>. [Dostopano 27. 6. 2016].
- [4] Amazon Elastic Compute Cloud Dimensions and Metrics - Amazon CloudWatch, Docs.aws.amazon.com, 2016. [Online]. Dosegljivo:
<http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/ec2-metricscollected.html>. [Dostopano 27. 6. 2016].
- [5] Amazon RDS Dimensions and Metrics - Amazon CloudWatch, Docs.aws.amazon.com, 2016. [Online]. Dosegljivo:
<http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/rds-metricscollected.html>. [Dostopano 27. 6. 2016].
- [6] Amazon RDS Pricing – Amazon Web Services (AWS), Amazon Web Services, Inc., 2016. [Online]. Dosegljivo:
<https://aws.amazon.com/rds/pricing/>. [Dostopano 27. 6. 2016].
- [7] Amazon Virtual Private Cloud (VPC) – Amazon Web Services, Amazon Web Ser-

- vices, Inc., 2016. [Online]. Dosegljivo:
<https://aws.amazon.com/vpc/>. [Dostopano 14. 6. 2016].
- [8] AngularJS, Docs.angularjs.org, 2016. [Online]. Dosegljivo:
<https://docs.angularjs.org/guide/introduction>. [Dostopano 27. 6. 2016].
- [9] App Engine Pricing, Google Developers, 2016. [Online]. Dosegljivo:
<https://cloud.google.com/appengine/pricing>. [Dostopano 14. 6. 2016].
- [10] AWS - Amazon Relational Database Service (RDS), Amazon Web Services, Inc., 2016. [Online]. Dosegljivo:
<https://aws.amazon.com/rds/>. [Dostopano 27. 6. 2016].
- [11] AWS | Amazon CloudWatch | Pricing, Amazon Web Services, Inc., 2016. [Online]. Dosegljivo:
<https://aws.amazon.com/cloudwatch/pricing/>. [Dostopano 27. 6. 2016].
- [12] Cenik storitev Inštituta za šport, Fakultetazasport.si, 2013. [Online]. Dosegljivo:
http://www.fakultetazasport.si/mma_bin.php?id=20130308125846. [Dostopano 14. 6. 2016].
- [13] ChromeDriver - WebDriver for Chrome, Sites.google.com, 2016. [Online]. Dosegljivo:
<https://sites.google.com/a/chromium.org/chromedriver/>. [Dostopano 27. 6. 2016].
- [14] CLIENTS « TMG, Tmg-bodyevolution.com, 2016. [Online]. Dosegljivo:
<http://www.tmg-bodyevolution.com/about/clients/>. [Dostopano 14. 6. 2016].
- [15] Cross-VM side-channel attacks: How to defend cloud infrastructures, SearchCloud-Security, 2016. [Online]. Dosegljivo:
<http://searchcloudsecurity.techtarget.com/tip/Cross-VM-side-channel-attacks-How-to-defend-cloud-infrastructures>. [Dostopano 14. 6. 2016].
- [16] Dropbox - Tour, Dropbox, 2016. [Online]. Dosegljivo:
<https://www.dropbox.com/tour/>. [Dostopano 14. 6. 2016].
- [17] D. Dugdale, R. Epstein and S. Pantilat, *Time and the patient-physician relationship*. J Gen Intern Med, št. 14, zv. 1 1999, str. 34–40.

- [18] EC2 Instance Types – Amazon Web Services (AWS), Amazon Web Services, Inc., 2016. [Online]. Dosegljivo:
<https://aws.amazon.com/ec2/instance-types/>. [Dostopano 27. 6. 2016].
- [19] Elastic Compute Cloud (EC2) Cloud Server & Hosting - AWS, Amazon Web Services, Inc., 2016. [Online]. Dosegljivo:
<https://aws.amazon.com/ec2/>. [Dostopano 27. 6. 2016].
- [20] Football (Soccer) in England and Wales: How many matches are there in the Premier League season?, quora.com, 2016. [Online]. Dosegljivo:
<https://www.quora.com/Football-Soccer-in-England-and-Wales-How-many-matches-are-there-in-the-Premier-League-season>. [Dostopano 14. 6. 2016].
- [21] General Python FAQ — Python 3.5.2 documentation, Docs.python.org, 2016. [Online]. Dosegljivo:
<https://docs.python.org/3/faq/general.html#what-is-python-good-for>. [Dostopano 27. 6. 2016].
- [22] Google Cloud Platform Documentation, Google Developers, 2016. [Online]. Dosegljivo:
<https://cloud.google.com/docs/>. [Dostopano 14. 6. 2016].
- [23] Hamstring Strain Treatment & Management: Rehabilitation Program, Surgical Intervention, Emedicine.medscape.com, 2016. [Online]. Dosegljivo:
<http://emedicine.medscape.com/article/307765-treatment>. [Dostopano 14. 6. 2016].
- [24] Informacijski pooblaščenec, “Varstvo osebnih podatkov & računalništvo v oblaku”, *Verzija 1.0*, 2012.
- [25] Is Cloud Computing Really Cheaper? [Online]. Dosegljivo:
<http://www.forbes.com/sites/reuencohen/2012/08/03/is-cloud-computing-really-cheaper/>. [Dostopano 14. 6. 2016].
- [26] JSON, Json.org, 2016. [Online]. Dosegljivo:
<http://www.json.org/>. [Dostopano 27. 6. 2016].
- [27] A. Mazrekaj, I. Shabani and B. Sejdiu. “Pricing Schemes in Cloud Computing: An Overview”, v zborniku: International Journal of Advanced Computer Science and Applications, vol. 7, no. 2, 2016, str. 80–86.

- [28] Modifying a DB Instance and Using the Apply Immediately Parameter - Amazon Relational Database Service, Docs.aws.amazon.com, 2016. [Online]. Dosegljivo: <http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Overview.DBInstance.Modifying.html>. [Dostopano 27. 6. 2016].
- [29] Muscle Strains, Healthline, 2016. [Online]. Dosegljivo: <http://www.healthline.com/health/strains#Outlook7>. [Dostopano 14. 6. 2016].
- [30] P. Krill, A developer's guide to the pros and cons of Python, InfoWorld, 2016. [Online]. Dosegljivo: <http://www.infoworld.com/article/2887974/application-development/a-developer-s-guide-to-the-pro-s-and-con-s-of-python.html>. [Dostopano 27. 6. 2016].
- [31] Prva slovenska nogometna liga 2015/16, Sl.wikipedia.org, 2016. [Online]. Dosegljivo: https://sl.wikipedia.org/wiki/Prva_slovenska_nogometna_liga_2015/16. [Dostopano 14. 6. 2016].
- [32] Quick Start WordPress for Google App Engine by GoogleCloudPlatform, Googlecloudplatform.github.io, 2016. [Online]. Dosegljivo: <https://googlecloudplatform.github.io/appengine-php-wordpress-starter-project/>. [Dostopano 14. 6. 2016].
- [33] Representational state transfer, En.wikipedia.org, 2016. [Online]. Dosegljivo: https://en.wikipedia.org/wiki/Representational_state_transfer. [Dostopano 14. 6. 2016].
- [34] Requests: HTTP for Humans — Requests 2.10.0 documentation, Docs.python-requests.org, 2016. [Online]. Dosegljivo: <http://docs.python-requests.org/en/master/#>. [Dostopano 27. 6. 2016].
- [35] Resizing Your Instance - Amazon Elastic Compute Cloud, Docs.aws.amazon.com, 2016. [Online]. Dosegljivo: <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-resize.html>. [Dostopano 27. 6. 2016].
- [36] Revealed: The Average Ages of All 20 Premier League Squads, 90min.com, 2015. [Online]. Dosegljivo:

<http://www.90min.com/posts/2405074-revealed-the-average-ages-of-all-20-premier-league-squads>. [Dostopano 14. 6. 2016].

- [37] G. van Rossum, *An Introduction to Python for Unix/C Programmers*. NLUUG na-jaarsconferentie, Dutch UNIX users group, št. 1993 1993.
- [38] Safe Harbor, Ip-rs.si, 2016. [Online]. Dosegljivo:
<https://www.ip-rs.si/varstvo-osebni-podatkov/obveznosti-upravljavcev/iznos-osebni-podatkov-v-tretje-drzave/safe-harbor/>. [Dostopano 14. 6. 2016].
- [39] Selenium - Web Browser Automation, Seleniumhq.org, 2016. [Online]. Dosegljivo:
<http://www.seleniumhq.org/>. [Dostopano 27. 6. 2016].
- [40] What Are Amazon CloudWatch, Amazon CloudWatch Events, and Amazon CloudWatch Logs? - Amazon CloudWatch, Docs.aws.amazon.com, 2016. [Online]. Dosegljivo:
<http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/WhatIsCloudWatch.html>. [Dostopano 27. 6. 2016].
- [41] L. Youseff, M. Butrico in D. Da Silva *Toward a Unified Ontology of Cloud Computing*. 2008 Grid Computing Environments Workshop, 2008, str. 1–10.