Računanje u oblaku korištenjem pythona

Dragunić, Marko

Master's thesis / Diplomski rad

2014

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:217:526507

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Repository of the Faculty of Science - University of Zagreb

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Marko Dragunić

RAČUNANJE U OBLAKU KORIŠTENJEM PYTHONA

Diplomski rad

Voditelj rada:
izv. prof. dr. sc. Luka Grubišić

Zagreb, rujan, 2014

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Ovaj rad posvećujem svima koji su bili prisutni kroz moje školovanje, u trenutcima kada je bilo najteže bili uz mene. Svima njima dugujem jedno veliko Hvala, a posebnu zahvalu dugujem svojim roditeljima koji su pokazali bezgraničnu podršku i ljubav tijekom cijelog školovanja. Ne postoje riječi kojima to mogu predočiti, stoga ovaj rad je poklonjen Vama.

Sadržaj

Sadržaj iv

Uvod **1**

1 Računarstvo u oblaku **4**
1.1 Virtualni strojevi kao standardni razvojni objekt 5
1.2 Građa Cloud computinga 6
1.3 Sudionici računarstva u oblaku 13
1.4 Što, kada i kako spremati u oblak? 15
1.5 Oblaci 16

2 Mrjob i PiCloud **19**
2.1 MapReduce 19
2.2 Hadoop 21
2.3 Mrjob 22
2.4 PiCloud 25

3 Financijska strana i sigurnost **27**
3.1 Ekonomski aspekt 27
3.2 Sigurnost 29
3.3 Povijest i budućnost 33

Bibliografija **35**

Uvod

U današnjem svijetu sve je zapisano, o svemu se vode podatci. Vjerojatno ne postoji osoba koja može sa sigurnošću odgovorit kolika je ukupna količina zapisanih podataka. Pohrana tolike količine podataka zahtjeva ogromne resurse. Prije par godina resursi za pohranu činili su se neograničenim, no međutim, već u novije vrijeme čak i lokalni serveri su postali premaleni. Počela se javljati potreba za novom tehnologijom, novim konceptom pohrane.

Jedna od mogućnosti kako se mogu zamijeniti lokalni resursi s nečim što je brže, veće i produktivnijeje koncept računarstva u oblaku (eng. Cloud computing). U nastavku ću često koristiti izraz Cloud computing umjesto računarstva u oblaku jer se fraza toliko popularizirala da se u današnjem tehničkim diskusijama slobodno i uobičajeno koristi engleski izraz.

Koncept Cloud computing je dobro prihvaćen među korisnicima i u skladu s tim se konstantno povećava broj korisnika. Usluga se korisnicima dostavlja putem Interneta i plaća se onoliko koliko se koristi. Zanimljivo je napomenuti da prave definicije Cloud computinga nema. Svoje (donekle različite) verzije definicije Cloud computinga imaju i IT stručnjaci, analitičari, kao i sami korisnici.

Najčešće verzije su:

- *mogućnost iznajmljivanja jednog ili više poslužitelja te pokretanje različitih aplikacija na njima*
- *mogućnost iznajmljivanja virtualnog poslužitelja gdje korisnici mogu pohranjivati i pristupati podacima*
- *jeftiniji način korištenja programskih rješenja koji će se unajmljivati po potrebi*
- *korištenje aplikacije koja se nalazi na Internetu koja pohranjuje i štiti podatke za vrijeme pružanja usluge*

Pojavom koncepta Cloud computing-a inženjeri prestaju brinuti o sklopovlju, podršci za aplikacije, sigurnosti i softveru. Mogu svu koncentraciju prebaciti na logistiku, i to je ono čemu se teži u današnjem računarstvu. Sve te nove ideje moraju

zainteresirati buduće programere, stoga je ideja ovog rada da se što više upoznamo sa ovom novom tehnologijom i konceptom Cloud computing-a.

U prvom dijelu dana je arhitektura Cloud computinga. Navedeni su modeli Cloud computing-a te razrađeni pojedini modeli s prednostima i nedostacima. Razrađeni su modeli izvedbe i predočene razlike među njima. Predočena je i struktura sudionika računarstva u oblaku te njihova uloga. Izneseno je i vlastito stajalište o tome kako i u kojem slučaju se preporučuje prijelaz na Cloud computing.

U drugom djelu u kontekstu model primjera prezentirat ćemo način na koji se može razviti aplikacija korištenjem koncepta i slobodno dostupnih usluga Cloud computing-a. U ovom dijelu razrađeni su MapReduce poslovi, koji su usko vezani sa Cloud computing-om, njihovo funkcioniranje i kreiranje. Objašnjeno je kreiranje računa u oblaku, pokretanje procesa i potrebitih alata na konkretnom primjeru Amazon Clouda. Navedeno je par primjera i dane slike u vidu lakšeg snalaženje i povezivanja s oblakom.

U trećem dijelu obrađeni su konkretni troškovi usluga. Navedene su najveće prijetnje Cloud computing-u i pitanja sigurnosti. Na kraju je razrađena povijest razvoja računarstva u oblaku te očekivani smjer razvoja u budućnosti.

Na kraju ovog uvodnog dijela navest ćemo prednosti i nedostatke Cloud computinga:

Prednosti su:

- *niža cijena sklopovske podrške*
- *korisniku je uvijek dostupna posljednja i najnovija inačica programske podrške*
- *podaci su dostupni sa svakog računala s kojeg korisnik ima pristup Internetu*
- *manji troškovi održavanja i nadogradnje programske podrške*
- *u uslugu je uključena profesionalna antivirusna zaštita*
- *dostupnost aplikacija*
- *mogućnost opsluživanja velikog broja korisnika*
- *fleksibilnost u izmjeni i prilagodbi aplikacija.*

Nedostatci su:

- *problem dostupnosti*

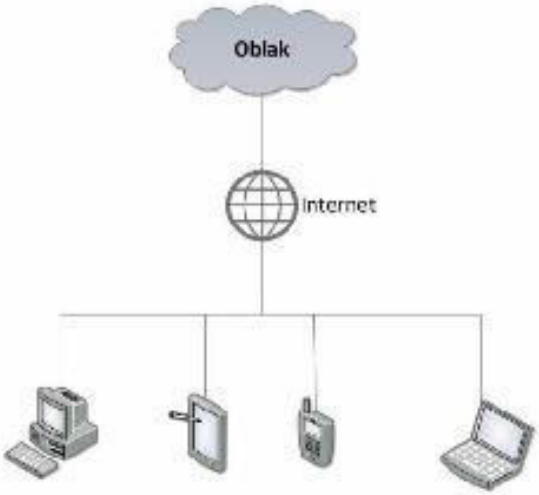
- *problem sigurnosti*
- *problem ovisnosti o jednom pružatelju usluge*
- *usvajanje novog načina razvoja aplikacija*
- *komplicirano premještanje postojećih aplikacija*
- *problem povezivanja aplikacija različitih proizvođača te razmjena i premještanje podataka između oblaka različitih proizvođača.*

Poglavlje 1

Računarstvo u oblaku

Oblak (engl. cloud) je vrlo često korištena metafora za Internet. Ona potječe od načina na koji se Internet označava na raznim dijagramima, ali i od same infrastrukture Interneta. Kada se ona koristi zajedno sa izrazom računarstvo (engl. computing) novi termin više nema isto značenje. Postoji mnogo definicija računarstva u oblaku, ali svima je zajednička ideja iznajmljivanja računalnih resursa po potrebi. **Možda najbolja definicija bi bila da je računarstvo u oblaku koncept podjele programskog okruženja koji koristi Internet kao platformu te omogućuje da aplikacije i dokumenti poslani iz bilo kojeg dijela svijeta budu pohranjeni i čuvaju se za to predviđenim poslužiteljima.** Jedan od glavnih razloga sve veće popularnosti računarstva u oblaku su smanjeni troškovi ulaganja u računalnu infrastrukturu. IT uvijek teži povećanju kapaciteta ili novih mogućnosti bez obuke novog osoblja, kupovine novih licenciranih programa ili bilo kakvih dodatnih troškova.

Organizacije koje pružaju uslugu računarstva u oblaku korisnicima omogućuju pristupanje aplikacijama i virtualnim poslužiteljima preko web preglednika.



1.1 Računarstvo u oblaku

Kod računarstva u oblaku uvodi se novi model plaćanja usluge. Usluga se plaća onoliko koliko se koristi (engl. pay-per-use). Korisnik sa svojim uređajem putem Interneta može s bilo kojeg mjesta pristupati podacima koji se nalaze u oblaku. Za ovakvu vrstu usluge koristi se pojam oblak zato što korisnici ne znaju gdje se nalaze poslužitelji ni gdje se izvršavaju aplikacije. Tvrtke koje koriste tradicionalan pristup poslovanja sve svoje potrebe za računalnim resursima i aplikacijama zadovoljavaju tako što koriste resurse koji su u njihovom vlasništvu. Za razliku od tradicionalnog pristupa, poslovanje koje se temelji na korištenju usluga zasnovanih na računarstvu u oblaku uvodi modele korištenja resursa koji nisu u vlasništvu tvrtke.

Na temelju mnogih istraživanja ugrubo se može reći što je sve računarstvo u oblaku:

- **MSP** (engl. managed service providers) je jedan od najstarijih oblika računarstva u oblaku. MSP je aplikacija namijenjena IT službi, a ne krajnjim korisnicima. Najbolji primjer toga je skeniranje zloćudnih programa koji se šire *email*-om.

- **SaaS** (engl. Software as a Service) i **PaaS** (engl. Platform as a service) bit će objašnjeni kasnije

- **Uslužno računarstvo** (engl. Utility computing) je relativno nova forma na tržištu informacijskih tehnologija. Pružatelj usluga osigurava računalne resurse i infrastrukturu korisniku prema potrebi. Koriste ga Amazon, Sun, IBM i mnogi drugi. U budućnosti bi ovaj model mogao zamijeniti dio baza podataka.

- **Web usluge u oblaku** (engl. Web services in the cloud) su usko povezane sa SaaS modelom. To su organizacije koje nude sučelja koja razvojnim inženjerima omogućava iskorištavanje funkcionalnosti preko Interneta. Najpoznatije web usluge su Google Maps te standardne usluge obrade kreditnih kartica.

- **Usluge komercijalnih platformi** (engl. Service commerce platforms) su hibrid SaaS i MSP modela. Usluga komercijalnih platformi nudi čvorište s kojim korisnici komuniciraju. Najčešće se upotrebljava u web trgovinama, a najpoznatiji primjer davatelja ove usluge je Rearden Commerce.

- **Integracija Interneta** (engl. Internet integration) je način međusobnog povezivanja nekog oblaka sa mnoštvom drugih izoliranih oblaka. Možda bolji naziv za ovu uslugu je sky computing. Integracija cloud computinga je još uvijek u ranim fazama razvoja pa tako konkretnije usluge možemo očekivati u budućnosti.

1.1 Virtualni strojevi kao standardni razvojni objekt

Virtualni strojevi u posljednjih nekoliko godina postali su standardni razvojni objekt. Ono što virtualni strojevi omgućavaju je puno veću prilagodljivost strojeva budući da

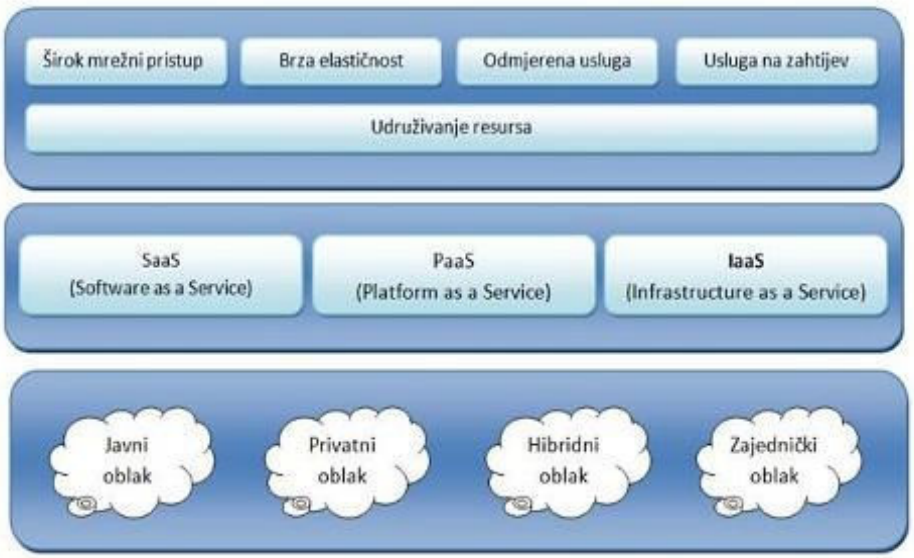
sklopovlje čini apstraktnim. Strojevi postaju apstraktni do te granice da je moguće da se programi više puta raspoređuju bez izravne vezanosti za specifičnog fizičkog poslužitelja. Virtualizacija omogućuje da se svi procesi obavljaju dinamički, od pohrane podataka do aplikacija za računanje. Tako je omogućeno bolje iskorištavanje ogromnih resursa koje fizički poslužitelji osiguravaju.

Virtualni strojevi postali su prevladavajuća apstrakcija. Njihovim korištenjem kao razvojnih objekata zadovoljava se velika većina korisničkih potreba, a i omogućuje brzo pregrupiranje i skaliranje aplikacija.

Oblaci za računanje su najčešće fizički povezani s oblacima za pohranu podataka. Oblaci za pohranu podataka pružaju virtualnu pohranu preko API-a (Application Programming Interface) koja olakšava virtualnim strojevima pohranu slika, datoteka, aplikacija...

1.2 Građa Cloud computinga

Američka vladina organizacija NIST koja se bavi standardima i tehnologijom se između ostaloga bavi i definiranjem i modelima cloud computinga. Cloud model promovira dostupnost i sastoji se od pet ključnih karakteristika, tri modela pružanja usluga te četiri modela izvedbe koja su navedena na slici.



1.2 Ključne karakteristike, modeli pružanja usluga, modeli implementacije

Ključne karakteristike cloud computinga

Postoji pet ključnih karakteristika koje pokazuju odnos i razlike cloud computing sustava u odnosu na tradicionalni pristup u računarstvu. To su pružanje usluge na zahtjev korisnika, širok mrežni pristup, udruživanje resursa, brza elastičnost i odmjerene usluge.

1.3 Modeli pružanja usluga

- Pružanje usluge na zahtjev korisnika (engl. On-demand self service) - Korisnik može samostalno odabrati i pokrenuti računalne resurse, ne mora birati vrijeme niti mrežni prostor za pohranu podataka jer nije potrebna interakcija s djelatnicima pojedinih davatelja usluga. Korisnik plaća uslužitelju usluge u ovisnosti o vremenu i obujmu koliko ih koristi. Ova usluga može se smatrati kao najam opreme koja se plaća ovisno o tome koliko je opreme, na koje vrijeme i s kojim uslugama je iznajmljeno.

Virtualizacija je ključ ovog modela. Organizacije koje koriste informacijske tehnologije shvaćaju da im virtualizacija omogućava brzo i jednostavno stvaranje kopija postojećih okolina, ponekad uključujući više virtualnih strojeva kako bi podržala ispitivanja, razvoj i pohranu aktivnosti. Trošak ovih okolina je jako malen jer one mogu postojati na istom poslužitelju.

Isto tako, nove aplikacije se mogu razvijati i rasprostraniti u novim virtualnim strojevima na postojećim fizičkim poslužiteljima otvorenima za uporabu preko Interneta. Aplikacije mogu biti skalirane ako su uspješne na tržištu.

- **Širok mrežni pristup (**engl. Broad network access) - ideja je da se s bilo kojim uređajem može spojiti na sustav nevezano za mjesto. Mogućnosti su dostupne putem mreže i njima se pristupa koristeći standardne mehanizme.

- Udruživanje resursa (engl. Resource pooling) - računalni resursi pružatelja usluga spajaju se kako bi poslužili sve korisnike koristeći model više zakupljenih jedinica, s različitim fizičkim i virtualnim resursima, koji se dinamički dodjeljuju i uklanjaju prema zahtjevima korisnika. Korisnik uobičajeno nema nadzor i znanje o točnom mjestu uporabljenih resursa, ali ga ipak može odrediti na većoj razini apstrakcije.

- **Brza elastičnost** (engl. Rapid elasticity) -cloud computing nudi korisnicima mogućnosti koje mogu biti elastično i ubrzano pokrenute, a ponekad i automatski, ovisno o potrebi za smanjenjem ili povećanjem mogućnosti. Krajnji korisnik ima dojam da su mogućnosti koje koristi beskonačne i da ih može kupiti bilo kada u bilo kojoj količini.

- Odmjerena usluga (engl. Measured service) - sustavi koji koriste cloud computing automatski provjeravaju i optimiziraju uporabu resursa. Uporaba resursa se može pratiti, provjeravati i o njoj je moguće raditi izvješća te tako imati detaljan uvid davateljima usluge i korisnicima.

Modeli cloud computinga

1.4 Modeli pružanja usluge

Cloud computing razvija se po modelu *sve kao usluga* (engl. everything-as-a-service). Virtualni strojevi, infrastruktura, posrednička platforma i aplikacije koriste se kao usluge u oblaku.

Postoje tri osnovna modela usluge računarstva u oblaku(često se i nazivaju SPI model, pri čemu SPI označava program, platformu ili infrastrukturu):

- Softver kao usluga (engl. Software as a Service) kratica SaaS
- Platforma kao usluga (engl. Platform as a Service) kratica PaaS
- Infrastruktura kao usluga (engl. Software as a Service) kratica IaaS.

Svaki model se razlikuje po vrsti usluga koje pruža korisniku.

- **SaaS** model omogućuje pristup aplikacijama i uslugama putem Interneta bez prethodne instalacije na korisničko računalo. Održavanje aplikacije je odgovornost pružatelja SaaS usluge. Aplikacije su dostupne s različitih klijentskih uređaja uz pomoć klijentskog sučelja (npr. web preglednik). SaaS omogućuje dostupnost aplikacija putem Interneta u obliku usluge koja se unajmljuje prema potrebi, umjesto da korisnik kupi zasebni program koji treba instalirati na vlastitom računalu.

Korištenjem SaaS modela tvrtka se suočava sa manjim financijskim rizikom jer se najam softvera u oblaku temelji na modelu plaćanja "plati koliko koristiš" što je prednost ove usluge s obzirom da tako tvrtka ne treba unaprijed trošiti novac na opremu i licencu ako nije sigurno da će taj softver imati predviđenu vrijednost, već plaća mjesečnu preplatu za korištenje usluge. Usluga se može prestati koristiti u bilo kojem trenutku. Uz to, potrebno je kraće vrijeme do početka korištenja usluge, tj. nije potrebna lokalna instalacija te u startu tvrtka može uštedjeti vrijeme i novac. Saas model uvijek nudi najnoviju inačicu softvera što je također prednost korištenja ove usluge jer se nadogradnja softvera izvršava u oblaku.

Jedan od problema korištenja SaaS usluge su pravna i regulatorna pitanja. U nekim državama, uključujući i Hrvatsku, zakon zabranjuje pohranu podataka izvan državnih granica. Neki od pružatelja usluga imaju podatkovne centre smještene u više država pa tako korisnici usluga koriste resurse podatkovnih centara koji se možda ne nalaze u njihovoj državi.

Danas najpoznatiji pružatelj raznih SaaS usluga je Google. Google nudi niz usluga pod nazivom Google Apps, a tu su uključene usluge Gmail, Google Groups, Google Calendar, Google Talk, Google Docs,... Tvrtka SAP nudi aplikaciju za planiranje resursa poduzeća Bussiness ByDesign. IBM nudi aplikaciju Lotus za međusobnu suradnju zaposlenika unutar poduzeća i razne druge poznate tvrtke nude razne SaaS usluge.

- **PaaS** model pruža okruženje za razvijanje i upravljanje aplikacija. Korisnicima se nude razvojni resursi uključujući virtualna računala, razvojne okvire i operacijske sustave. Pružatelj PaaS usluge održava infrastrukturu oblaka, operacijske sustave i sav softver koji služi kao podrška korisničkim aplikacijama, dok je korisnik PaaS usluge odgovoran za razvijanje, instaliranje i održavanje vlastite aplikacije. Platforma u oblaku omogućuje korisnicima jednostavan pristup resursima, a naplaćuju se samo oni resursi koje aplikacija stvarno koristi. Korisnici ne stvaraju izravno vlastite virtualne strojeve, nego na postojećoj platformi razvijaju aplikaciju koja se na toj platformi može izvršavati. Dakle, sve što korisnik treba napraviti jest na već postojećoj platformi stvoriti i popuniti bazu podataka, učitati aplikaciju i pokrenuti je. Za ostale stvari poput raspoređivanja, opterećivanja i ažuriranja operacijskog sustava na virtualnim strojevima odgovoran je pružatelj usluge. Pri korištenju PaaS modela korisnik može brže razvijati softver jer dio posla vezan uz platformu obrađuje pružatelj usluge. Također, troškovi korištenja ovog modela su niži jer je manje posla za koji je zadužena tvrka pa je tako manja i potreba za osobljem.

Nedostatak PaaS modela je to što smanjuje ovlasti i kontrolu korisnika. Usluga nameće ograničenja kao što je platforma na kojoj korisnik radi, unaprijed je određena što ne odgovara korisnicima koji preferiraju sami kontrolirati i modificirati okruženje. Smanjena mogućnost migracije još je jedan od nedostataka korištenja ove usluge. Obzirom na to da okruženje pružatelja PaaS usluge nije isto kao i okruženje tvrtke, mogućnost prenošenja postojeće aplikacije u oblak je smanjena. Isto tako, nije jednostavno promijeniti pružatelja PaaS usluge jer migracija nije moguća ili je preskupa pa bi tako korisnik trebao ostati kod određenog pružatelja usluge.

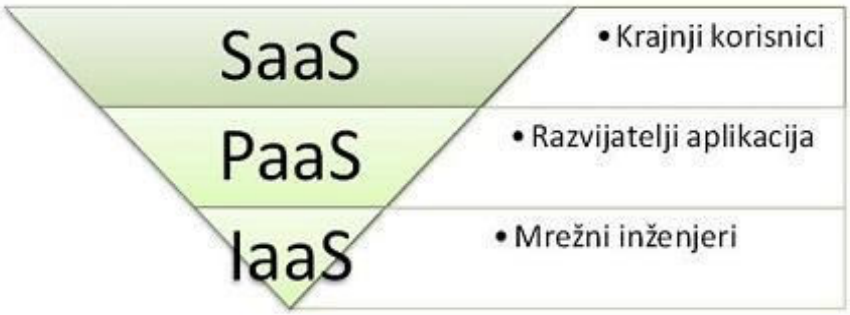
Iz ovog se može zaključiti da je PaaS model dobro rješenje u slučajevima za koje je određeno okruženje namijenjeno, no nije toliko prikladan za široku uporabu poput IaaS modela pa se stoga i manje koristi.

Primjeri PaaS usluga su Google - AppEngine, Amazon Web Service - Elastic Beanstalk. Microsoft - Windows Azure i Salesforce - AppForce.

-**IaaS** model označava skup računalnih, memorijskih i mrežnih resursa oblaka. Pružatelji IaaS usluge upravljaju čitavom infrastrukturom, dok je korisnik odgovoran za ostale aspekte implementacije usluga. Korisnik može unajmiti čitava fizička ili virtualna računala te mu se pruža i mogućnost pohrane podataka.

Pružatelj usluge korisniku omogućuje pristup virtualnim strojevima te korisnik odabire virtualni stroj za bazu podataka i virtualni stroj na kojima će se aplikacija izvršavati. Kod IaaS modela korisnik sam upravlja uravnoteženjem opterećenja pa prema tome odabire i broj virtualnih strojeva koje će koristiti. Nakon odabira

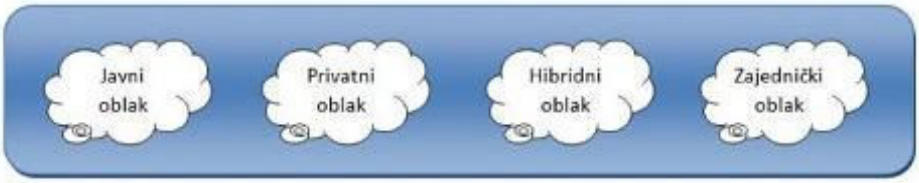
virtualnih strojeva korisnik stvara i puni bazu podataka, a zatim na virtualne strojeve instalira svoju aplikaciju. Za eventualnu daljnju nadogradnju virtualnih strojeva i administraciju brine se korisnik.



1.5 Karakteristike pružatelja usluge u oblaku

Neprekidne inovacije dovode do brzog razvoja različitih novih rješenja. Razvojem sučelja korisnici i davatelji usluga moći će uživati u različitim metodama interakcije sa cloud computing uslugama. Tim razvojem povećat će se broj konzultanata te će se olakšati komunikacija sa korisnicima i lakše će se ostvarivati njihovi zahtjevi (npr. koji model najbolje odgovara njihovim potrebama).

Modeli izvedbe ili primjene



1.6 Modeli izvedbe Računarstva u oblaku

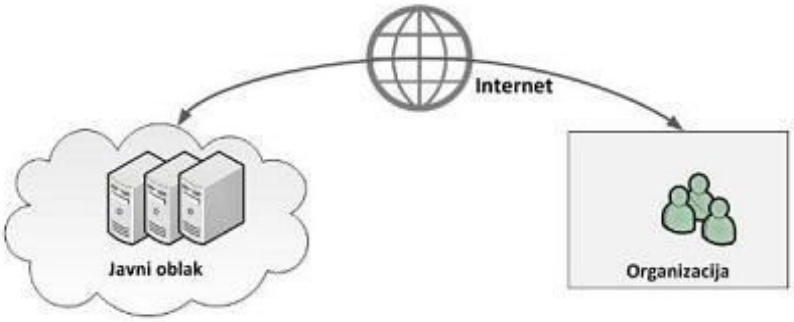
Neovisno o modelima pružanja usluga postoje četiri različita modela provođenja cloud computing usluga. Svaki od njih je izveden drugačije ovisno o potrebama. Detaljan opis u nastavku teksta.

- Javni oblak (engl. public cloud) je u vlasništvu tvrtke koja daje u najam usluge u oblaku. Oblak je dostupan i otvoren za javnost bez obzira radi li se o pojedincu ili o organizaciji. Kada su u pitanju javne platforme postavlja se pitanje sigurnosti vlastitih podataka.

Aplikacije različitih korisnika često se nalaze na istim poslužiteljima, sustavima za pohranjivanje i mrežama. Javni oblaci smanjuju troškove i sigurnosne rizike pružanjem promjenjive infrastrukture. Oni čine privremeno zakupljenu infrastrukturu organizacija.

Ako je javni oblak realiziran s pažnjom usmjerenom na izvedbu, sigurnost i položaj podataka, druge aplikacije pokrenute na oblaku ne bi trebale stvarati probleme prema arhitekturi oblaka i krajnjim korisnicima. Značajna prednost javnog oblaka je da oni mogu biti puno veći od privatnih oblaka. Javni oblaci nude mogućnost povećanja ili smanjivanja zakupljenog dijela oblaka. Postoji i mogućnost prebacivanja odgovornosti s organizacija na davatelja usluga u slučaju pojavljivanja neplaniranih rizika. Premda promjenjiva infrastruktura (korisnik ne zna gdje je smješten oblak) smanjuje sigurnosne rizike, ipak sigurnost je i dalje najveći problem javnih oblaka.

Korisnik može zakupiti dijelove javnog oblaka pa tako on može postati privatni podatkovni centar. Korisnik koji zakupljuje podatkovne centre ima uvid u infrastrukturu oblaka, dok obični korisnici imaju samo djelomično.

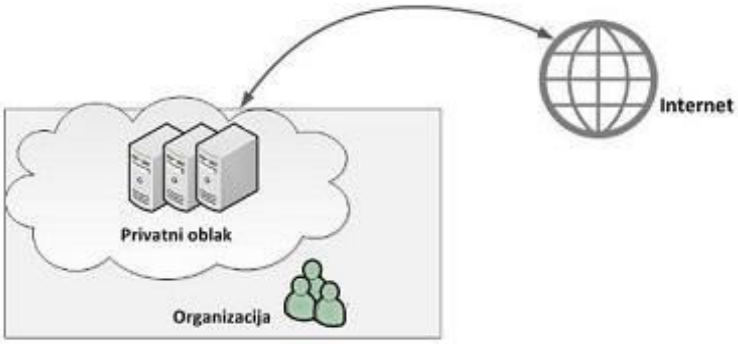


1.7 Javni oblak

- **Privatni oblak** (engl. private cloud) koriste organizacije u slučaju kada žele veći nadzor nad podacima nego što ga mogu imati pri korištenju javnog oblaka. Infrastruktura privatnog oblaka dostupna je samo jednoj organizaciji. Njome može upravljati sama organizacija ili netko drugi. Organizacije koriste privatne oblake kada trebaju ili žele veći nadzor nad podacima.

Privatni oblaci su napravljeni isključivo za uporabu jednog klijenta. Korisnik tada ima najveću sigurnost i nadzor nad podacima na vlastitoj infrastrukturi i može sam upravljati strukturom u oblaku. Organizacija posjeduje infrastrukturu i ima nadzor nad raspodjelom aplikacija na njoj. Privatni oblaci mogu biti raspoređeni i unutar organizacijskog podatkovnog centra.

Davatelji usluga grade privatne oblake i upravljaju njima. Organizacije koje posjeduju privatne oblake na njima mogu instalirati programe, aplikacije, pohranjivati podatke, upravljati strukturom te pružaju kompanijama visoku razinu nadzora nad korištenjem resursa oblaka.

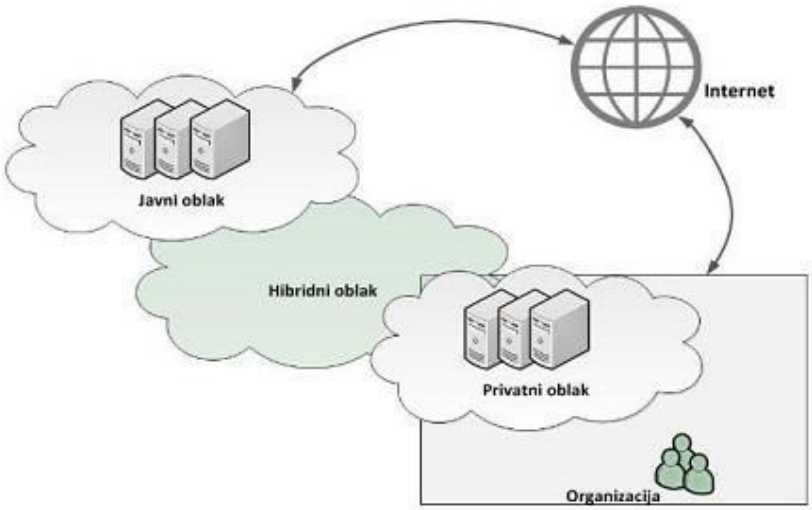


1.8 Privatni oblak

- Zajednički oblak (engl. community cloud) je oblak gdje nekoliko organizacija dijeli strukturu oblaka. Najčešće se koristi kada nekoliko organizacija ima zajedničke potrebe, misije, zahtjeve sigurnosti i slično.

- Hibridni oblak (engl. hybrid cloud) čini kombinaciju privatnih i javnih oblaka (dva ili više). Dio aplikacije se pohranjuje u javni oblak, dok se ostatak sustava nalazi u privatnom oblaku. Međusobno su povezani standardiziranim ili prikladnim tehnologijama koje omogućuju efikasan prijenos aplikacija ili podataka. Mogućnost proširivanja privatnog oblaka s resursima javnog oblaka može se koristiti za održavanje usluga pod velikim opterećenjem. Primjer toga je uporaba oblaka za pohranu podataka kako bi podržali Web 2.0 aplikacije.

Najveći problem s kojim se susreću hibridni oblaci je kako raspodjeliti aplikacije po javnom i privatnom oblaku.



1.9 Hibridni oblak

Uslijed povećanja zahtjeva korisnika i prilika na tržištu pojavljuju se potrebe za uvođenjem novih cloud modela. Premda ima novih, za sada su najpopularnija ova četiri koja smo naveli.

Kod dizajniranja oblaka dizajneri trebaju paziti na arhitekturalno razmještanje podataka jer razmještaj podataka ima velik utjecaj na buduću prilagodljivost, sigurnost i mobilnost podataka.

1.3 Sudionici računarstva u oblaku

Usluge pružanja računarstva u oblaku se mogu podijeliti u tri vrste:

- pružatelj usluge u oblaku
- korisnik oblaka
- krajnji korisnik

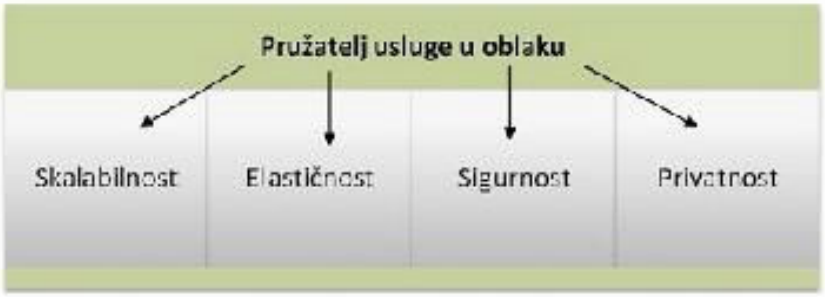
Pružatelj usluge u oblaku

Pružatelj usluge u oblaku posjeduje i upravlja računalnim resursima u oblaku kako bi se usluga dostavila korisnicima. Odgovorni su za usluge koje se koriste u oblaku kao i za cijene korištenja oblaka kao usluge.

Za SaaS model pružanja usluge pružatelj postavlja, konfigurira, održava i ažurira rad aplikacija na infrastrukturi u oblaku tako da su određeni resursi rezervirani za korisnika oblaka. Pružatelj usluge preuzima većinu odgovornosti u upravljanju i kontroli aplikacije i infrastrukture, dok korisnici oblaka imaju ograničen administrativni pristup aplikaciji.

Kod PaaS modela, pružatelj usluge upravlja računalnom infrastrukturom za platformu i pokreće aplikaciju u oblaku koja pruža komponente platforme kao što je trajanje i izvođenje aplikacije, baze podataka i ostalih posredničkih sustava. Korisnik oblaka upravlja aplikacijom, ali nema pristup infrastrukturi na nižim slojevima mreže.

Pružatelj usluge za IaaS model ima fizičke računalne resurse kao što su mreža, pohrana i poslužitelji. Korisnik oblaka ima pristup osnovnim oblicima računalnih resursa, a samim time i veću kontrolu nad komponentama aplikacije uključujući i operacijski sustav. Pružatelj usluge s druge strane ima kontrolu nad hardverom i aplikacijom koja omogućuje podjelu resursa.



1.10 Pružatelj usluge u oblaku

Pružatelj usluge u oblaku zadužen je za četiri glavne karakteristike: skalabilnost, elastičnost, sigurnost i privatnost.

Pojam **skalabilnosti** označava sposobnost sustava da se prilagodi promjenama opterećenja tako da obilježje dostupnosti i pouzdanosti ostane očuvano. Osiguranje usluga ponekad nije moguće predvidjeti pa je jedan od najčešćih načina osiguravanja skalabilnosti dinamičkih sustava prekomjerno dodjeljivanje računalnih resursa. Primjenom ovog rješenja osigurava se dovoljno resursa za moguće iznenadne skokove u zahtjevima na sustav. Nedostatak ovog pristupa je to, što većinu vremena nije pod potpunim opterećenjem što znači da resursi nisu u potpunosti iskorišteni. Drugo rješenje je dimenzioniranje sustava prema prosječnom opterećenju korištenja usluge.

Elastičnost u **računarstvu** u oblaku podrazumijeva mogućnosti jednostavnog proširivanja i smanjivanja računalnih resursa. Ovisno o tome da li se poveća ili smanji broj korisničkih zahtjeva, na isti način se poveća ili smanji količina računalnih resursa. Za provedbu elastičnosti potrebno je izgraditi i konfigurirati proces koji će omogućiti skaliranje bez ljudske intervencije.

Elastičnost u oblaku se može postići na tri načina:

- Proaktivno cikličko skaliranje (periodično sa fiksnim intervalima)
- Proaktivno skaliranje temeljeno na događaju (u slučaju kada se očekuju velike promjene događaja)
- Auto-skaliranje na zahtjev (sustav prati uslugu i poduzima odgovarajuće mjere).

Sigurnost i **privatnost** podataka u oblaku su vrlo važne karakteristike. Sigurnosni zahtjevi koje je poslužitelj dužan osigurati su povjerljivost, integritet, dostupnost i nepriznavanje.

Povjerljivost-osiguravanje da se informacije ne otkriju osobama koje nemaju pravo pristupa

- Integritet-osiguravanje da se informacije ne modificiraju od strane neovlaštene osobe
- Dostupnost-osiguravanje da resursi za obradu informacija nisu nedostupni zbog utjecaja zlonamjernih korisnika
- Nepriznavanje-osiguravanje da postoje dokazi o postignutim dogovorima.

Korisnici oblaka

Korisnici računarskog oblaka su ujedno i pružatelji aplikacija kao usluga krajnjim korisnicima. Njihova uloga je izrada i održavanje aplikacije koja će se izvoditi u oblaku. Korisniku je cilj da od pružatelja usluge dobije što veću kvalitetu usluge uz što nižu cijenu. Za odredbu uvjeta i razinu korištenja usluge korisnici trebaju dokumente.

Dokument definira vezu između pružatelja usluge i korisnika i naziva se SLA ugovor. Postoje dvije vrste SLA ugovora: infrastrukturni i aplikacijski. Infrastrukturni SLA garantira dostupnost infrastrukture resursa i odnosi se na dostupnost napajanja, mreže i podatkovnih centara (definirano je i vrijeme dostupnosti određenog resursa). Aplikacijski SLA određuje perfomanse aplikacije i razinu usluge koja je pružena korisniku.

1.4 Što, kada i kako spremati u oblak?

Početak korištenja cloud computinga složena je odluka za svaku organizaciju budući da mora preispitati sve prednosti i nedostatke koje to donosi.

Cloud computing nije nužno više ili manje siguran od uobičajenog spremanja podataka na poslužitelje i druge korisničke sustave. Kao u bilo kojoj tehnologiji, tako i u ovoj, otvaraju se novi rizici, ali i nove mogućnosti. Cilj ovog vodiča nije da korisnicima kaže što, gdje i kako pomaknuti u oblak, nego da im pruži praktične preporuke i odgovore na ključna pitanja i da rad bude što sigurniji.

Security Guidance koji je objavio Cloud Security Aliance (više u kasnijem dijelu rada) trebao bi pomoći korisnicima u odabiru te im dati trenutne preporuke na što obratiti pažnju, a što izbjeći. Taj dokument se sastoji od tri osnovna dijela: arhitekture oblaka, upravljanja oblakom i operacije u oblaku. Kroz vodič su dane preporuke kako reducirati rizike prihvaćanjem cloud computinga. Naravno, nijedan vodič ne nudi potpune preporuke i upute jer cloud computing se sastoji od mnoštva različitih modela pa je nemoguće predvidjeti sve mogućnosti.

Okvir koji pomaže korisnicima u određivanju rizika i sigurnosti naveden je u slijedećim potpoglavljima.

Identifikacija sredstava koja korisnik želi pohraniti u oblaku

Prvi korak u procjeni rizika cloud computinga je određivanje što točno korisnici žele pohraniti, tj. da li žele pohraniti podatke, aplikacije ili funkcije. Korištenjem cloud computinga podaci i aplikacije ne moraju biti pohranjene na istim adresama.

Procjena važnosti sredstava

Idući korak je određivanje važnosti podataka ili funkcija za pojedinu organizaciju. Korisnici trebaju barem grubu procjenu osjetljivosti i važnosti podataka, aplikacija i funkcija.

Odabir modela

Korisnici moraju znati kolika je važnost sredstava. Prema tome biraju model cloud computinga koji im je pogodan. Naravno, svaki model donosi svoje rizike, stoga korisnik mora provjeriti može li podnijeti rizike vezane uz taj model.

Ocjenjivanje rizika

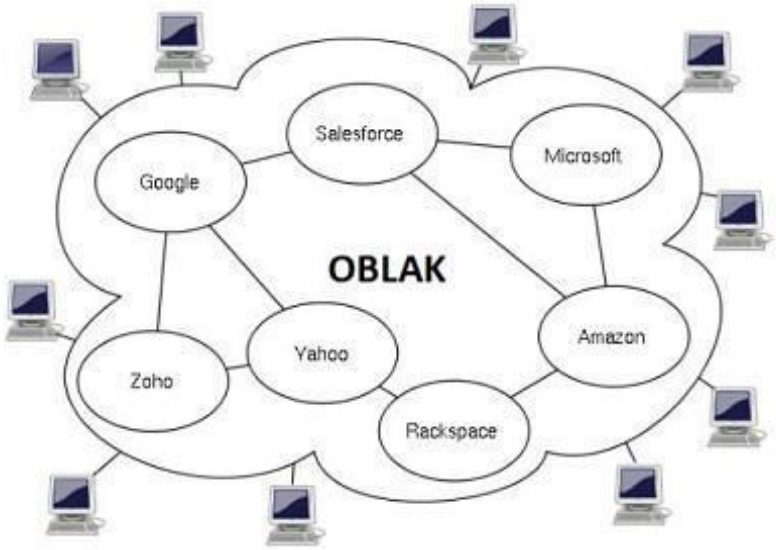
U ovom koraku korisnici se usmjeravaju na određivanje stupnja provjere. Ako već postoje određeni zahtjevi može ih uključiti u ocjenjivanje rizika.

Skiciranje potencijalnog protokola podataka

Ako korisnik razmatra specifičnu razvojnu mogućnost, trebao bi napraviti potpuni plan protoka podataka između vlastite organizacije i svakog drugog čvora.

1.5 Oblaci

Pojavom cloud computinga tržište je učinilo svoje. Svakim danom nastaje sve više davatelja usluga. Kao što to obično bude, svaki od njih ima svoje prednosti i mane. Dobra stvar je mnoštvo proizvođača na tržištu jer se stvara pozitivna konkurencija. Konkurencija ih prisiljava na neprekidnu potrebu usavršavanja i snižavanja cijena ako žele zadržati svoje korisnike. Svaki od korisnika ima svoje potrebe i različite su mu potrebe pa stoga se odlučuje za poslužitelje koji najbolje zadovoljavaju njihove potrebe.



1.11 Oblaci

Iznenađujuće je da su davatelji usluga poprilično različiti. Amazonov Elastic Cloud nudi Linux strojeve s administratorskim pristupom i s mogućnosti pokretanja bilo koje aplikacije. Googleov App Engine također dopušta i pokretanje bilo kojeg programa napisanog u Pythonu koji koristi Googleovu bazu podataka.

Poslužitelji nude različite mogućnosti spremanja podataka. Svaka osobina cloud computinga koja pojednostavljuje rad čini to uklanjanjem pojednih sklopki iz dosega korisnika te mu preostaje korištenje skupa rutina.

Oblaci su uglavnom zajednički poslužitelji. Cloud computing omogućava korištenje više procesa kad god se za to pojavi zahtjev. Ipak nisu u mogućnosti riješiti probleme raspoređivanja i spremanja podataka.

Amazon Elastic Compute cloud

Amazon je bio jedna od prvih organizacija koja je lansirala rješenje namijenjeno široj javnosti, koji i dalje ima jedan od najprofinjenijih i razrađenijih kompleta mogućnosti. Ako korisnik treba više procesora, korisnik može ubrzati virtualne strojeve s EC2 web uslugom. Amazon je izgradio i novi sustav korištenja baze podataka nazvan S3.

Korištenjem sigurnosnih alata i upisivanjem naredbi u linije za upisivanje podataka, korisnik šalje naredbe vlastitom setu polužitelja u oblaku. Korisnik izabire vlastiti virtualni poslužitelj na koji želi instalirati vlastite programe i korisničke instalacije. Svi virtualni poslužitelji koriste neki od Linux operacijskih sustava.

Google App Engine

Google App Engine nudi potpuno druge stvari u odnosu na Amazon. Kod Amazona korisnik dobiva administratorske ovlasti, na App Enginu ne može zapisati datoteku ni u svom vlastitom direktoriju. Google je iz sigurnosnih razloga zabranio korisniku zapisivanje podataka izvan Googleove baze podataka.

Rezultat svih ti ograničenja nije nužno loša stvar. Web aplikacije imaju samo osnovni skup osobina i prilično je dobro okruženje za njihovu dostavu. Google nudi dobre alate za traženje pogrešaka u aplikacijama na korisnikovu stroju, ali inzistira na registraciji i povezivanju. Najčešće to bude pomoću Gmail korisničkog računa ili broja mobitela.

AppNexus

AppNexus poslužitelj je dobar izbor za korisnike kojima se sviđa ideja cloud computinga, ali nisu sigurni žele li napustiti korištene jezgre Unix operacijskog sustava te nekih administratorskih alata. Organizacija u čijem je vlasništvu AppNexus je uzela veliki poslužitelj industrijske veličine s najboljim alatima za zajedničko učitavanje i pohranu podataka te pronašla način prodavanja malih dijelova poslužitelja korisnicima. AppNexus pruža nekoliko apstraktnih naredbenih linija kojima korisnici dobivaju mogućnost uključivanja i isključivanja poslužitelja te pristup datotečnom sustavu.

Osnove fukcionalnosti AppNexus oblaka su slične Amazonovom EC2. Korisnik se prijavljuje preko naredbene linije i pokrene pripremljenu distribuciju operacijskog sustava Linux. Ako se pojavi potreba, AppNexus može inicijalizirati virtualni poslužitelj i s drugih izvora. Pri takvoj izgradnji dolazi do zamjenjivanja jezgre s inačicom koja je pogodnija za virtualno okruženje. Korisnici na jednostavan način mogu pratiti pohranu i dijeljenje podataka unutar mreže poslužitelja.

Bolje rješenje je korištenje mreže poslužitelja za pohranu podataka koji imaju ugrađen vlastiti skup HTTP poslužitelja. Takav se sustav se naziva CDN sustav. U njemu korisnik može odmah započeti sa spremanjem statičkih podataka iz vlastitih datoteka. Podaci postaju dostupni odmah nakon njihovog zapisivanja. AppNexus raspodjeljuje podatke u oblaku za pohranu podataka kako bi stvorio više neovisnih podatkovnih centara.

GoGrid

GoGrid se sam smatra prvim svjetskim upravljačkim sustavom s više poslužitelja. Ono što GoGrid predstavlja funkcionalno je slično kao i Amazonov EC2, ali korištenje termina kontrolni okvir čini se ipak boljim opisom onoga što se događa nego što se to vidi korištenjem termina oblak. Korisnik započinje i prekida učitavanje podataka na isti način kao relativno stari alati poput cPanela i Pleska na način da dodaju i oduzimaju usluge.

GoGrid nudi mnoge iste usluge kao Amazon EC2. Njegov web control panel korisnicima je ipak mnogo lakši za uporabu od EC2-ove naredbene linije. Ono što olakšava korisnicima rad su pop-up prozori pa nema potrebe za izrezivanjem i kopiranjem. Korisno je i to da za vrijeme korištenja poslužitelja, korisnici imaju konstantan uvid u troškove.

GoGrid nudi mnoštvo pripremljenih operacijskih sustava spremnih za uporabu. Postoji uobičajena zbirka CentOS/Feddora i LAMP rješenja. Postoje i pripremljeni operacijski sustavi s instaliranim aplikacijskim poslužiteljima tipa Facebook ili PostrgeSQL.

Nedostatak usluga za zajedničko spremanje informacija otežava pokretanje i gašenje poslužitelja. Još jedan od problema korisniku stvara nemogućnost prestanka plaćanja usluge. Ako korisnik želi prestati plaćati, prisiljen je izbrisati poslužitelj i izgubiti sve podatke koji se u tom trenutku na njemu nalaze.

Poglavlje 2

Mrjob i PiCloud

U ovom poglavlju cilj nam je pokazati konkretne stvari u vezi cloud computinga. Bit će pojašnjeno što je to mrjob i picloud te čemu služe. Bit će prikazano i na par konkretnih primjera.

2.1 MapReduce

Funkcija Map obavlja filtriranje i sortiranje. Glavni čvor prihvaća ulazne podatke, dijeli ih na manje pod-probleme i distribuira na ostale čvorove u clusteru. Čvor može ponoviti isti postupak. Posljednji čvor u stablu procesira manji problem i šalje ga natrag glavnom čvoru.

Funkcija Reduce obavlja operaciju sumiranja. Glavni čvor skuplja sva rješenja pod-problema i kombinira ih. Na taj način se dobija izlazni podatak - rješenje početnog problema.

Funkcije Map i Reduce su definirane na strukturi (ključ, vrijednost). Funkcija Map uzima jedan par (ključ, vrijednost) iz podatkovne domene i vraća listu parova (ključ, vrijednost) u drugoj domeni. Funkcija Map se primjenjuje paralelno za svaki par ulaznih podataka. Na ovaj način nastaje lista parova za svaki poziv. Svi parovi s istim ključem iz svih lista se prikupljaju te tvore po jednu grupu za svaki ključ. Funkcija Reduce se primjenjuje paralelno na svaku grupu. Rezultat je lista vrijednosti u istoj domeni. Jedan poziv Reduce funkcije može vratiti nijednu, jednu ili više vrijednosti.

map (k1, v1) -> list (k2, v2)
reduce (k2, list (v2)) -> list (v2)

MapReduce se može promatrati kroz 5 koraka koji se izvršavaju paralelno i distribuirano:

1. **Priprema ulaznih podataka za Map funkciju** - *MapReduce određuje procesore za izvršavanje Map funkcije i dodjeljuje ulazne parove (ključ, vrijednost) za svaki procesor*

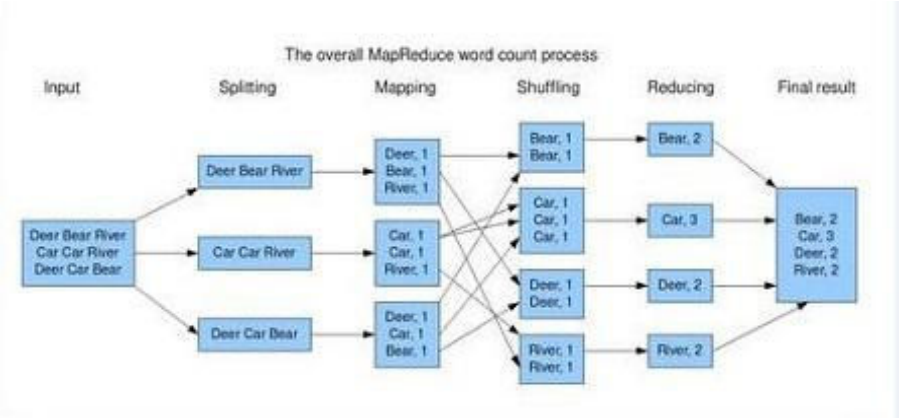
2. **Izvršavanje Map funkcije** - *točno jedanput za svaki ključ generirajući izlazne podatke (ključ2, vrijednost2)*

3. **"Shuffle" (miješanje) izlaznih podataka iz Map funkcije** - *MapReduce određuje procesore za izvršavanje Reduce funkcije te im dodjeljuje (ključ2, vrijednost2) parove*

4. **Izvršavanje Reduce funkcije** - *točno jednom za svaki ključ2*

5. **Vraćanje konačnog izlaznog rezultata** - *MapReduce skuplja sve izlazne podatke iz Reduce funkcije i sortira ih po ključu2 da bi dobio konačni rezultat.*

Najveća prednost ovog modela je što programer mora implementirati samo Map i Reduce funkcije.



2.1 MapReduce

Primjer:
Izračunati broj pojavljivanja svake riječi u velikom skupu dokumenata.

```
Map(String key, String value):
   // key:  document name
   // value:  document contents
  for each word w in value:
     EmitIntermediate(w, "1");
Reduce(String key, Iterator values):
 // key:  a word
  // values:  a list of counts
   int result = 0;
  for each v in values:
   result += ParseInt(v);
   Emit(AsString(result));
```

U ovom primjeru funkcija Map pridružuje svakoj riječi broj pojavljivanja. Funkcija Reduce sumira sve brojeve pojavljivanja za svaku pojedinu riječ.

MapReduce model se jednostavno koristi. Detalji paralelizacije, otpornosti na pogreške te lokalne optimizacije su sakriveni od programera. Različiti problemi se mogu predstaviti preko ovog modela. Na primjer, Google koristi ovaj model za web search usluge, sortiranje, strojno učenje itd. MapReduce ima mogućnost implementacije na više tisuća računala te je zbog toga veoma primjenjiv model.

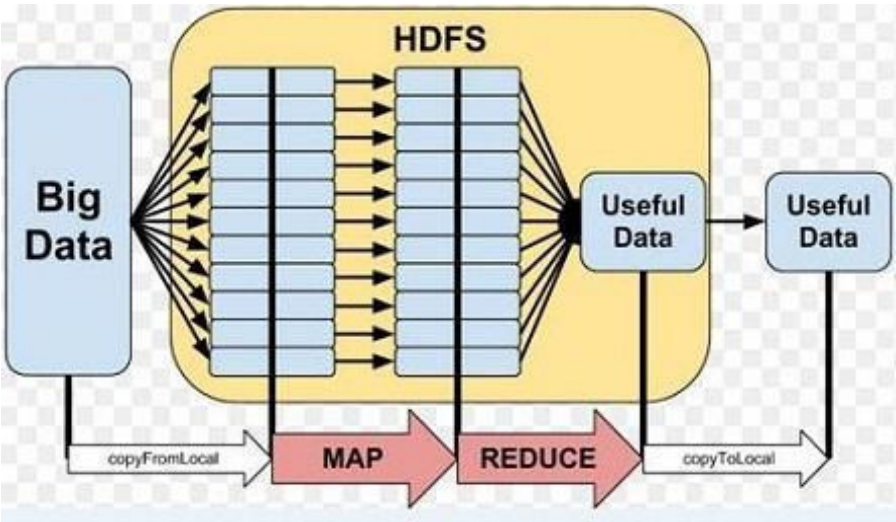
2.2 Hadoop

Hadoop je besplatan, na Javi temeljen programski okvir koji podržava obradu velikih skupova podataka u računalnom okruženju. To je dio projekta Apache pod pokroviteljstvom Apache Software Foundation.

Hadoop omogućava pokretanje aplikacija na sustavima s tisućama čvorova koji uključuju tisuće terabajta. Njegov distribuirani datotečni sustav olakšava brzi prijenos podataka između čvorova i omogućava sustavu da nastavi s radom bez prekida i u slučaju kvara čvora. Ovakav pristup smanjuje rizik od katastrofalnog kvara sustava, čak i ako je značajan broj čvorova postao neupotrebljiv.

Hadoop je inspiriran Googleovim MapReduce-om, softverskim okvirom u kojem se aplikacije dijele u brojne manje dijelove. Svaki od tih dijelova (koji se nazivaju i fragmenti ili blokovi) mogu se izvoditi na bilo kojem čvoru u klasteru. Doug Cutting, kreator Hadoopa, naziva framework imenom djetetove igračke (plišanog slona). Trenutni Apache Hadoop ekosustav sastoji se od Hadoop kernel-a, MapReduce-a, Hadoop distribuiranog datotečnog sustava (HDFS) i niza drugih projekata, kao što su Apache Hive, HBase i Zookeeper.

Hadoop okvir se koristi od strane velikih igrača, uključujući Google, Yahooa i IBM, uglavnom za primjene koje uključuju tražilice i oglašavanje. Preferira operacijske sustave Windows i Linux, ali Hadoop također može raditi s BSD i OS X.



2.2 Hadoop

2.3 Mrjob

Mrjob je najlakši put do pisanja Python programa koji radi na Hadoop-u. Prednost korištenja mrjob-a je da ga možete pokrenuti lokalno bez instalacije Hadoopa ili ga možete pokrenuti u klasteru.

Velika je prednost što je mrjob povezan sa Amazon Elastic Mapreduceom. Jednom kada se postavke namjeste, pokretanje u oblaku je jednostavno kao i pokretanje na vlastitom računalu. Programi koji se rade u mrjob-u nazivaju se procesi (engl. job).

Nekoliko savjeta koji čine pisanje MapReduce poslova lakšima:

- stavljati sve MapReduce kodove u jednu klasu
- *uploadati* i instalirati sve kodove i podatke u ovisnosti o *runtime*-u
- prebaciti ulazne i izlazne podatke u jednu liniju koda
- automatsko preuzimanje i analiziranje pogrešaka za Python *tracebacks*
- staviti naredbu filters poslije Python koda.

Velika prednost mrjob-a je što se ne mora biti stručnjak u Hadoopu da bi se koristila moć MapReduce-a. Mrjob ima veću količinu dokumentacije od bilo kojeg drugog *frameworka* ili *library*-ja kojeg smo upoznali. On dopušta da se kod pokrene lokalno bez Hadoopa, dok drugi *frameworkovi* zahtjevaju Hadoop instancu. Pruža i konzistentno sučelje za svako okruženje koje ga podržava.

Za mrjob je nevažno da li *job* pokrećete lokalno, u oblaku ili vlastitom klusteru, Vaš Python kod se uopće ne mijenja. Mrjob koristi velik broj strojeva za dobivanje koda i podataka od i do klastera koji pokreće Vaš proces. Ne treba za to imati nikakve skripte.

Mrjob kompajliranje čini jednostavnijim. Lokalno može pokrenuti jednostavnu MapReduce implementaciju, tako ćete *traceback* dobiti u svojoj konzoli umjesto u nejasnoj *log* datoteci. Na klasteru ili na Elastic MapReduce analizira se pogreška evidencije za Python *tracebacks* kao i druge moguće pogreške.

Za pokretanje *job*-a možemo birati par opcija. Možemo pokrenuti lokalno, naredba *-r local*, na hadoopu *-r hadoop* ili na oblaku *-r emr*, a ako ne izaberemo nijednu onda se pokreće *defaultno* (inline).

Ovaj rad se bazira na računarstvu u oblaku stoga ćemo kod pokretati sa *-r emr*. Prva stvar koju ćemo napraviti je otvoriti račun na Amazonu. Ako otvaramo prvi puta račun, kao što je moj slučaj, imamo godinu dana besplatnog korištenja. Pri registraciji treba paziti da se ne koriste slova sa kvačicom jer ih poslužitelj ne podržava. Registracija je dostupna na linku *http://aws.amazon.com/*. Nakon par koraka popunjavanja naših podataka, račun je kreiran.

Osim računa na Amazonu, potreban je i *job* koji ćemo pokretati. U ovom primjeru pokrenut ćemo *job* koji pronalazi riječ u tekstu koja se najčešće spominje.

```
from mrjob.job import MRJob
import re
WORD_RE = re.compile(r"[\w']+")
class MRMostUsedWord(MRJob):
    def mapper_get_words(self, _, line):
        for word in WORD_RE.findall(line):
        yield (word.lower(), 1)
    def combiner_count_words(self, word, counts):
        yield (word, sum(counts))
    def reducer_count_words(self, word, counts):
        yield None, (sum(counts), word)
    def reducer_find_max_word(self, _, word_count_pairs):
        yield max(word_count_pairs)
    def steps(self):
        return [
            self.mr(mapper=self.mapper_get_words,
            combiner=self.combiner_count_words,
            reducer=self.reducer_count_words),
            self.mr(reducer=self.reducer_find_max_word)
              ]
if __name__ == '__main__':
MRMostUsedWord.run()
```

Nakon instalacije Pythona ako nemamo potrebne module moramo ih instalirati. Nakon toga prvo definiramo kako naša riječ izgleda. Napravimo klasu koja se sastoji od četiri koraka:

- prvi korak je grupiranje (engl. mapping), cilj je izdvojiti sve zasebne riječi i dodijeljujemo im vrijednost jedan.
- u drugom koraku zbrojimo sve iste riječi
- u trećem koraku zamijenimo ključ to jest prvo nam bude broj pojavljivanja riječi, onda riječ (zbog sortiranja)
- četvrti korak je odabiranje riječi koja se najviše puta pojavila.

Lokalno ovaj kod ne bi predstavljao nešto značajno, dosta brzo bi se izvrtio, ali ono što ćemo sada vidjeti kako bi to izgledalo u oblaku. Pošto smo kreirali račun sada se spojimo na njega. Klikom na *Security Credentials*, prvo stvaramo vezu i dobijemo *AWS_ACCESS_KEY_ID i AWS_SECRET_ACCESS_KEY*. Kada smo dobili ta dva ključa potrebno je na *home*-u stvoriti novu datoteku *mrjob.conf* i u nju upisati:

runners:
    aws_access_key_id: <tvoj key ID>
    aws_secret_access_key: <tvoj secret ID>

Preostalo je samo izabrati tekst koji ćemo pretražiti. Sve je spremno. Možete pokrenuti svoj *job* u oblaku. Slijedi par *screen shot*-ova:

```
using configs in /home/ena/.mrjob.conf
using existing scratch bucket mrjob-a4fec71f6db6ac21
using s3://mrjob-a4fec71f6db6ac21/tmp/ as our scratch dir on S3
creating tmp directory /tmp/krsnil.ena.20140908.205345.447892
writing master bootstrap script to /tmp/krsnil.ena.20140908.205345.447892/b.py
Copying non-input files into s3://mrjob-a4fec71f6db6ac21/tmp/krsnil.ena.20140908.205345.447
892/files/
Waiting 5.0s for S3 eventual consistency
Creating Elastic MapReduce job flow
Job flow created with ID: j-2NLRN6KRDNIVM
Created new job flow j-2NLRN6KRDNIVM
Job launched 30.6s ago, status STARTING: Provisioning Amazon EC2 capacity
Job launched 61.5s ago, status STARTING: Provisioning Amazon EC2 capacity
Job launched 92.1s ago, status STARTING: Provisioning Amazon EC2 capacity
Job launched 122.9s ago, status STARTING: Provisioning Amazon EC2 capacity
Job launched 153.4s ago, status BOOTSTRAPPING: Running bootstrap actions
Job launched 184.2s ago, status BOOTSTRAPPING: Running bootstrap actions
Job launched 214.8s ago, status BOOTSTRAPPING: Running bootstrap actions
Job launched 245.7s ago, status BOOTSTRAPPING: Running bootstrap actions
Job launched 276.4s ago, status RUNNING: Running step (krsnil.ena.20140908.205345.447892: S
tep 1 of 2)
Job launched 307.2s ago, status RUNNING: Running step (krsnil.ena.20140908.205345.447892: S
tep 1 of 2)
Job launched 337.8s ago, status RUNNING: Running step (krsnil.ena.20140908.205345.447892: S
tep 1 of 2)
Job launched 368.5s ago, status RUNNING: Running step (krsnil.ena.20140908.205345.447892: S
tep 1 of 2)
Job launched 399.2s ago, status RUNNING: Running step (krsnil.ena.20140908.205345.447892: S
tep 2 of 2)
Job launched 430.0s ago, status RUNNING: Running step (krsnil.ena.20140908.205345.447892: S
tep 2 of 2)
Job launched 460.5s ago, status RUNNING: Running step (krsnil.ena.20140908.205345.447892: S
tep 2 of 2)
Job completed.
Running time was 260.0s (not counting time spent waiting for the EC2 instances)
ec2_key_pair_file not specified, going to S3
Fetching counters from S3...
Waiting 5.0s for S3 eventual consistency
```

2.3 obrada procesa

Primjećujemo da je vrijeme trajanja procesa bilo 460 sek, od toga je EC2 trajao 260 sek, ostatak je potrošeno na spajanje na oblak.

```
  FileSystemCounters:
    FILE_BYTES_READ: 1299
    FILE_BYTES_WRITTEN: 137902
    HDFS_BYTES_READ: 6553
    S3_BYTES_WRITTEN: 7
  Job Counters :
    Data-local map tasks: 4
    Launched map tasks: 4
    Launched reduce tasks: 1
    SLOTS_MILLIS_MAPS: 73420
    SLOTS_MILLIS_REDUCES: 36337
    Total time spent by all maps waiting after reserving slots (ms): 0
    Total time spent by all reduces waiting after reserving slots (ms): 0
  Map-Reduce Framework:
    CPU time spent (ms): 3350
    Combine input records: 0
    Combine output records: 0
    Map input bytes: 2383
    Map input records: 129
    Map output bytes: 2383
    Map output materialized bytes: 1555
    Map output records: 129
    Physical memory (bytes) snapshot: 886149120
    Reduce input groups: 1
    Reduce input records: 129
    Reduce output records: 1
    Reduce shuffle bytes: 1555
    SPLIT_RAW_BYTES: 588
    Spilled Records: 258
    Total committed heap usage (bytes): 606666752
    Virtual memory (bytes) snapshot: 3314245632
Streaming final output from s3://mrjob-a4fec71f6db6ac21/tmp/krsnil.ena.20140908.205345.4478
92/output/
11      "u"
removing tmp directory /tmp/krsnil.ena.20140908.205345.447892
```

2.4 prikaz potrebnnih resursa

Najčešća riječ u tekstu je bilo slovo u i pojavilo se jedanaest puta.
Ulazni, izlazni podaci i podaci o procesu su spremljeni i na Amazonov S3.

Name	Storage Class	Size
b.py	Standard	510 bytes
krsni1.py	Standard	1.3 KB
mrjob.tar.gz	Standard	159.5 KB
my_file.txt	Standard	1 KB

2.5 prikaz obrađenih podataka pohranjenih u S3

2.4 PiCloud

PiCloud Vam daje superračunalo na dohvat ruke. U samo nekoliko redaka koda omogućava Vam pristup tisućama procesora i tisuće terabajta za pohranu podataka bez potrebe za upravljanjem, održavanjem ili konfiguriranjem poslužitelja. PiCloud platforma je idealna za računarstvo kojemu su potrebne visoke perfomanse, skupne obrade i znanstvene računalne aplikacije.

Sučelje PiCloud Plaforme mogu koristiti njegovi klijenti. Za riješavanje Python funkcija posebno je dizajniran *Cloud library*, premda u njemu možemo izvršavati i bilo koje funkcije napisane u drugim programima (Java, R, C, C++, Matlab,...).

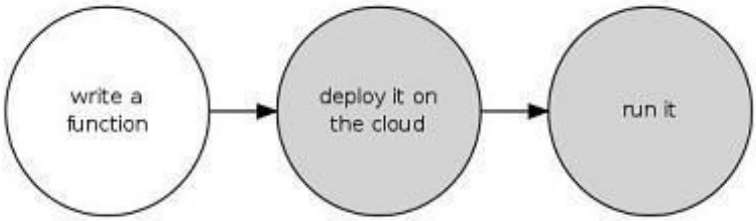
Kreira se prvo job. PiCloud doda *job* u red (QUEUE), kada je jezgra slobodna automatski ga pokreće. Ako izvršavanje procesa ne može čekati onda postoji mogućnost zakupljivanja vlastitog reda, tada samo mi možemo koristiti tu jezgru. Sljedeći primjer pokazat će kako se kreira *job*.

```
>>> # definiramo funkciju
    >>> def add(x, y):
    ...  return x + y

>>> # uključimo oblak
    >>> import cloud
>>> # kreiramo job na na oblaku, i vrati nam identifikacijski broj
procesa
    >>> job_id = cloud.call(add, 1, 2)
    >>> print job_id
    42
>>> # provjerimo status ('queued', 'processing', or 'done')
    >>> cloud.status(job_id)
    'done'

>>> # ispišemo rezlutate
    >>> cloud.result(job_id)
```

```
    3
>>> # možemo pokrenuti više puta i na više podataka
    >>> cloud.map(add, datapoints)
```



2.6 prikaz postupka

PiCluod će staviti *job* na oblak već u dvije linije koda. Ali ipak nije tako jednostostavno, jer klijenti moraju imati kontrolu nad tim pa stoga treba postaviti razne postavke. Kada se kod pošalje u cloud prepušta se poslužitelju na povjerenje i vraća se izračunato. Sigurnost je defaultno postavljena. Sva komunikacija s PiCloud-om je šifrirana pomoću Secure Sockets Layer (SSL) protokola. Vaš broj i podaci o skupini su zaštićeni u više slojeva sigurnosti koje posjeduju posix dozvole. PiCloud funkcionira kao i ostali oblaci, plaća se točan iznos koliko iznosi vrijeme računanja i točna količina pohrane podataka, a održavanje, iskoristivost i resursi su briga poslužitelja.

Poglavlje 3

Financijska strana i sigurnost

3.1 Ekonomski aspekt

Korištenjem cloud computinga moguće je izbjeći velike troškove kupnje skupih sklopovlja, programa i usluga. Korisnici cloud computing usluga plaćaju samo ono što koriste. Uglavnom ne postoje zahtjevi za plaćanje unaprijed, a troškovi su jako mali u odnosu na korištenje vlastite IT strukture. Ovaj pristup organizaciji IT rješenja korisnicima nudi jednostavan pristup podacima i mnoštvu različitih aplikacija. Druge prednosti ovog pristupa su podijeljena infrastruktura i niski troškovi nadzora.

Općenito, korisnici uvijek mogu uštedjeti na kapitalnim troškovima, ali s druge strane pri korištenju cloud computinga organizacije moraju biti jako oprezne. Ovisno o potrebama organizacije, troškovi usluga mogu biti jako skupi pa korištenje cloud computinga ne mora dovesti do velikih financijskih ušteda. U situacijama kada bi glavni troškovi bili relativno mali ili kada organizacija ima veću fleksibilnost u svom osnovnom proračunu nego u operacijskom proračunu, cloud model i nema nekog velikog financijskog smisla. Drugi faktori koji utječu na bilo koje druge potencijalne uštede uključuju učinkovitost organizacije baze podataka pojedine organizacije u usporedbi s oblakom nekog dobavljača, postojeće troškove organizacije, razinu prihvaćanja cloud computinga i tip funkcionalnosti koju oblak posjeduje.

Uvjeti i cijene

Korisnikov prvi korak ka pristupu cloud computingu je registracija. Nakon registracije korisnik i davatelj usluge sklapaju ugovor. Ugovori su specifični i jasni. U njima su navedeni uvjeti koje korisnik mora poštovati i mogućnosti koje potpisivanjem dobiva davatelj usluge.

GoGrid u ugovoru navodi brojčane pragove za standardne vrijednosti poput vremena čekanja, mijenjanja faze signala, gubitka paketa i dr. Ako sustav ne zadovoljava dogovorene stavke onda GoGrid obećava uslugu sto puta veću od izgubljenog iznosa.

Amazon u ugovoru o korištenju usluge cloud computinga ima pravo ukinuti korisnički račun i obrisati spremljene podatke. Amazon pri tome nije dužan dati objašnjenje za brisanje korisničkih podataka.

Potpisivanjem ugovora između Googla i korisnika, Google dobiva pravo skeniranja, pregledavanja, obilježavanja, filtriranja, mijenjanja, odbijanja i uklanjanja bilo kojeg sadržaja sa poslužitelja. Ipak Google nudi bolje uvjete za brisanje korisničkih podataka. Google će obrisati račun samo ako korisnik ne poštuje uvjete ugovora ili je sklon ilegalnim radnjama.

Poslužitelji cloud computinga su prilagodljivi pri izgradnji mreže uređaja, ali još je daleko od savšenstva. Veliki problem je mogućnost rušenja poslužitelja ili mogućnost gubitka podataka za vrijeme procesa. Ako korisnik ne želi da sigurnost utječe na njegove podatke, mora ih pohraniti u podatkovnu memoriju oblaka, a to je jako skupo.

Svaka od organizacija na različite načine vrši naplatu usluge. GoGrid ima Intel Xeon poslužitelje, smatraju ih najboljim na tržištu, stoga GoGrid povećava cijenu svojih usluga. Google svoju uslugu naplaćuje po broju ciklusa CPU-a, dok Amazon EC2 naplaćuje po veličini strojeva.

Primjeri cijena:

Resurs	Jedinica	Cijena u $
Premještanje podataka na poslužitelj	GB	0.12
Premještanje podataka s poslužitelja	GB	0.10
Vrijeme CPU	sati rada CPU	0.10
Pohrana podataka	GB mjesečno	0.15

Googleove cijene

Resurs	Jedinica	Cijena u $
Premještanje podataka na poslužitelj	GB	0.10
Premještanje podataka s poslužitelja	GB	0.15
Vrijeme CPU	sati rada CPU	0.12
Pohrana podataka	GB mjesečno	0.15
Prijenosi pohranjenih podataka	10K aplikacijskih zahtjeva	0.01

Amazonove cijene

Resurs	Jedinica	Cijena u $
Premještanje podataka na poslužitelj	GB	0.10
Premještanje podataka s poslužitelj	GB	0.14
Vrijeme CPU	sati rada CPU	0.125
Pohrana podataka	GB mjesečno	0.15

Microsoftove cijene

Napomena:
Otvaranje novog računa na Amazon-u ima probnu verziju koja sadrži:

- *12 mjeseci besplatnog pristupa*
- *Amazon EC2: 750 hrs/month of Windows and Linux Micro Instance usage*
- *Amazon S3: 5 GBs of Storage*
- *Amazon RDS: 750 hrs/month of Micro DB Instance usage*

Premda bi accaunt po reklami trebao biti skroz besplatan ipak sam dobio neki račun stoga treba biti oprezan sa korištenjem Cloud computinga kao što je u radu i navedeno.

3.1 cijena pokrenutog procesa u Amazonu

3.2 Sigurnost

U dosadašnjem dijelu rada, a i u mnogim člancima vezanim za cloud computing, često se spominje sigurnost. Glavnu ulogu u poboljšanju sigurnosti ima neprofitabilna organizacija Cloud Security Alliance sačinjena od mnoštva stručnjaka iz cijelog svijeta iz različitih disciplina računarastva.

Cloud Security Alliance se bavi:

- *educiranjem korisnika o načinu uporabe cloud computinga*
- *promicanjem međusobnog razumijevanja između korisnika i pružatelja usluga*

- *promicanjem nezavisnih istraživanja u područjima sigurnosti*
- *podizanjem svijesti i organiziranjem edukacijskih tečajeva*
- *stvaranjem liste problema i smjernica za povećanjem sigurnosti.*

Cloud Security Alliance koncept nastao je 20. kolovoza 2008. godine u Las Vegasu na konferecniji ISSA CISO Forum. Na toj konferenciji Jim Reavis održao je predavanje o nadolazećim trendovima zbog kojih se moraju početi poduzimati značajne akcije vezane uz sigurnost cloud computinga. Reavis i Nils Puhlmann su nakon toga odredili početne misije i strategije CSA. U prosincu iste godine nakon mnogo sastanaka službeno je osnovana organizacija Cloud Security Alliance. 2009. objavljen je i prvi vodič za najbolju upotrebu cloud computing tehnologije. Cloud Security Alliance izdao je i novu inačicu dokumenta "Security Guidance for Critical Areas of Focus in Cloud Computing" koji služi kao vodič i pri upravljanju rizicima pri korištenju cloud computinga.

Zahvaljujući Cloud Security Allianceu danas se neprekidno pojavljuju nova rješenja za uklanjanje sigurnosnih problema, a i ideje za rješavanje budućih mogućih problema.

Premda CSA obavlja ogroman posao, problemi još uvijek postoje. Navest ćemo sedam osnovnih prijetnji sigurnosti:

Zlouporaba cloud computinga

Zlonamjerni korisnici relativno jednostavno i nekažnjeno iskorištavaju sigurnosne propuste tako da razvijaju tehnologije koje će im omogućiti pristup i korištenje podataka drugih korisnika, a pri tome im nije ugrožen njihov vlastiti identitet.

Da bi se korisnici i davatelji usluga zaštitili od napada trebali bi:

- *uvesti složeniju početnu registraciju i provjeru procesa*
- *poboljšati praćenje i koordinaciju priljeva koje se izvode preko kreditnih kartica*
- *uvesti cjelokupno provjeravanje mrežnog prometa korisnika*
- *nadzirati zlonamjerne korisnike (pomoću njihovih adresa).*

Nesigurna sučelja i API

Davatelji cloud computing usluga otkrivaju dio programskih sučelja ili API-a koje korisnici koriste za upravljanje cloud uslugama(npr. username i password). Zlonamjerni korisnici za izvršavanje napada na korisnike često iskorištavaju to što se

korisnici susreću s anonimnom potrebom za ponovnom lozinkom(zaborave staru) ili nekim sličnim problemom.

Da bi se korisnici i davatelji usluga zaštitili od napada trebali bi:

- *analizirati sigurnosne modele sučelja davatelja cloud computing usluga*
- *prenositi šifrirani signal*
- *razumjeti da se preko sučelja povezuju na druge sustave koji su možda sigurnosno ugroženi.*

Zlonamjerni korisnici koji napade izvode iznutra

Davatelji usluga ne otkrivaju način na koji svojim korisnicima daju pristup fizičkim i virtualnim sredstvima, niti način na koji prate korisnike, analiziraju i izvještavaju o suradnji. Zlonamjerni korisnici ove informacije rado iskorištavaju te dolaze do povjerljivih podataka i dobivanja potpunog nadzora nad uslugom s jako malom vjerojatnošću da će biti otkriveni. Ti korisnici mogu narušiti ugled organizacije, smanjiti produktivnost ili na razne druge načine je oštetiti.

Da bi se korisnici i davatelji usluga zaštitili od napada potrebno je:

- *provoditi strogi nadzor nad lancem nabave i provoditi cjelokupne procjene isporučitelja*
- *odrediti zahtjeve za ljudskim resursima kao dio pravnog ugovora*
- *zahtjevati transparentnost u informacijskoj sigurnosti*
- *odrediti proces obavještavanja o sigurnosnim problemima.*

Zajednički tehnološki problemi

Dijelovi diska, predmemorija procesa, GPU i drugi zajednički elementi nikada nisu bili namijenjeni snažnoj podjeli. Kao rezultat toga zlonamjerni korisnici se usmjeravaju na izvođenje operacija na prostorima drugim korisnicima i tako dobivaju neovlašteni pristup podacima.

Da bi se korisnici i davatelji usluga zaštitili od napada potrebno je:

- *implementirati najbolje sigurnosne postavke za instalaciju i konfiguraciju*

- *proučavati okruženje*
- *nadzirati pristup administrativne radnje*
- *promicati ugovore o razini usluge kako bi se smanjio utjecaj ranjivosti.*

Gubitak i neovlašteno gubljenje podataka

Postoje brojni načini na koje zlonamjerni korisnici mogu ugroziti važne podatke organizacije. Primjer toga je brisanje ili promjena podataka bez kopije orginalnog sadržaja. Jedan od načina je raskidanje veze između dijela podataka pohranjenog na nekom drugom dijelu poslužitelja. Bilo kakav gubitak ili neovlašteno otkrivanje može ugroziti ugled i reputaciju organizacije te dovesti do prekida poslovne suradnje.

Da bi se korisnici i davatelji usluga zaštitili od napada potrebno je:

- *implementirati sučelja s dobrom kontrolom pristupa*
- *kriptirati podatke i zaštititi integritet podataka*
- *analizirati zaštitu podataka za vrijeme dizanja i izvođenja*
- *nakon prestanka suradnje poslužitelj bi trebao izbrisati podatke*

Krađa korisničkih imena

Krađom korisničkog imena napadači mogu iskorištavati usluge koje plaćaju korisnici. Mogu se koristiti korisnikovim identitetom, vidjeti korisnikove aktivnosti i transakcije. Ovi problemi su dosta česti zbog čestih korisničkih grešaka ili korištenja čestih i jednostavnih šifri.

Da bi se korisnici i davatelji usluga zaštitili od napada potrebno je usmjeriti se na:

- *korištenje snažne dvofaktorske autentikacijske tehnike*
- *izvođenje proaktivnih praćenja za otkrivanje neovlaštenih aktivnosti*
- *razumijevanje sigurnosne politike i SLA.*

Nepoznati profil rizika

Jedna od karakteristika cloud computinga je smanjivanje vlasništva nad sklopovljem i programima. To dovodi do financijskih i operacijskih prednosti koje su kontradiktorne sa sigurnosnim problemom. Sigurnost se može jednostavno postići skrivanjem podataka, ali može rezultirati nepoznatim opasnostima.

Da bi se korisnici i davatelji usluga zaštitili od napada potrebno je:

- *razotkrivanje primjenjivih zapisa i podataka*
- *praćenje i upozoravanje na potrebne informacije kojima se korisnici mogu smanjiti izloženom riziku.*

3.3 Povijest i budućnost

Da bi shvatili kako se razvio sam pojam Cloud computing-a moramo se vratiti u vrijeme kada je u svijetu dominirao telefon. Naime engleska riječ cloud u tehnološkom smislu označava nešto što se odvija putem nekakve mreže. Tako se za telefonsku mrežu znalo metaforički reći da je u "oblacima". Nakon pojave Interneta ista se metafora počela koristiti i za Internet. Cloud computing je tako samo produžena varijanta već ukorijenjenog izraza.

Prvo spominjanje koncepta Cloud computing-a datira u 1960-te godine i povezuje se sa dvojicom pionira u području računalstva i njihovim idejama o budućnosti tog polja. Prvi od njih J.C.R. Licklider, inače ključna figura u razvoju računalne znanosti, zamislio je računalstvo u obliku globalne mreže, tj. da svi na Zemlji budu povezani te imaju pristup programima i podacima na svakoj web stranici s bilo koje lokacije na svijetu. To je ideja koja uvelike sliči današnjem Cloud computing-u. Drugi od dvojice, John McCharty napisao je da će "Računalstvo jednog dana biti organizirano kao javno komunalno dobro". Cloud computing je upravo to, usluga dostupna javnosti putem globalne mreže-Interneta.

Nekoliko desetljeća kasnije, točnije 1997. godine, profesor Ramnath K.Chellappa je svojim znanstvenim radom doprinio da se službeno po prvi put, definira i akademski upotrijebi izraz 'Cloud computing" izražavajući pritom ideju koju su mnogo prije, podjednakim doprinosima, predvidjela dvojica pionira modernog računalstva.

Velika prekretnica dogodila se 1999. godine nastankom Salesforce.com-a. Salesforce je bio pionir u konceptu dostavljanja aplikacija preko jednostavnih web stranica. Njihovi poslužitelji su popločali put specijaliziranim programskim organizacijama za dostavljanje aplikacija preko Interneta.

Sljedeću ključnu ulogu odigrao je Amazon moderniziranjem svojih podatkovnih centara. Nova arhitektura oblaka rezultirala je značajnim i učinkovitim unutrašnjim poboljšanjima koje omogućava malim organizacijama lakše i brže dodavanje podataka u oblake. Prva njegova značajna inovacija bio je Amazon Web service 2002. godine. Pružao je usluge pohrane i obrade podataka. 2006. godine Amazon je objavio EC2,

koji je služio kao komercijalni web poslužitelj i mogle su ga iznajmiti male organizacije i individualni korisnici. Amazon EC2/S3 je bio prvi komercijalni Cloud computing poslužitelj.

Google, IBM i drugi uključili su se 2007. u veliki istraživački projekt daljnjeg razvoja. 2009. godine dogodila se još jedna važna prekretnica Web 2.0. Google i druge organizacije počele su nuditi aplikacije zasnovane na preglednicima.

Microsoft i Google su počeli objavljivati velik broj uspješnih aplikacija koje su brzo prihvaćene na tržištu. Sljedeći korak je bila virtualizacija tehnologije. Sve je to dovelo do toga da danas mnoge organizacije prepoznaju prednosti Cloud computinga i iz dana u dan sve se više koristi.

Očito je da su lokalni resursi ograničeni, stoga očekivano Cloud computing sve više vlada tržištem, pravi boom se tek očekuje. Ono što ga koči su sigurnosni problemi. Analitičari smatraju da će to biti uskoro riješeno. Nakon toga više nema granice za cloud computing i munjevit rast se očekuje. Poslužitelji će omogućiti korisnicima širenje vlastitih infrastruktura, dodavanje kapaciteta na zahtjev, povećat će se prilagodljivost, a u ponudi će biti i sve više različitih resursa, a to će povećati i uštede.

Organizacije imaju problem u Cloud computingu sa pohranom i obradom zahtjevnijih aplikacija. Međutim to je pitanje trenutka kada će riješiti, tako da je Cloud computing očekivani smjer razvoja računalnih resursa.

Bibliografija

[1] *Cloud Security Alliance*, http://www.cloudsecurityalliance.org/.

[2] *Cloud Security, Cloud Security Threats Survey*, http://cloudsecurity.org/.

[3] *Global Security Challenge*, http://www.globalsecuritychallenge.com/gsc_competitions.php#cloud.

[4] *Korištene slike su preuzete sa Google-a i Wikipedia*, https://www.google.hr/ http://en.wikipedia.org/.

[5] *MapReduce*, http://en.wikipedia.org/wiki/MapReduce.

[6] *mrjob*, http://pythonhosted.org/mrjob/index.html.

[7] *PiCloud*, http://www.picloud.com/.

[8] *Wikipedia, Cloud_Computing_Security,*, http://en.wikipedia.org/wiki/Cloud_computing_security.

[9] Jon Brodkin, *Gartner: Seven cloud-computing security risks*, 2008.

[10] Noah Gift, *Solve cloud-related big data problems with MapReduce*, white paper, DevelopersWorks, IBM, November (2010).

[11] Galen Gruman, *Early experiments in cloud computing*, InfoWorld. com, April **7** (2008).

[12] Eric Knorr i Galen Gruman, *What cloud computing really means*, InfoWorld **7** (2008).

[13] Alexander Lenk, Markus Klems, Jens Nimis, Stefan Tai i Thomas Sandholm, *What's inside the Cloud? An architectural map of the Cloud landscape*, Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, IEEE Computer Society, 2009, str. 23–31.

[14] Arif Mohamed, *A history of cloud computing*, Computer Weekly **27** (2009).

[15] Charalampos E. Tsourakakis, *Data Mining with MAPREDUCE: Graph and Tensor Algorithms with Applications*, Data Analysis Project, March (2010).

[16] R Revathy Kirpal A Venkatesh i Kishorekumar Neelamegam, *Using mapreduce and load balancing on the cloud*, white paper, DevelopersWorks, IBM, July (2010).

[17] Peter Wayner, *Cloud versus cloud: A guided tour of Amazon, Google, AppNexus, and GoGrid*, InfoWorld, July **21** (2008).

Sažetak

U ovom radu predočeno je računanje u oblaku pomoću programskog jezika Pythona, okruženje u kojem se računanje odvija te prednosti i nedostatci koje može donijeti korištenje ove tehnologije. U prvom poglavlju definiran je pojam računarstva u oblaku, njegova građa i karakteristike. Navedeni su modeli, tipovi i sudionici računarstva u oblaku, kao i razlozi zbog kojih bismo se trebali odlučiti za računarstvo u oblaku. U drugom poglavlju predočeni su MapReduce poslovi i njihova povezanost s računanjem u oblaku. Razrađen je jak alat Mrjob koji je poveznica između oblaka i samog procesa. Također, objašnjen je postupak za pokretanje procesa u oblaku (Amazonu), prikazani su primjeri i slike iz terminala i oblaka. Nadalje, opisan je pojam PiCloud i naveden je primjer procesa u njemu. U trećem poglavlju dana je ekonomska struktura, sigurnosni problemi te su predložena rješenja tih problema. Na samom kraju je opisan razvoj računarstva u oblaku kroz povijest te predviđanja smjera razvoja u budućnosti.

Summary

In this work we present a possibility to us the cloud computing paradigm using programming language Python. This is an environment in which computing takes place and the advantages and disadvantages that this technology can make. The first chapter introduces the concept of cloud computing, its structure and characteristics. In this chapter we also mention models, various types and users of cloud computing and reasons for using cloud computing. In the second chapter were mentioned MapReduce jobs and their connection with the cloud computing. We presented a strong tool named Mrjob which is the link between the cloud and the process itself. Also, we descibed a procedure for starting this cloud process on Amazon cloud and have provided examples of usage. The third chapter contains an application in economic analysis. Also we outline some possibilities of future development.

Životopis

Rođen sam u Sinju gdje sam 2002. godine završio osnovnu školu te potom upisao Opću gimnaziju Dinka Šimunovića u Sinju. Sudjelovao sam na raznim natjecanjima iz matematike te bio redoviti sudionik županijskih natjecanja, a najbolji mi je uspjeh državno natjecanje 2005. godine. Maturirao sam 2006. godine sa odličnim uspjehom. Iste godine sam upisao Preddiplomski sveučilišni studij matematike na Prirodoslovnom-matematičkom fakultetu u Zagrebu. Na istom fakultetu 2013. godine upisujem Diplomski sveučilišni studij Računarstvo i matematika.