

Универзитет у Крагујевцу
Факултет инжењерских наука



Назив студијског програма: Машинско инжењерство

Ниво студија: Основне академске студије

Модул: Информатика у инжењерству

Предмет: Базе података

Број индекса: 2/2008

Јован В. Поповски
Услужно рачунарство
завршни рад

Комисија за преглед и одбрану:

1. **др. Милан Ерић - ментор**

2. _____

3. _____

Датум одбране: _____

Оцена: _____

У оквиру овог завршног рада кандидат треба да проучи препоручену, као и ширу литературу која се односи на област услужног рачунарства. На бази услојеног знања кандидат треба да детаљно обради архитектуру услужног рачунарства. Основне моделе услужног рачунарства, виртуализацију, безбедност као и практичну примену услужног рачунарства.

Препоручена литература:

- [1] Weiss, Aaron. Computing in the clouds. *netWorker* 11, 4 (Dec. 2007),
- [2] <http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031>
- [3] Pollette, Chris. “How the Google-Apple Cloud Computer Will Work.” *Howstuffworks.com*
- [4] Introduction to Cloud Computing Architecture ,Sun Microsystems June 2009

...

Крагујевац, датум

ментор:

др. Милан Ерић

Резиме

Ова теза покушава да објасни основе услужног рачунарства , наравно са освртом на технолошке карактеристике и безбедност. Почињемо са историјатом , преко архитектуре и основних модела који су данас заступљени . Један од главних појмова је виртуализација на којој се базира цео систем и врсте скалирања података.

Са главним освртом на безбедност којом се бавимо у четвртом поглављу где је детаљно објашњен сваки аспект. Даље се разматра у следећем поглављу примена услужног рачунарства данас и ко су главни провајдери са циљем да читаоца што боље упозна са материјом и приближи овај концепт.

кључне речи: облак, виртуализација, услужно рачунарство

Abstract

This thesis seeks to explain the basics of cloud computing, of course, with emphasis on technology and security features. We begin with the history, architecture and more than basic models that are now represented. One of the main concepts of virtualization on which the whole system and the type of scaling data.

With major emphasis on safety that we do in the fourth chapter in which is explained in detail every aspect. It is further discussed in the next chapter cloud computing application today and who are the main providers in order to better acquaint the reader with the matter and to bring this concept.

keywords: cloud, virtualization, computing

Садржај

1. Увод	6
2. Услужно рачунарство	7
Архитектура услужног рачунарства	11
Основни модели услужног рачунарства	13
Основне карактеристике услужног рачунарства	15
Нивои услужног рачунарства	18
Врсте услужног рачунарства	22
3. Виртуелизација	26
Хоризонтално скалирање	28
Вертикално скалирање	29
Паралелизација	30
Физичка својства података	33
4. Безбедност услужног рачунарства	35
Безбедносне области услужног рачунарства	36
Сигурносна својства података	38
Мрежна безбедност у практичној примени	39
5. Практична примена услужног рачунарства	41
Amazon Web Services	44
Google App Engine	46
Microsoft Azure	49
6. Закључак	51
7. Литература	52

1. Увод

Употреба интернета и нових технологија данас, за пословање, личну употребу је постала неизоставна и као таква део свакодневног живота.

У новије време како је интернет напредовао и технологија заједно са њим, дошло је до великог помака у брзини протока информација и то је отворило широк спектар могућности као нпр. могућност да неку информацију сачувате на неком удаљеном уређају (eng. server) и да касније њој приступите са било ког места на планети.

Данас много људи користи разне услуге са интернет мрежа као што су читање електронске поште, комуникација, учење. Они заправо покрећу програме (eng. applications) са персоналних рачунара који приступају преко интернет мреже до сервера који те информације обрађују. Када корисник упише име сајта у интернет прегледачу (eng. browser) он почиње интеракцију са сервером који му даље дозвољава да дели, шаље, прима информације од којих поједине које могу бити приватне и важне које ће наравно бити заштићене. Свакодневно се све више користи услужно рачунарство (eng. cloud computing) . Име је настало од метафоре о самом интернету као нечему у облаку (eng. cloud) и наравно унутар кога се ништа не види.

Услужно рачунарство данас је све више у порасту. Али о чему се ту заправо ради ? Да ли је то исто као екстернализовање хостинг веб апликација ? Зашто би то било корисно и коме тј. ко би био циљна група? Како ће то и на који начин променити пословање предузећа? Узмимо за пример здравствене системе у установама које су заиста отворени , хетерогени и способни да подрже провајдере како мале тако и велике. У прошлости информациони системи су у предузећима били последња карика у низу. Можда би сада требало подићи ниво имплементације ових система у самим предузећима као и изван њих, где би имали значајну улогу од почетка до краја производног процеса.

Јасно се види да је услужно рачунарство погодно да задовољи захтеве предузећа. Али све док су ти захтеви комплементарни са оним чиме предузећа располажу. Чињеница је да услужно рачунарство може помоћи да се цео процес производње убрза и учини

флексибилнијим. Много је вероватно да ће се справити са великим захтевима двадесет првог века.

2.Услужно рачунарство

Услужно рачунарство можемо дефинисати као настајућу парадигму у свету информатике где подаци и услуге бораве у великим центрима којима се може приступити са било ког уређаја који има везу са интернетом.

Услужно рачунарство је начин обезбеђивања различитих услуга са виртуелних машина издвојених на врху великог машинског парка који живи - постоји у облаку. Оно долази до изражаја само када размислимо о томе шта је заправо био циљ информационих технологија, а то је да нађе начин да повећа капацитет и да дода већи спектар могућности у лету, без улагања у новију инфраструктуру, у обучавање новог особља или лиценцирања новог софтвера.

Данас када имамо толико компјутерских и меморијских могућности које се налазе у дистрибуираном окружењу око облака.(eng. cloud). Услужно рачунарство искоришћава могућности ових ресурса и чини их доступним као јединствени ентитет који може бити променљив како би задовољио тренутне потребе корисника. Основа овог типа рачунарства је да се састави група виртуелних сервера који ће бити на располагању са огромним ресурсима који ће бити дати корисницима на употребу. Сваки уређај са приступом интернету (рачунар, лаптоп, паметни телефон(eng. smartphone), таблет рачунар(eng. tablet pc)) се може користити за приступ ресурсима преко виртуелних сервера као што видимо на слици 1.



Слика 1 . илустрација система услужног рачунарства (edudemic)

Са пословне тачке гледишта, услужно рачунарство је поступак који се бави величином и доступношћу апликација, и тиче се примене истих како на великом, тако и на малом нивоу без великих проблема. Будући да се ресурс који је намењен за клијента се мења зависно од потребе клијента то се обавља без великих проблема.

Један од кључних принципа услужног рачунарства је да обрада 1000 података не мора да буде 1000 пута тежа. Како се количина података увеличава, услуге (eng.services) које услужно рачунарство поседује нам омогућава да се ефикасније справимо са већим проблемима и да обрада задатака буде лакша.

У доба класичних персоналних рачунара и сервера хардвер је био кључни фактор који је диктирао брзину и количину обраде података. Али са појавом услужног рачунарства цео аспект је бачен на циклусе и бит-ове (eng. bytes) тј. услуга се корисницима наплаћује на темељу броја циклуса извршених операција или број бајтова (eng. bytes) пренесених података. Хардвер тј. машине на којима се поступак обраде података извршава нису видљиви кориснику.

Количина потребног хардвера за извршавање процеса обраде података је брига руководства, а кориснику се врши обрачун зависно од тога како апликација користи та средства-ресурсе.

Тежња за иновацијама и непрекидан развој информационих технологија као што је раније у уводу речено довеле су до развоја услужног рачунарства, јавила се аутоматски и потреба за већим капацитетима и новим могућностима без великих улагања у постојећу инфраструктуру.

Пре настанка услужног рачунарства постојали су (корисник /сервер) и (peer to peer) модели. До 1980 тих година све је функционисало на основу модела корисник/сервер. Рачунарске апликације (програми) ,подаци и контрола извршавали су се преко класичних сервера(eng.mainframe). Уколико је корисник хтео приступити одређеним податцима или покренути програм , било је неопходно спојити се на сервер како би добио приступ податцима ,односно програму. Корисници су се спајали на сервере (eng.mainframe) преко терминала тј. (eng.client computer), и тај рачунар је имао онолико снаге колико је потребно за сврху спајања са сервером. Корисници који су хтели користити услуге сервера,могли су добити приступ само уз одобрење ИТ администратора, и није био могућ приступ више од једног корисника на сервер у исто време. Модел корисник/сервер иако сличан концепту услужног рачунарства није нудио квалитетну корисничку околину због ограничености управљања серверима (eng.mainframe).

Потреба спајања једног рачунара са другим резултирала је развојем P2P (eng peer to peer) технологија. Ова технологија представља мрежну архитектуру у којој сваки рачунар има једнаке могућности и одговорности, дизајнирана је као супротност претходној тј корисник / сервер технологији у којој је више рачунара зависило од једног сервера(eng.mainframe). У оваквом окружењу сваки рачунар је истовремено и корисник и сервер .Препознавајући једнако све рачунаре на мрежи, ова технологија омогућила је директну размену сервиса и услуга, те више није било потребе за једним главним „сервером“. Она је сама по себи децентрализујући концепт где рачунари функционишу по закону једнакости.

Потреба да више корисника тимски ради на једном документу омогућила је појаву услужног рачунарства. Прве колаборативне групе које су радила на једном документу у стварном времену за више корисника појавиле су се у доба P2P технологије (eng peer to peer). Како би могли сарађивати и деловати заједно на једном пројекту ,они пре свега морају имати међусобну комуникацију (тескт,звук,видео) .Корисници такође морају имати могућност међусобне измене како би то функционисало како треба.

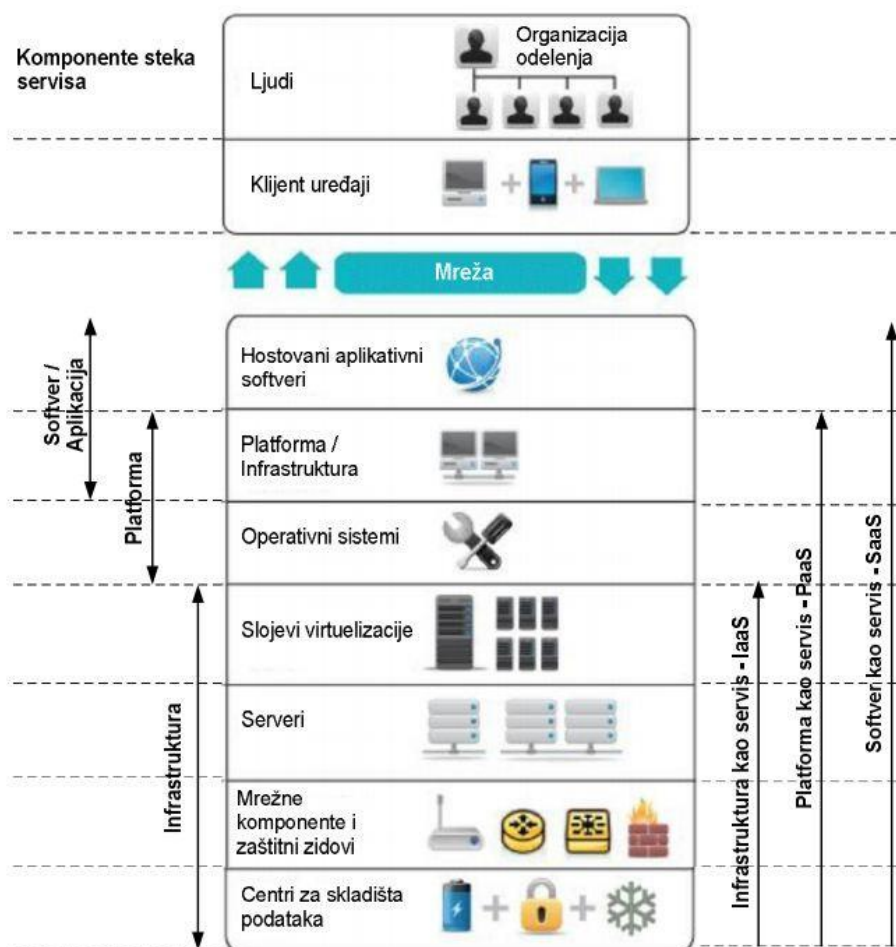
Растом интернета , није имало потребе за ограничавањем тимског рада на пројектима и документима на само једну мрежу већ то се примењује на глобалном нивоу за потребе већих компанија. За такав рад није битна географска локација , битно је само смештање података у облак ,како би му се могло приступити са било које локације на планети. Концепт услужног рачунарства развијао се у предузећима која су имала велике капацитете као што је Гугл(eng. Google). Огромне серверске капацитете искористили су за апликације базиране на интернету тј. облаку. Осим Гугла међу пионирима ове технологије су IBM и Sun Systems.Данас , људи користе ове сервисе како би складиштили, стварали, делили ,тражили и организовали податке свих врста ,наравно ова услуга није само за рачунаре већ и за мобилне уређаје,телевизоре нових генерација итд.

Архитектура услужног рачунарства

Као што смо у уводу објаснили сам појам cloud (срп. облак) је најчешће коришћена метафора за интернет. Када је користимо заједно са изразом computing (срп. рачунарство) тај термин више нема исто значење .Постоји гомила различитих дефиниција услужног рачунарства (eng. Cloud computing), али можда је најбоља од свих да је услужон рачунарство концепт поделе програмског окружења који користи интернет као платформу те омогућује да апликације и документи послати из било ког дела планете буду сачувани на за то унапред предвиђеним серверима. Ова врста рачунара која настаје заједно са развојем услужног рачунарства смањује потребу за куповином додатне опреме и програма те отварају нови облик сарадње. Приступ „податцима у облаку“ се одвија преко прегледача или апликација предвиђених за исто.

„Облак“ одваја апликативне и информационе ресурсе од инфраструктуре која лежи у позадини .Облак одваја и механизме којима се ти ресурси достављају. Такође повећава сарадњу, покретљивост, скалирање и приступачност рачунарских система те доноси нове могућности развоја апликација у тимском окружењу. Услужно рачунарство дефинише начин употребе више сервера ,апликација ,информација и инфраструктуре која се састоји од великог броја рачунарских, мрежних, информацијских ресурса и ресурса за складиштење података и програма како би се смањили финансијски трошкови корисника и омогућило брже и јефтиније решење проблема . Ове компоненте могу бити брзо процесуиране , интерпретиране, примењене и скалиране пружајући тако модел расподеле и употребе који функционише на захтев корисника.

На слици 2. је илустрација архитектуре „у облаку“ према књизи (Introduction to cloud computing Architecture) од Sun Microsystems-а. У истој је главни акценат бачен на три категорије : софтвер као услуга **SaaS** (eng. Software as a Service), платформа као сервис **PaaS** (eng. Platform as a service), и ИТ инфраструктуру. Наравно постоје и међупреклапања као што се види на слици .О овим а и неким другим облицима услужног рачунарства више у следећем поглављу.



Слика 2. Архитектура система услужног рачунарства (PCWorld)

Основни модели услужног рачунарства

SaaS (eng. Software as a Service), модел услужног рачунарства који преко интернет прегледача (eng. browser), доставља једну апликацију ширем броју корисника. Коришћењем овог модела корисници не морају да врше инвестиције у нове сервере и лиценцирани софтвер. Трошкови провајдера услуга су при томе мањи у односу на стандардну услугу чувања података на серверу. Детаљнији опис SaaS услуге налази се у поглављу о моделима пружања услуга.

Web services in cloud (срб. Интернет услуге у облаку) Интернет услуге су уско повезане са **SaaS** моделом. Организације које омогућавају интернет услуге нуде корисничка окружења, која омогућавају коришћење услуга преко интернета. Интернет услуге могу имати широк распон делатности тако да се простиру од дискретних пословних услуга (Strike Iron и Xignite) , па све до јако добро развијеног сервиса Google Maps, или аутоматске обраде података након плаћања кредитним картицама преко интернета Google Checkout.

PaaS (eng. Platform as a service) PaaS је још једна врста претходно поменутог SaaS модела. Овај модел „облака“ као услугу пружа развојну околину. Корисник гради своје апликације које се покрећу на инфраструктури даваоца услуге- провајдера , па се преко прегледача достављају кориснику.

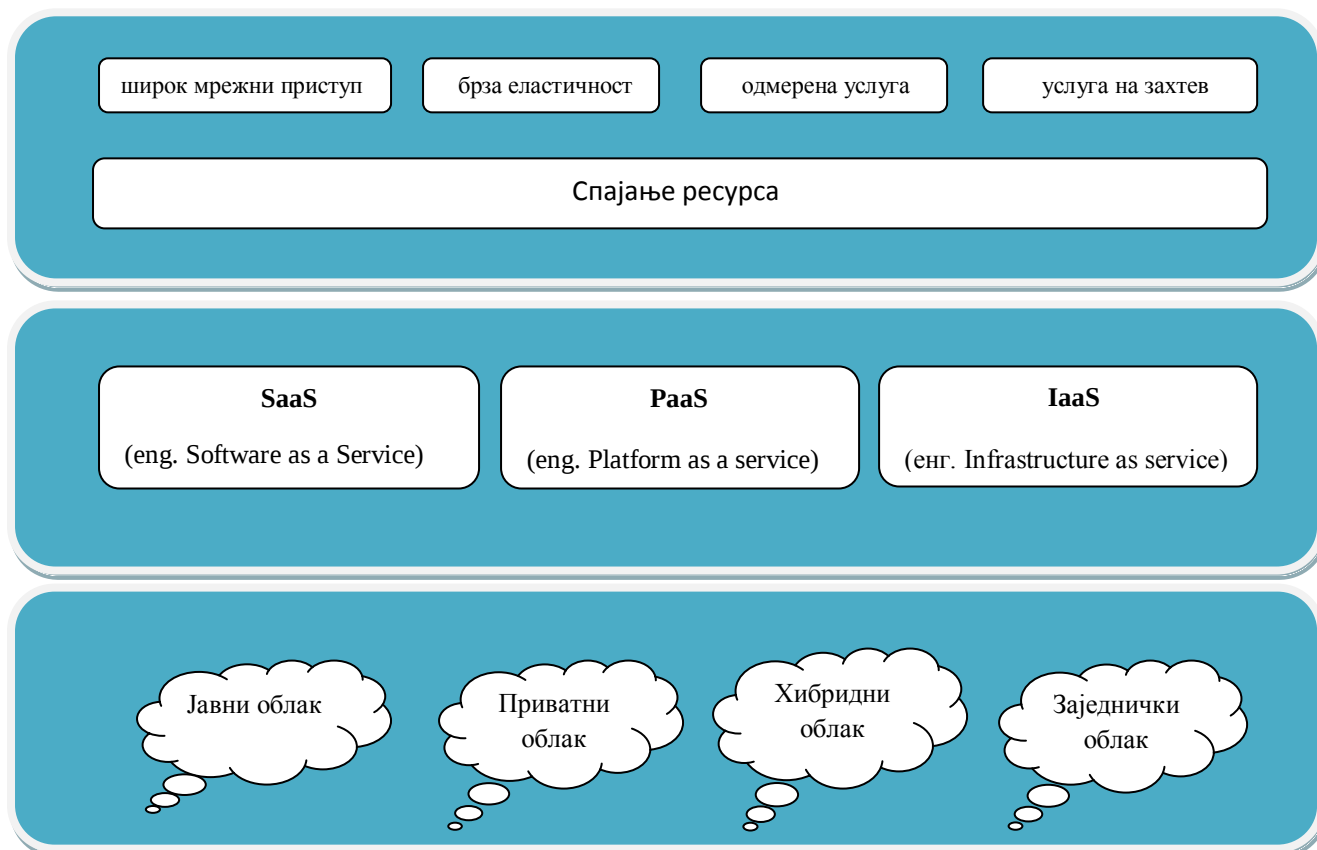
MSP (eng.Managed service providers) MSP је један од најстаријих облика модела услужног рачунарства .Управљана услуга је апликација која је искључиво намењена информатичкој служби а не крајњем кориснику. Пример је услуга коју нам омогућава софтвер када скенира злоћудне поруке електронске поште или услуге управљања апликацијама као што је на пример Mercury.

Услуге комерцијалних платформи (eng. Service commerce platforms)- Ове су платформе хибридне верзије SaaS и MSP модела. Услуга комерцијалних платформи нуди чвориште (eng. hub) преко кога корисници комуницирају. Најчешће се употребљава у трговинама преко интернета, попут скупих управљачких система који корисницима дозвољавају наручивање мобилних услуга заједничких платформи. Платформе тада координирају пружање услуга одређивањем цене унутар спецификација које је поставио корисник.

Интеграција интернета (eng. Internet integration) Данас је интеграција услуга у „облаку“ и даље у својим раним фазама. Начин међусобног повезивања заснован на „облаку“ би се можда требао назвати sky(срб. небо) computing (срб. рачунарство) са много облака на које се корисници могу индивидуално повезивати. На другој страни све више фирми и организација примењује виртуализацију и SOA архитектуру (eng. Service oriented architecture), повећава се потреба за добро повезаним услугама које се налазе на интернету. Идеја овакве инфраструктуре је да се једног дана свакој организацији направи чвор у облаку.

Основне карактеристике услужног рачунарства

Према Америчкој владиној организацији која је бави стандардима и технологијом NIST (eng. National Institute of Standards and Technology) се међу осталим делатностима бави и услужним рачунарством. Тачније ова се организација бави дефинисањем овог типа рачунарства и поделом на карактеристичне моделе. Према њима се услужно рачунарство састоји од 5 кључних карактеристика, три модела пружања услуга, и четири модела имплементације које су наведене на слици 3.



Слика 3. Основне карактеристике (NIST)

Постоји **пет кључних карактеристика** услужног рачунарства које показују разлику у односу на класичан приступ рачунарству.

Пружање услуге на захтев корисника (eng. On demand self service) ,корисник може самостално одабрати и покренути рачунарске ресурсе.Може такође бирати време када ће се вршити услуга и мрежни простор за чување података без потребе за посебном интеракцијом са особљем даваоца услуге. Данас већина провајдера своје услуге заснива на овом приступу да корисници плаћају услуге у зависности од времена и обима коришћења. Ова особина (eng.Self service) организацијама и тимовима омогућује стварање флексибилне околине која се повећава и смањује зависно од радних потреба и услова. Ова особина тј. „наплата по коришћењу“ „услужног рачунарства“ се може сматрати као позајмљивање опреме које се плаћа по томе у којој количини и на које време је изнајмљена. Виртуелизација је кључ овог модела. Организацијама које користе информационе технологије јасно је да им виртуелизација омогућава брзо и једноставно стварање копија постојећих окружења, понекада укључујући више машина истовремено како би омогућили испитивања или рад. Трошак при коришћењу ових окружења је мали јел оне не могу постојати на истом серверу као производна околина. Исто тако нове апликације се могу развијати и распростирати у новим виртуелним окружењима и машинама које се налате на више сервера који су отворени за рад на интернету.Апликације могу бити скалиране ,ако су успешне на тржишту . Могућност коришћења и плаћања само оних ресурса који су у употреби пребацује бригу колико инфраструктуре заузети при раду са „облаком“ са корисника на провајдера услуга. Такође помера и одговорност за одлуке које се тичу распоређивања ресурса.

Широк мрежни приступ (eng. Broad network access), Ресурси су доступни путем мреже и њима се приступа користећи стандардне механизме који форсирају ширу употребу тањих или богатијих клијентских платформи (пример мобилни уређаји, лаптопови,PDA уређаји,паметни телефони) као и традиционалних програмских услуга заснованих на „облаку“.Ово је врло блиско Мајкрософтовој П + У / програм + услуга (eng. S+S/software + service) стратегији ,наравно основна идеја је да било који уређај може приступити систему са било ког дела планете.

Удруживање ресурса (eng. Resource pooling) Рачунарски ресурси провајдера спајају се како би омогућили свим корисницима рад, и при томе користе користе модел више закупљених јединица (eng. Multi-Tenant model), са различитим физичким и виртуалним ресурсима, који се динамички додељују и уклањају према захтевима корисника . Корисник уобичајено нема надзор и знање о тачном месту употребљених ресурса, али ипак може учествовати у делимичном одређивању на већем нивоу апстракције .Примери ресурса укључују мрежни простор ,процесоре, меморију, мрежни протока па и виртуелне машине.

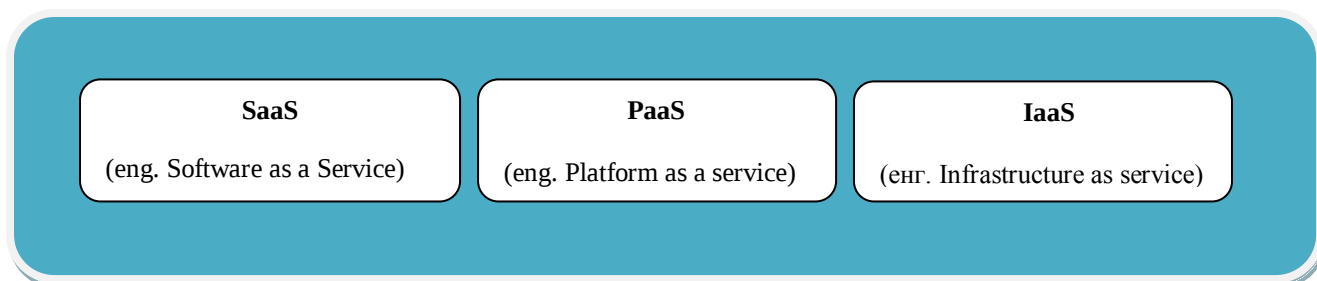
Брза еластичност (eng. Rapid elasticity) Могућности које корисницима нуди „рад у облаку“ могу бити брзо и флексибилно покренуте, у неким случајевима и аутоматски, како би се по потреби остварило пропорционално повећање или смањење могућности када оне више нису потребне. Крајњем кориснику могућности које користи могу изгледати као да немају ограничења и могу се купити у било којој количини и у било које време .

Одмерена услуга (eng. Measured service) Системи који користе услужно рачунарство аутоматски проверавају и оптимизују употребу ресурса. Употреба ресурса се оптимизује тако што се пажња обрати на степен апстракције који је неопходан ,тј да ли се ради о чувању података,или некој сложенијој операцији. Даље употреба ресурса се може пратити проверавати па самим тим имати и увид о самој услузи која је омогућена корисницима.Важно је напоменути да се сервери са овим услугама често али не и увек користе заједно са технологијама заснованим на виртуелизацијом. Међутим не постоје захтеви који уско везују апстракцију средстава и виртуализационе технологије па некада се и избегава коришћење виртуализације.

Нивои услужног рачунарства

Услужно рачунарство није само једна врста технологије . Оно заправо представља комбинацију много постојећих технологија а то су виртуелизација , чување и смештање података . повезаност и процесорска снага који чине ново окружење. Примена ове технологије је у великом успону ,омогућује предузећима и организацијама бољи фокус на пословање а у исто време омогућава информатичком одељењу смањивање разлике између доступног рачунарског капацитета и постојећих захтева . Као резултат тога добијамо већу ефикасност и бољу искоришћеност капацитета. Архитектура услужног рачунарства се састоји од модела пружања услуга (Saas, PaaS, IaaS), Типова услужног рачунарства (јавних,приватних и хибридних „облака“).

Опште прихваћена подела модела пружања услуга је SPI модел (eng. Software-Platform-Infrastructure),овај акроним представља три главна сервиса (eng. SaaS- Software as a Service , PaaS – Platform as a Service i IaaS – Infrastructure as a Service) , инако постоји још врста ове поделе , ова је прихваћена према NIST у (National Institute of Standards and Tehhology) на слици 3 , и у даљем тексту се бавимо сваком технологијом појединачно.

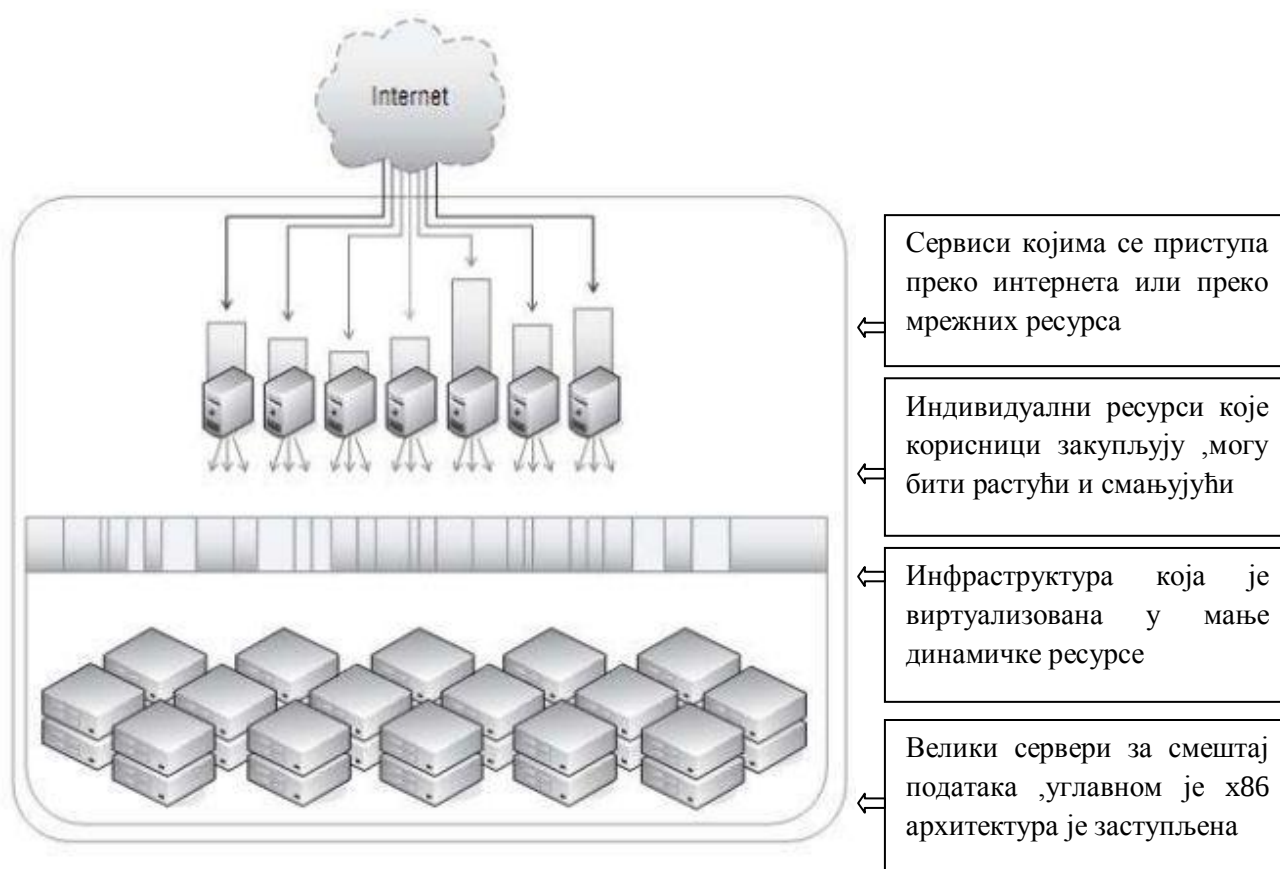


Слика 3. Основни модели пружања услуге (NIST)

- **SaaS (eng. Software as a Service)** Кориснику је пружена могућност употребе доступних апликација које се налазе у инфраструктури облака. Апликације су доступне са различитих корисничких окружења (нпр. интернет прегледача). При томе корисник не мора проверавати позадину структуре, коју чини мрежа, сервис, оперативне системе или програмске могућности. Једини могући изузетак су специфични захтевни корисника. Другим речима SaaS је технолошка платформа која омогућава доступност програма – апликација преко интернета у облику услуга које се изнајмљују према потреби, уместо да се купују као посебни програми који се инсталирају на кућним (односно канцеларијским) рачунарима. У великом је расту тренд прелаза на тај пословни модел, који предузећима омогућава изнајмљивање програма за текстуалних податак, табличних, календарских или других програма према потребама, чиме у старту се избегава трошак куповине инсталације, надоградње и одржавања програма на канцеларијским рачунарима. Овај модел услужног рачунарства доставља једну апликацију преко корисничког прегледача хиљадама корисника који користе архитектуру предвиђену за већину претплатника. Из корисничког угла то значи да нема додатног улагања у сервере или програмске лиценце, а провајдерима услуга трошкови су мали у односу на традиционалну услугу држања датотека на серверу. Пример SaaS (eng. Software as a Service) су Google Apps. Корисници SaaS а не морају бринути о компатибилности хардвера, оперативног система и софтвера, лиценцама, неауторизованим копијама софтвера, одржавању, подршци и ажурирању. SaaS (eng. Software as a Service) пружа много користи за организациону структуру смањују се и трошкови лиценцирања, управљања хардвером и мањи трошкови коришћења апликације. Једноставнији је приступ апликацији преко прегледача. Нема дуплирања софтвера и додатних надограђивања.

- **PaaS (eng. Cloud Platform as a Service)** ово је заправо варијанта . SaaS a (eng. Software as a Service) која као услугу доноси развојно окружење .Корисник сам гради своје апликације које се покрећу на инфраструктури провајдера услуге. Апликације се корисницима достављају преко окружења сервера приступачног преко интернета. Наведени сервери су у власништву провајдера услуга. Ове услуге су ограничене дизајном и могућностима достављаоца тако да корисник нема потпуну слободу. Корисник не може проверавати структуру облака нити мрежу, системе чувања, оперативне системе и сервере, али ипак има надзор над развијеним апликацијама. Понекада чак има и могућност прегледа конфигурације окружења. Неки од примера су Coghead и Google App Engine. Користећи концепт PaaS a (eng. Cloud Platform as a Service) не морамо инсталирати развојни софтвер на свој рачунар већ користимо Web апликацију, затим и тај софтвер можемо сместити у облак. PaaS a (eng. Cloud Platform as a Service) укључује слој софтвера за развој већих апликација и сервиса. Сервер нуди корисницима развојно окружење, апликациони стандард, алате за израду и дистрибутивни канал за готове апликације. PaaS a (eng. Cloud Platform as a Service) нуди могућност дизајна апликација са малим трошковима ,која притом елиминише потребу за скупим хардверским и софтверским ресурсима.Заправо служи као комплетно решење за развој , тестирање и пласман апликације
- **IaaS (eng. Infrastructure as a Service)** Кориснику је као услуга пружена могућност да користи рачунарску инфраструктуру (то се углавном односи на виртуелне платформе) . Корисници не купују сервере ,програме ,просторе за смештање података или мрежну опрему ,већ купују наведене ресурсе као спољашњу услугу. Кориснику је пружена могућност да управља обрадом , чувањем података , умрежавањем и другим основним рачунарским ресурсима. Корисник може покренути различите врсте програмске подршке, од операционог система до апликација. Корисник нема надзор над инфраструктуром облака, али има надзор над операционим системима и, чувањем података и развојем апликација. Корисник може имати и ограничен

надзор над одређеним компонентама умрежавања. Неки од провајдера IaaS услуге су IBM, Oracle, Terremark ,Dropbox, Amazon simple storage service ,најзначајнији од свих наведених је Amazon са великом понудом различитих сервиса. На слици 4 је приказан пример IaaS сервиса, од велике просторне и мрежне инфраструктуре, виртуелизованих мањих ресурса до индивидуалних корисника којима се нуде сервиси на интернету.



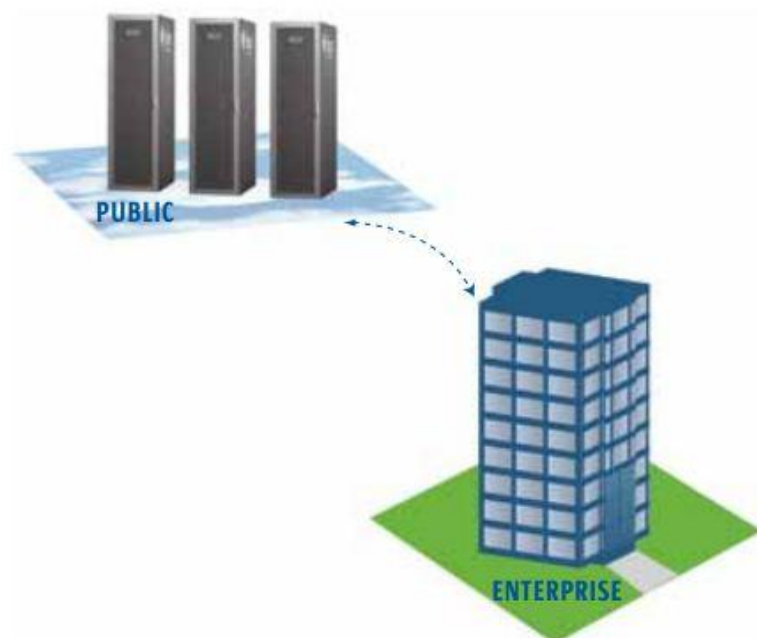
Слика 4. Пример IaaS модела (Krutz and Vines, 2010)

Врсте услужног рачунарства

Постоје четири врсте услужног рачунарства које можемо окарактерисати као основне:

- **Јавни облак (eng. Public Cloud)** то је cloud computing (срб. услужно рачунарство) платформа доступна и отворена за јавност, било да се ради о појединцу или организацији. Он се налази у власништву фирме која продаје услуге cloud computing -а (срб. услужног рачунарства). У случају јавних платформи поставља се питање сигурности сопствених података. Апликације различитих корисника често се налазе на истим серверима, системима за чување и мрежама. Јавни облаци смањују сигурносне ризике и трошкове пружањем променљиве инфраструктуре. Они чине привремено закупљену инфраструктуру организација. Ако је јавни облак реализиран са пажњом усмереном на перформансе, сигурност и положај података друге апликације покренуте на облаку не би требало стварати проблеме према архитектури облака и крајњим корисницима. Једна од предности јавних облака је да они могу бити пуно већи него што могу бити приватни облаци. Јавни облаци нуде могућност повећања или смањивања закупљеног дела облака и пребацивање одговорности, ако се појаве непланирани ризици, са организација на провајдера. Делови јавног облака могу бити и под искључивом употребом само једног корисника, чинећи тако приватни центар са податцима. (eng. datacenter). Заузимање такозваних слика виртуалних машина (eng virtual machine images) у јавном облаку не даје корисницима потпуни увид у инфраструктуру облака, док закупљивање центара са податцима даје корисницима увид у саму инфраструктуру. Тада корисници могу управљати не само са сликама виртуелних машина, већ и серверима, системима чувања података, мрежним уређајима. Стварање приватног виртуалног центра са податцима са компонентама које се налазе у истом објекту смањује проблем постојања већине различитих локација података зато што је брзина преноса пуно већа при повезивању података унутар истог

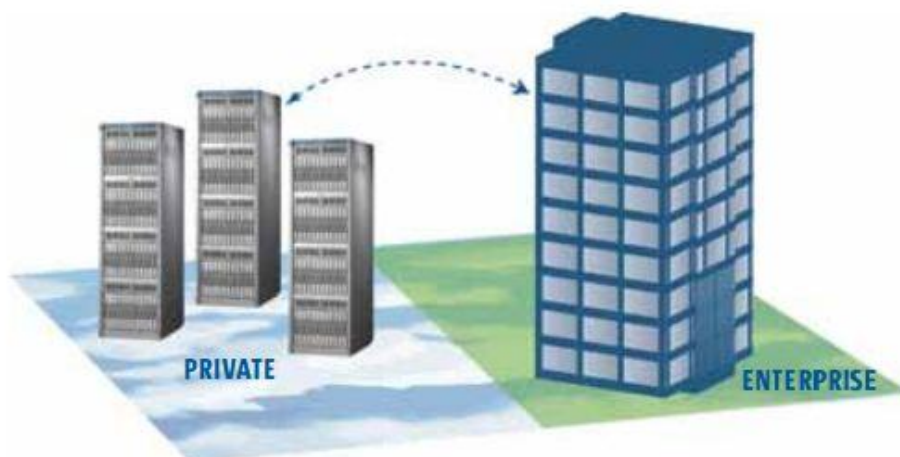
облака. Примери јавних облака су Amazon Web Services, Google App Engine, Microsoft Windows Azure. На слици 5 се налази илустрација јавног облака.



Слика 5. Илустрација јавног „облака“ (Sun Microsystems)

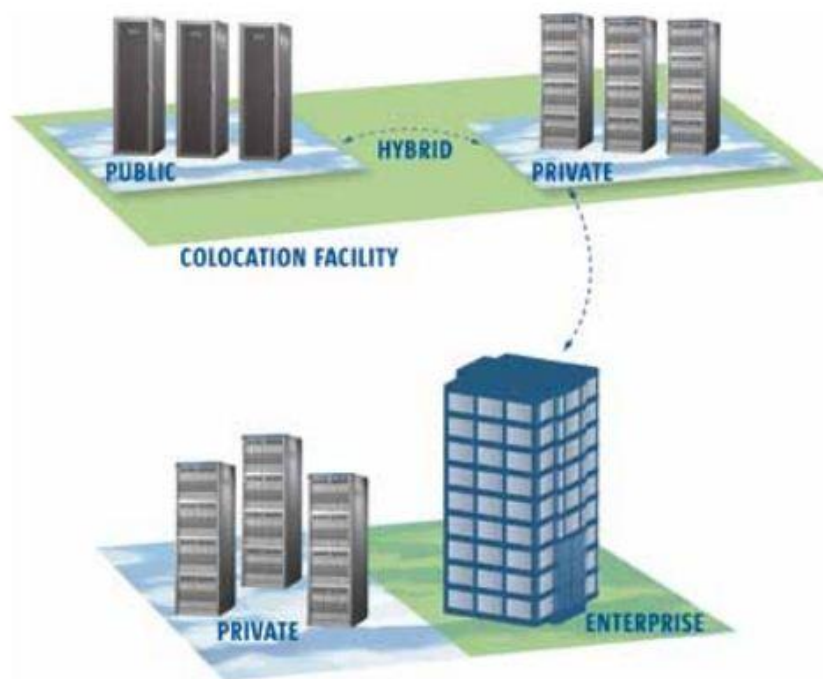
- **Приватни облак (eng. Private Cloud)** Cloud computing инфраструктура доступна је искључиво једној организацији. Са њом може управљати сама организација или неко други . Организације користе приватне облаке када им је потребан или желе већи преглед над податцима него што могу добити коришћењем јавног облака . Приватни облаци су направљени искључиво за једног клијента , пружајући му преглед над податцима и већу сигурност својине која се налази на серверу. Организација поседује инфраструктуру и има преглед при расподели апликација на својој инфраструктури . Приватни облаци могу бити распоређени и унутар информатичког центра организације. ИТ службе компанија или провајдери услуга граде своје приватне облаке и управљају њима . Организације које поседују приватни облак на њему могу инсталирати програме, чувати податке и управљати структуром облака. Такође приватни облици пружају компанијама високи ниво надзора над

коришћењем ресурса самог облака јер коришћењем приватног облака организације имају потребле вештине и могућности за успостављање и управљање околином. На слици 6 је илустрација приватног облака.



Слика 6. Илустрација приватног облака (Sun Microsystems)

- **Хибридни облак** је заправо комбинација јавних и приватних облака који се такође може користити за управљање планираним великим оптерећењима .Приватни облаци се могу користити за извођење периодичних задатака који се једноставно распоређују на јавне облаци . Хибридни облаци се сусрећу са сложености одређивања како да се расподеле апликације по јавном и приватном облаку. Поред овог проблема мора се узети у обзир и однос између података и обраде ресурса. Ако су потатци мали или апликације не памте стања , хибридни облак може бити боље решење од преписивања велике количине података у јавни облак(у којем се изводи једноставна обрада). На слици 7 је илустрација хибридног облака.



Слика 7. Илустрација хибридног облака (Sun Microsystems)

- **Заједнички облак (eng community cloud)** Неколико организација дели структуру облака. Инфраструктура подржава посебне заједнице које имају заједничке потребе, мисије, сигурносне захтеве и слично. Њима могу управљати саме организације или неко други (даваоц услуга услужног рачунарства). Хибридни облак повезује јавне и приватне моделе облака. Моогућност проширивања приватног облака са ресурсима јавног облака може се употребити за одржавање услужних домена како би се лакше издржала велика оптерећења. То се најчешће може видети код употребе облака за чување података како би подржали Web 2.0 апликације.

3. Виртуелизација

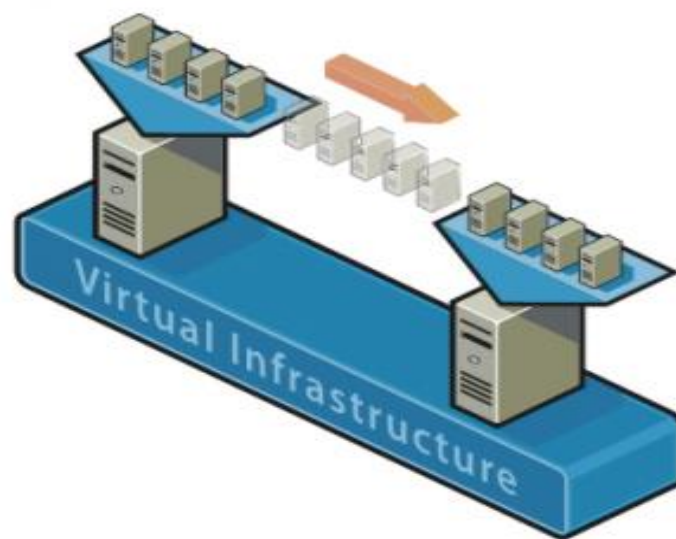
Циљ виртуелизације је да током времена увећа вредност предузећа у односу на инфраструктуру, затим смањење оптерећења хардвера, потрошњу енергије, трошкова и информатичког система, повећавајући тако еластичност система. Трошкови су делимично смањени због сервиса који се нуде бесплатно као што су Microsoft Virtual Server или Vmware Server. Осим користи од смањења трошкова, виртуелизација доноси предност у бољој искоришћености ресурса (физички сервери су ретко у целости испуњени, процесом виртуелизације се повећава искоришћеност са 5% на 60 – 80% , са тиме да више виртуелних сервера буде на једном физичком серверу). Виртуелизација се може применити и на оперативни систем , платформу, простор на диску , мрежу (симулације) и на саме апликације.

Са концептом виртуелизације се први пут срећемо 1960 године (Имао га је IBM). Од тада се наставља са континуираним напретком и полако се примењује на меморију, чување података , процесе, софтвер, мреже и све остале сервисе које су пружале информационе технологије. Данас у понуди имамо виртуалних решења , нека од њих су Vmware, Virtual Box , Xen.

Последњих неколико година виртуелне машине постале су стандардни развојни објект . Виртуелизација повећава флексибилност ,пошто чини да хардвер изгледа до те мере апстрактно да програми могу бити дистрибуирани без директног везивања на конкретан физички сервер. Виртуелизација омогућава постојање једне динамичке базе података ,где физички сервери омогућавају ресурсе колико је потребно да би та база функционисала ,тј. само по потреби. Однос програма за обрачун , чување и мрежне ресурсе се динамички мења како би се постигло захтевано радно оптерећење и практичност система. Имплементација програма је одвојено од имплементације сервера ,па се програми могу брже скалирати и распоређивати ,без потребе за физичким серверима . Виртуелне машине постали су главна апстракција јер су они најмање заједничко окружење које дели даваоце услуга и развојне организације.

Употребом виртуелних машина као развојних објеката могу се задовољити 80 % корисничких потреба,а и помаже при регруписању и скалирању програма. Облаци за рачунање су најчешће физички повезани са облацима за чување података. Облаци који су намењени за чување података пружају виртуално чување преко API ја које олакшава

виртуалним машинама чување слика, датотека или компоненти типа веб сервера , апликацијских и пословних података. Виртуалне машине и апликације побољшавају способност брзог стварања и развијања апликација . кључ даљег развоја услужног рачунарства лежи у комбинованом коришћењу виртуелних машина и програма-апликација. На слици 8 је илустрација виртуелног система.



Слика 8. Илустрација виртуелизације (dtm.ca)

Хоризонтално скалирање

Услужно рачунарство сачињава масивна количина хоризонталне скалабилности над апликацијама које могу имати користи од тога . Новији трендови ка пројектовању и реорганизацији апликација да раде у хоризонтално мањим срединама заправо значи да се број апликација које су прилагођене за „облак“ повећао.

Апликације које имају предности од хоризонталног скалирања требало би да су написане тако да буду фокусиране на укупну расположивост која је намењена истима, наравно са претпоставком да ће поједини делови компоненти тих апликација бити неупотребљиви. Већина оваквих платформа „у облаку“ су изграђене на виртуелној бази серверских ресурса, тако да ако би неки од физичких ресурса не успе да се конектује онда аутоматски виртуелна машина на којој се налази хостинг се рестартује на неки други физички сервер. Комбинација без физичких стања и слабо спрегнутих апликативних компоненти са хоризонталним скалирањем промовише нову стратегију која не зависи од поузданости било које од компоненти .

Хоризонтално скалирање не мора бити ограничено на само један „облак“ . У зависности од величине и локације података „велики талас“ може бити употребљен за проширење облака. Способност да се прилагоди привременом повећању обима посла. У данашњем рачунарству које је у свакодневном порасту, апликација која ради у једном приватном облаку може уполити додатна средства-ресурсе из јавног облака ако се за то укаже прилика.

Рачунарство данас умногоме зависи од количине података које се налазе на локалу. У случају приватног облака који се налази у пословном центру са податцима који се шири за потребе коришћења на неком месту на интернету, количина података која треба бити пребачена на приватни „облак“ треба да буде урачуната у једначину (на слици 9 даље у тексту). У случају да се приватни „облак“ налази на хосту где је јавни сервер, питање смештања података знатно је решено јер је простор за то практично неограничен, слободан проток повезује ова два облака.

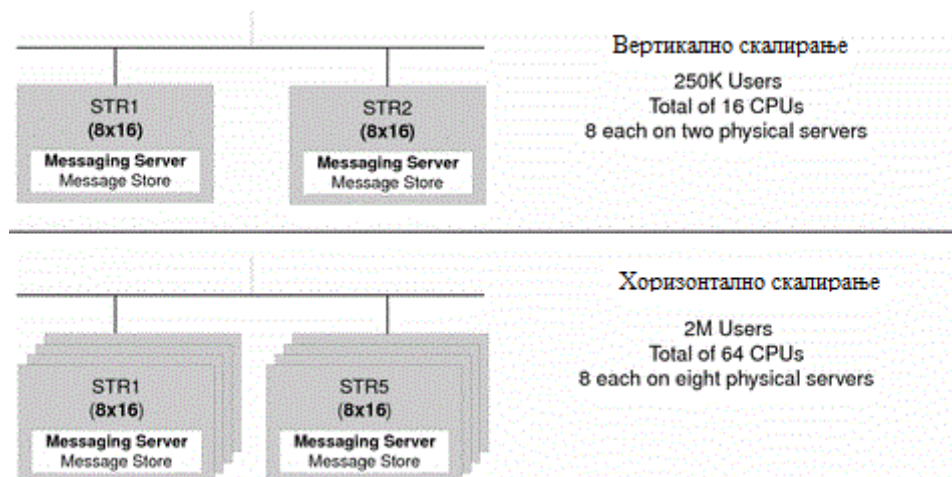
Вертикално скалирање

Питање вертикалне скалабилности је врло важно када разматрамо коришћење платформе у „облаку“ , јер се често плаћање на основу рачунарских ресурса који се користе налик на старије сервер(eng. mainframe) моделе. Ако је сама апликација лоше скалирана , то ће свакако повећати трошкове за покретање у облаку.

Провајдери услужног рачунарства не могу, а и не би ако би могли (то би их дакако коштало), да се баве адресирањем вертикалне скалабилности , јер је тај проблем засебан за сваку апликацију појединачно. Не постоји спољашње решење које може оптимизовати код до те мере да максимално појача верикалну скалабилност . Спољна решења могу дакако повећати перформансе , то постижу оптимизацијом протокола и над апликацијама, и смањењем потреба за пропусним опсегом , али могу да преуређују код и редослед по коме се врше спајања унутар SQL упита , да преправе лоше написанз петљу или део кода да би се ефикасније користили податци, самим тиме добија се и боља структура података.

Вертикална скалабилност, да ли је апликација распоређена унутар локалног центра са податцима или у самом „облаку“ , и даље је у домену апликативног програмера. Док се програмери могу користити технологјама као што су мрежно скриптовање и инхерентне карактеристике у примени испоруке решења која се користе са наменом повећања вертикалне скалабилности ,постоји ограничење које се односи на то каква решења можете направити и какав би она успех имала са решавањем узрока неког захтева који је неуспешно скалиран.

Побољшање вертикалне скалабилности апликација је важно за постизање предности у смањењу трошкова везаних како за услужно рачунарство тако и за виртуелизацију . Апликација која је неуспешно вертикално скалирана , на крају може коштати више ако би се њен развој обављао у „облаку“ због додатних захтева око ресурса и средстава која су неопходна за обављање таквог задатка. На слици испод текста је илустрација са приказом разлика између вертикалног и хоризонталног скалирања (слика 9).



Слика 9. Упоредбање хоризонталног и вертикалног скалирања (Oracle)

Паралелизација

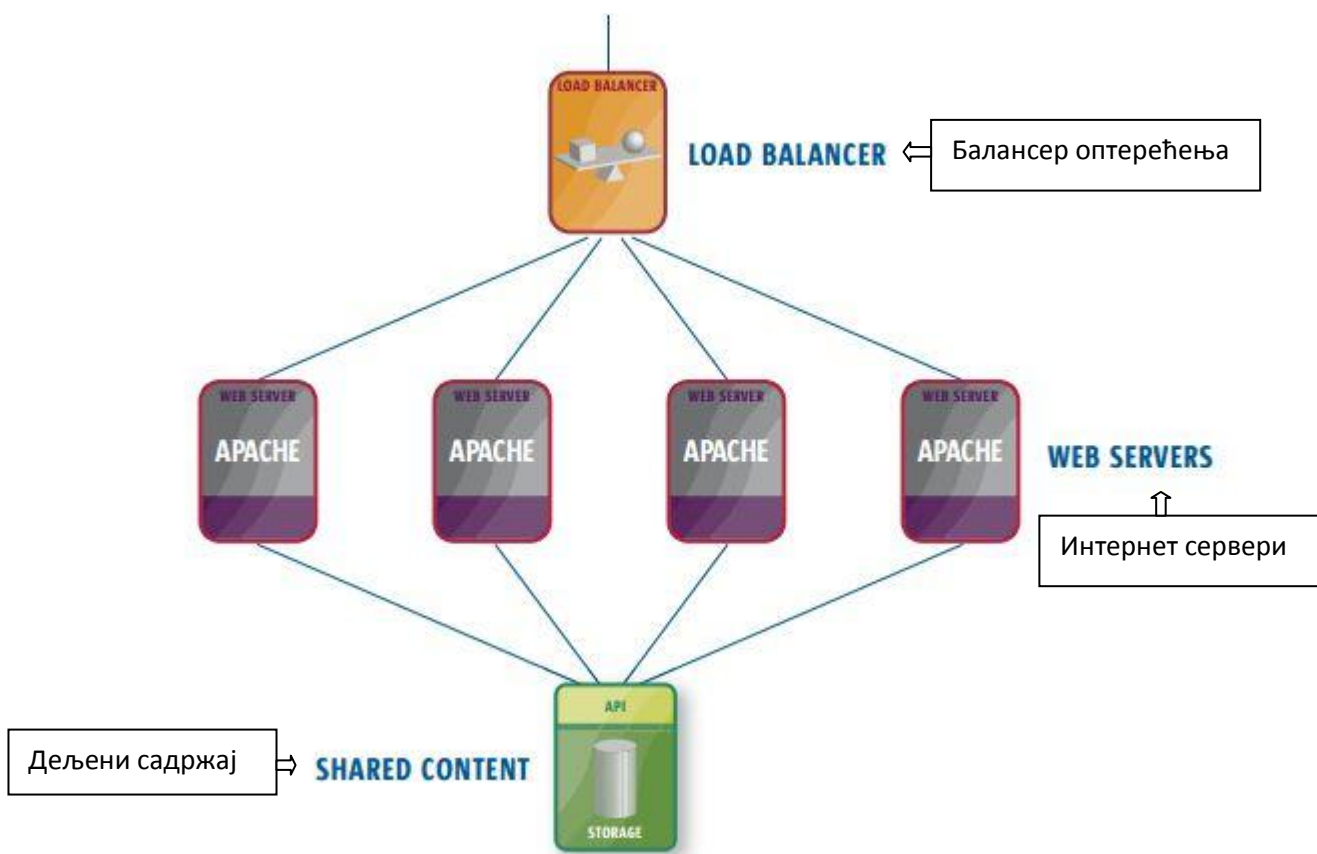
Хоризонтално скалирање и паралелизација иду скоро увек заједно, дакако данас је однос размере и имплементације промењен. На микроскопском узорку жаргонски речено у компјутерском свету, софтвер може да користи вертикално скалирање на симетричним мултипроцесорским јединицама у циљу да направи више тема где паралелизација може убрзати операције и повећати време одзива. Али са данашњим компјутерским окружењима која се полако пребацују на подножја са два или четири подножја, вертикално скалирање има само паралелне обраде колико и сервер језгара.

На макроскопском нивоу, софтвер који користи паралелизацију низ више сервера може да скалира на хиљаде сервера, нудећи тако више скалабилности него што би било могуће са симетричним мултипроцесингом.

У физичком свету паралелизација је често имплементирана заједно са балансерима оптерећења или прекидачима садржаја који имају намену да дистрибуирају долазне захтеве преко једног броја сервера. У свету услужног рачунарства, паралелизација може бити имплементирана заједно са балансерима оптерећења или прекидачима садржаја (eng. content switch) који дистрибуира захтеве преко једног броја виртуелних машина. У оба случаја, апликације могу бити

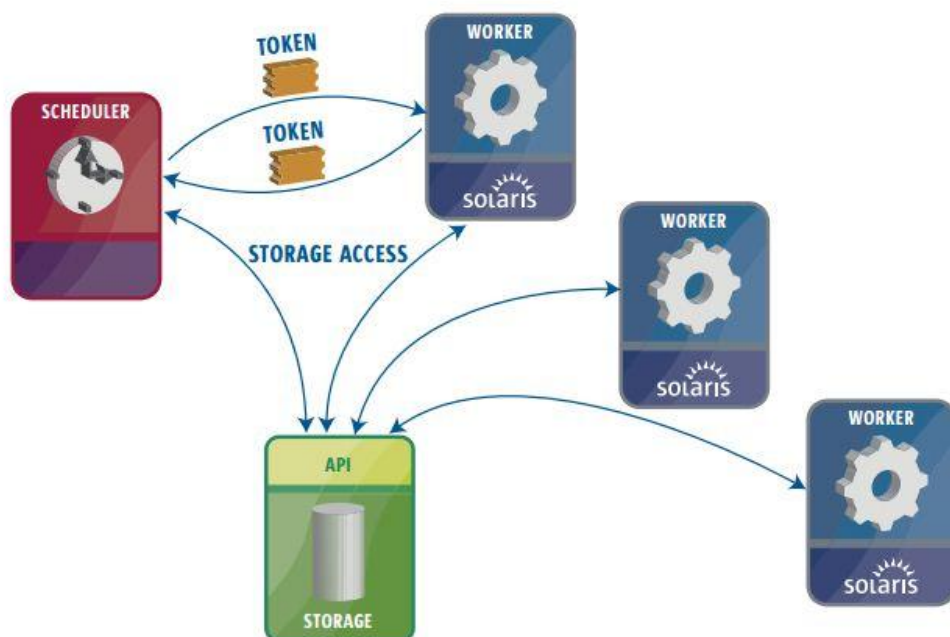
дизајниране да упошљавају додатне ресурсе да би сместили импулсна оптерећења (eng. workload spikes) .

Класичан пример паралелизације са балансирањем оптерећења је више веб сервера без тренутног машинског стања(eng. stateless) који истовремено покушавају да приступе податцима, где се долазни саобраћај регулише преко више сервера. На слици 10 је пример коришћења паралелизације и распоређивања оперећења користећи хоризонтално скалирање.



Слика 10. Класичан пример паралелизације (Sun Microsystems)

Постоји много других начина да се користи паралелизација у системима услужног рачунарства . Примена која користи значајан однос процесорског времена при обради података може користити модел који је приказан на слици 11 . Распоређивач добија послове од корисника , смешта податке у репозиторијум, онда стартује по једну виртуелну машину за сваки посао , онда даје виртуелној машини знак да омогућава преузимање података из репозиторијума . Када виртуелна машина заврши свој задатак , она даје знак распоређивачу који омогућава да се пројекат који је успешно завршен врати кориснику, и виртуелна машина се гаси .



Слика 11.Паралелизација (Sun Microsystems)

Физичка својства података

Под појмом физичка својства података се сматра однос између обраде податковних елемената и физичке локације података, где су они заправо смештени. Како већина обрађених података у „облаку“, остаје сачувана на „облаку“, а не на класичном серверу онда је потребно извесно време да се ти податци пребаце на сервер и обраде. Физичка својства су регулисана простом једначином (слика 12) која описује колико времена је потребно да се податци преместе са места одакле су направљени , сачувани , обрађени и архивисани . „Облаци“ су добри за чување података, него нису баш идеални за трајно архивирање и уништавање истих са унапред предвиђеним роком. Велике количине података , или такозвана „уска грла“ продужавају време које је потребно за пребацивање података са једног места на друго.

$$\text{време} = \frac{\text{бајт} \cdot 8}{\text{пропустни опсег}}$$

Слика 12. Основна једначина (Sun Microsystems)

Ова једначина је релевантна и за тренутну обраду података како и за дугорочно планирање . Она нам може помоћи да унапред предвидимо да ли има смисла, на пример имплементирати нову компјутерску стратегију где ће можда више времена бити потребно да пребацимо податке на јавни „облак“ него што би нам времена требало да исте обрадимо. Такође нам може помоћи при одређивању трошкова операције померања са једног провајдера'- даваоца услуге на други, како су наравно податци смештени код једног , морају бити пребачени код другог а тај процес може потрајати .

Трошкови премештања података се могу изразити и преко времена као на слици 11, али и преко пропустног опсега. Хибридни модел који је илустрован на слици 7, где постоји колокација приватног „облака“ једне компаније са јавним „облаком“ провајдера, може доста помоћи да се трошкови смање значајно. Пропустни опсег у оквиру колокацијског објекта је генерално широк и бесплатан, што ову стратегију чини веома успешном.

Ова својства су ту да опишу и размотре однос између обраде података и премештања истих из складишта до места где се обрађују може коштати доста времена а и новца. Ево пар кључних аспеката које требамо размотрити:

- Податци ускладиштени без компјутерске снаге у близини имају ограничену вредност, и провајдери услуга у „облаку“ би требало да буду транспарентни у погледу мрежног односа ових двеју компоненти. Која је величина жица? Колико је кашњење? Каква је поузданост везе? Услуга у „облаку“ би требала долазити заједно са одговорима на ова питања.
- Архитекте-ствараоци система у „облаку“ треба да буду у могућности да одреде локацију вирелних компоненти и услуга, тако што постоји добро дефинисан однос између виртуелних машина и места којима оне приступају.
- Провајдери услуга у облаку могу да оптимизују овај однос аутоматски за потрошаче, али размотрити претходно да ли би та примена њима била корисна.
- У мрежном окружењу, понекада је корисније (брже, мање латентности) да се израчуна вредност него ту исту преузимати из мрежног репозиторијума. Размотрити и направити компромис између коришћења израчунавајућих циклуса и премештања података.

4. Безбедност услужног рачунарства

Безбедност услужног рачунарства (често названо као „безбедност у облаку“) , је у развоју као под домен компјутерске сигурности, безбедности рачунарских мрежа , и шире гледано , информационе безбедности. То се односи на широк скуп дозвола , технологија и контрола распоређених за заштиту података, апликација као и . неопходне инфраструктуре за услужно рачунарство.

Сигурност у „облаку“ не треба мешати са понундама безбедносних софтвера који су засновани на „облаку“ као основи(такође модел безбедност као услуга).

Постоји низ безбедносних питања у вези са услужним рачунарством , али ова питања спадају у две категорије:

- Безбедностна питања која се тичу организација која омогућавају услуге засноване софтверској платформи
- Безбедностна питања која се тичу провајдера и организација које омогућавају услуге засноване на инфраструктури и безбедностним питањима са којима се суочавају њихови клијенти

У већини случајева , провајдер услуга мора да буде сигуран у томе да његова инфраструктура је безбедна и да његови клијенти морају бити сигурни да је он преузео одговарајуће безбедносне мере и циљу заштите њихових информација и података.

Широка употреба виртуелизације у функционисању инфраструктуре услужног рачунарства доноси јединствене сигурносне проблеме за клијенте јавног рачунарског „облака“.

Виртуелизација мење однос између оперативног система и основног хардвера, било да се ради о рачунању, чувању или чак умрежавању. То представља додатни слој-виртуелизацију који сам по себи мора бити исправно конфигурисан, управљан и наравно осигуран.

Посебни проблеми доводе могу да доведу у питање потенцијал софтвера за виртуелизацију или „хипервизора“. Иако су ови проблеми углавном теоријски, они постоје.

Безбедносне области услужног рачунарства

Одговарајуће безбедностне контроле треба спровести у складу са имовином, претњама и рањивошћу за процену матричне процене ризика. Док безбедностни проблеми услужног рачунарства се могу груписати у било који број димензија, док Безбедносна Алијанса (eng. Cloud Security Alliance) идентификује чак четрнаест проблематичних области. Ове области су груписане у три опште врсте:

- **Безбедност и приватност**
- **Усаглашеност**
- **Правна питања и интелектуална својина**

У даљем делу текста бавимо се безбедношћу и приватношћу а ово су кључне особине :

- **Управљање корисницима и контрола приступа:**

Свака фирма односно предузеће има свој систем за управљање и контролу приступа корисника тј. запослених до одређених информација и компјутерских ресурса. Провајдери услуга или интегришу систем за контролу приступа у своју инфраструктуру наравно у складу са законом и правним прописима, или обезбеђују друга решења за управљање корисницима.

- **Физичка безбедност опреме и корисника:**

Провајдери услуга морају обезбедити приступ до физичких машина тј. сервера и да осигурају тај приступ , као и да све релевантне податке купаца не само заштити већ и документује.

- **Доступност**

Даваоц услуге осигурава клијенте у смислу да ће они имати редован и несметан приступ својим податцима и апликацијама.

- **Примена безбедности**

Даваоц услуге гарантује да ће апликације које се дистрибуирају као сервис преко облака су безбедне применом процедура за потребе процеса аутсорсинга или апликативног кода који је спакован. То такође захтева примену мере безбедности у развојном-проиводном окружењу.

- **Приватност**

Коначно даваоци услуга се брину да сви поверљиви податци (бројеви кредитних картица на пример) су сакривени и само је присуп таквим податцима омогућен ауторизованим корисницима. Свакако, дигитални идентитети и референце морају бити заштићене како и и сви потаци које провајдер скупља о корисничким активностима унутар „облака“.

Сигурносна својства података

Податци су често највреднија имовина привредног друштва, и они морају бити заштићени са већом пажњом у односу на било који други део фирме. Лако се може закључити зашто су податци тако вредни и зашто се тој теми баш посвећује пажња. Потенцијални уљез може украсти податке са било ког места на планети користећи интернет. Ево и пар корака како избећи такво нешто:

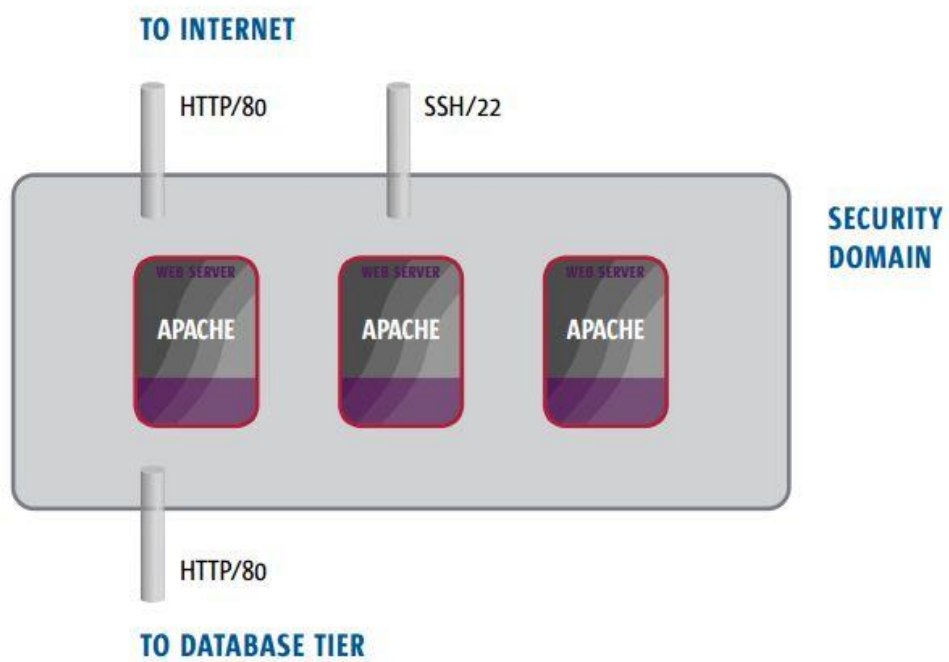
- Шифровати податке у **REST**-у (енг. *Representational State Transfer*), то је метод који је широм коришћен на мрежи, којим се постиже да уљез који проба да изврши улаз у систем, или било каква грешка у конфигурацији која би угрозила приватност података, чини аутоматски податке недоступним на одређено време.
- Шифровање података у саобраћају. Претпоставимо да ће податци свакако проћи јавну инфраструктуру и може се сматрати да тамо могу бити видљиви од било које стране која је између.
- Захтевати јаку аутентификацију између апликативних компоненти, тако да су подаци преносе само познатим лицима.
- Обратити пажњу на криптографију и како алгоритми који су превазиђени мењају новијим током времена. На пример сада је доказано да је MD5 алгоритам подложен нападима, користити јачу технику као што је SHA-256.
- Управљати ко има приступ пријављивању и како :
 - Размислити о коришћењу јаке аутентикације на бази знакова, за административне улоге
 - Када се корисник пријави у систем, размотрити ко управља провером да ли је то фирма или провајдер услуга у „облаку“.
 - При анонимном приступу за чување података, на пример анонимни FTP, размислити о томе да ли ће купац услуге морати да се региструје да би даље имао приступ или да услуге у облаку могу да буду спојене са провером на серверу фирме која даје услуге у „облаку“

Мрежна безбедност у практичној примени

Добра безбедностна пракса прожима скоро сваки аспект системског дизајна , имплементације и распоређивања. Пријаве морају бити сигурне дизајном, са корисничким окружењем које омогућава приказ само појединих података овлашћеним корисницима. Током имплементације система , програмери морају водити рачуна о томе да се избегне пракса шифровања која би могла резултирати рањивошћу као што је преоптерећење бафера (eng. buffer overflow) или SQL инјекција (eng. SQL injection). Када је имплементиран , оперативни систем треба да очврсне и сваки слој софтвера мора се редовно ажурирати најновијим безбедностним закрпама.

У услужном рачунарству, апликације су распоређене у дељеном мрежном окружењу, и много напредне безбедностне технике се користе као што је филтрирање VLAN а и портова се користи за поделе и заштите различите слојеве архитектуре , како и да одвоје кориснике сваког појединачно. Неки приступи мрежној безбедности укључују:

- Коришћење безбедносних домена са циљем груписања виртуелних машина заједно, и онда контролисањем приступа на мрежи преко могућности филтрирања портова коју провајдер има. На пример, креирање сигурносне мрежу за крајње интернет сервере , затим отварање HTTP или HTTPS портова ка спољњој мрежи и филтрирање саобраћаја од интернет доменских сервера на једну од база података која је садржана као на слици 13.
- Контролисање саобраћаја помоћу провајдеровог филтрирања портова, или наменским пакетом који користи прекидаче(eng. switch), а може се користити заштитни зид (eng. firewall) где је то прикладно.



Слика 13. Мере безбедности при коришћењу безбедностних домена (Sun Microsystems)

5. Практична примена услужног рачунарства

Концепт рачунарства у „облаку“ (eng. cloud computing) је релативно нов , мада услуге које се нуде у облаку (cloud) нису нове . На пример Windows Live Hotmail , један од најпопуларнијих сервиса размене порука , покренут је 1996 те године и данас опслужује око 400 милиона корисника. Оно што је ново у cloud окружењу је све већи број сервиса и алтернативних модела пружања услуга који омогућавају уштеде у трошковима , скалабилност , приступачност , лаку употребу , квалитет, флексибилност и коришћење најновијих технологија. Услужно рачунарство обезбеђује брже реаговање информационих технологија на промене у окружењу, олакшава управљање ризицима, омогућава дугорочније планирање потреба и брже и квалитетније опслуживање захтева клијената .

У овом поглављу се бавимо функционалним прегледом популарних cloud платформи које су данас у широкој употреби. Детаљно су описани технички аспекти свих појединачно и направљено је групно поређење , а такође се говори и о пословним проблемима .

Као увод у ово поређење , платформе смо поделили на основу нивоа апстракције, да ли се ту ради о инфраструктури (IaaS) или платформи (PaaS) као сервису , а можда и комбинацији оба модела .

- Инстанце које раде на хардверском моделу (IaaS) , и пружају их *Amazon*, *Grip* и *Mosso servers* . Тиме је само неколико апликационих програмских окружења (eng. API) се налази на врху за постављање инстанце. То омогућава покретну примену , и уобичајене релационе базе података , која због постојећих проблема са скалабилношћу, имају ограничен потенцијал за еластичност. Алтернативно *Amazon* нуди просте базе података за изградњу скалабилних програма за складиштење података .
- Хибридни модел (комбинација IaaS и PaaS) јавног облака , као што је *Microsoft Azure* чиме се постиже да се више апликационих програмских

окружења (eng. API) врати изворном скалабилном окружењу. Међутим оно шта се догађа испод површине углавном је непознаница (у *Microsoft* овом случају). Користећи *Microsoft Azure* платформу , програмери су у стању да планирају еластичност функција у својим апликацијама (користећи уграђене функције где је то могуће) , и дозвољавају неким степеном аутоматизације опоравак од кvara / скалабилну оптимизацију када се користи пријава на центар са подацима / „облак“. NET примене је још увек могуће извршити , мада не користи у потпуности могућности своје платформе (неке апликације могу чак да раде спорије).

- Услуге у облаку засноване на (PaaS) моделу , као што су *Google AppEngine* и *SalesForce/Force.com* су заштићене , и доменски специфични PaaS/SaaS модела (SaaS јер се испоручује као интернетска апликација) који се могу произвољно програмирати до одређеног нивоа и за одређени пословни задатак . *Google* (као и *Amazon*) је делове својих сопствених алгоритама за изузетно флексибилно рачунарство и складиштење учинио јавно доступним, док платформа за пословни развој *Force.com* дозвољава програмерима да коришћењем *Apexa* (за оне који су мање вични програмирању постоји витулени GUI *Visualforce*) просто испрограмирају причагодљиве екстензије својим главним SaaS апликацијама.

Генерално гледано када се упоређују платформе , препоручује се да се могућности складиштења и могућности рачунара посматрају одвојено . Истраживања која су спроведена показују да већина материјала који се користи за маркетинг, као и за опис cloud производа не стављају акценат на разлику између складиштења и облака који ће се користити за разне прорачуне . Чак и неки научни радови једва да се дотичу теме , а ипак ова разлика је веома важна за компанију која треба да пребаци свој информациони сектор на „облак“ . Уколико се менаџмент неке компаније пребаци на „облак“ не треба да раздвоје своје потребе за простором за складиштење од

рачунарских потреба у циљу одабира правог решења . Табела на слици 14 показује неке техничке и функционалне карактеристике неких провајдера – давалаца услуга .

	Amazon Web Services	Mosso Cloud Servers	Windows Azure Services	Google AppEngine	Sales-force.com, Force.com
(X)-a-a-S Type	IaaS	IaaS	IaaS/PaaS	PaaS	PaaS/SaaS
Application runs in a visible VM	X	X	X	-	-
Provides administrative access to VM	X	X	-	-	-
Can create own OS installations	X	-	-	-	-
Service Type	Compute, Storage (S3)	Compute, Storage	Compute, Storage, Framework	Compute, Web Application	Web Application
Virtualization	OS Level running XEN	OS Level running XEN	OS Level	Application Container	Application Container
Programming Framework	Linux-based Amazon Machine Image (AMI)	N/A, TBA	MS .NET Languages	Python or Java	Apex (script) or Visualforce (GUI)
Runs normal on-premises applications	X	X	-	-	-
Suitable for creating moderately scalable applications	X	X	X	-	-
Suitable for creating very scalable applications	X	-	X	X	X
Suitable for running parallel processing applications	X	-	X	-	-

Слика 14. поређење техничких и функционалних карактеристика провајдера

Amazon Web Services

Amazon Web Services (скраћено **AWS**) је скуп више удаљених рачунарских услуга , широм света познатији као интернет сервиси , које заједно чине платформу услужног рачунарства коју Amazon нуди путем интернета . Најпозантије од тих услуга су Amazon EC2 и Amazon S3 .

Amazon Elastic Compute Cloud (EC2) је централни део cloud платформе коју нуди Amazon . EC2 допушта корисницима да изнајме виртуални рачунар на којима извршавају своје апликације на рачунару . EC2 дозвољава прилагодљиву примену апликација коју обезбеђује интернет услуга , преко које корисник може да покрене неку слику виртуалне машине (енг. Amazon Machine Image) са циљем да направи виртуелну машину , коју Amazon назива инстанцом , која може да садржи софтвер за који се корисник определи . Корисник може да креира , покреће , гаси инстанце колико је год то њему потребно , плаћајући по сату за активне сервере , па одатле потиче термин „еластична услуга“ . EC2 нуди корисницима контролу над географском локацијом која омогућава оптимизацију због кашњења и високог степена редундантности .

Elastic Compute Unit (ECU) је представљена од стране Amazon EC2 као апстракција рачунарских ресурса . Amazon – ова дефиниција ECU а наводи „Ми користимо стандарде и тестове за управљање конзистентности и предвидљивост у вршењу EC2 јединице“ . Једна EC2 компјутерска јединица даје рачунарску снагу еквивалента процесора из 2007 ме 1,0 – 1,2 GHz AMD Opteron или Intel Xeon процесора .

Amazon S3 (eng. Simple Storage Service) је мрежна услуга за складиштење коју нуди Amazon . Amazon S3 омогућава складиштење преко мрежних интерфејса (REST - Representational State Transfer , SOAP - Simple Object Access Protocol) . Amazon S3 је први пут јавно покренут у Сједињеним Америчким Државама 2006. године , а у Европи у новембру 2007.године . Детаљи дизајна S3 нису јавно објављени . Он има за циљ да обезбеди проширивост и високу доступност , а при том ниску латенцију у робним трошковима . Дизајниран је да пружи 99% трајности и 99,99% доступности објеката у задатој години . Amazon S3 смешта произвољне фајлове величине до 5 терабајта у величини , од којих је сваком придружено по 2 килобајта метаподатака . Објекти су

организовани у сегменте (од којих је сваки у власништву Amazon Web Services - а) , и сваки сегмент има своју појединачну идентификацију . Amazon Machine Images које су модификоване у Elastic Compute Cloud – у могу бити извезене као снопови на S3 . Снопови и објекти могу бити креирани , листани , и преузети користећи REST - стил HTTP интерфејса или SOAP интерфејс . Захтеви су овлашћени коришћењем приступне контролне листе (енг. access control list) , која је повезана са сваким сегментом и објектом . Имена сегмената и кључева су таква да се лако могу адресирати преко интернет адресе :

- <http://s3.amazonaws.com/bucket/key>
- <http://bucket.s3.amazonaws.com/key>
- <http://bucket/key> (где је сегмент DNS запис који показује на `bucket.s3.amazonaws.com`)

Пошто су објекти доступни од неизмењених HTTP клијента , Amazon S3 може бити употребљен да замени постојећу инфраструктуру. Amazon AWS механизам за аутентикацију омогућава власницима сегмената да креирају овлашћену интернет адресу са ограниченим роком трајања . То је неко ко може направити интернет адресу и предати трећој страни на приступ за период од наредних 30 минута , или наредних 24 часова .

Dropbox, Zmanda и Ubuntu One су многе од бројних компанија за прављење резервних копија и синхронизацију података , које користе S3 као њихово складиште за смештај и пренос података .

Google App Engine

Google App Engine (често ословљаван са GAE или просто App Engine , а такође се користи акроним GAE/J) је платформа која се заснива на сервису (PaaS) , и користи се за развој и примену интернет апликација на Google управљачким центрима података . Пријаве су заштићене (енг . Sandbox) и покренуте на више сервера . App Engine нуди аутоматско скалирање интернет апликација , како се њихов број повећава . App Engine издваја више средстава за интернет апликацију да би се справио са све већом потражњом .

Google App Engine је бесплатан до одређеног нивоа утрошених ресурса . Накнада се наплаћује за додатни простор за смештање података , већи пропустни опсег или часова колико је потребно за израду апликације . Први пут је издат као верзија за преглед у априлу 2008.године , да би био званично пуштен у рад септембра 2011. године .

Када посетимо интернет адресу App Engine а <http://code.google.com/appengine> примећујемо пар карактеристика :

- Није потребна додатна опрема.
- Google App Engine излаже потпуно интегрисано развојно окружење.
- Лак за скалирање.
- Бесплатан је за почетак

Наведене карактеристике сажимају Google ову стратегију за PaaS. Google не покушава да помогне својим клијентима да њихов апликације буду покренуте на свакој платформи . Он је уско фокусиран да помогне клијентима да изграде темељну интернет апликацију. Платформа је намењена за развој и имплементацију

У суштини програмер пружа Google App Engine у интернет адресу (URL) за апликацију коју креира и основа аутоматски мапира тај код на Google ову развојну платформу. App Engine рукује циклусом развоја интернет апликација, укључујући

рутинске задатке као што су листе захтева , провера статуса пријаве , ажурирање верзије софтвера , вођење базе података, руковање током посла.

Google је интегрисао сва развојна окружења у једно окружење. Када клијенти вежу своја развојна окружења у циклусно окружење које нуди Google, добијају приступ Google овом IaaS у . На овај начин клијенти могу да додају већи капацитет на захтев .

Као додатак развоју услуга Google исто тако обезбеђује и друге интегрисане платформе , укључујући али не ограничавајући се на следеће :

- Google налози за проверу идентитета
- Google изворни фајл под називом GFS (Google File System)
- BigTable платформа (за управљање податцима), дистрибуисани систем за складиштење који управља веома великим структурним податцима

Она такође укључује и инфраструктурне услуге, као што су :

- Балансирање оптерећења
- Трајно складиштење са са упитима (енг. queries)
- Сортирање и трансакције
- Програмски интерфејс за подршку аутентикације корисника и слање поште помоћу Google налога
- Планирање задатака за покретање догађаја у задатом времену

То је суштински иста платформа коју Google користи за развој свог софтвера

Поред ових инфраструктурних алата , Google App Engine такође укључује развојни стек. Google зове назива ово скалабилном порцијом која спаја интернет апликације са Google овим окружењем.

То се постиже интеграцијом следећих алата :

- Python runtime: за креирање апликације неопходан је пре свега програмски језик. Један од првих који је Google подржавао био је Python, то је програмски језик на високом нивоу (eng. high level) који омогућава лак и брз развој сложених апликација са минималним програмирањем. Python обухвата моделе , пакете и подржава поновну употребу кода.
- Java runtime: Google је додао Java као други подржани програмски језик . Овај преводилац је интегрисан са Google овим алатима и намењен је да се користи са AJAX - ом (asynchronous JavaScript and XML) или интерактивне интернет апликације.
- Развојни софтверски алат (енг. Software Development Kit - (SDK)) . Овај скуп развојних алата омогућава програмерима да пишу код апликације.
- Административна конзола заснована на интернету . Ова конзола помаже програмерима да управљају својим апликацијама .
- Центар са подацима . Центар са подацима је слој софтвера који смешта податке у интернет апликације . Изграђен је на BigTable платформи

.

Microsoft Azure

Microsoft, који је актуелан у развоју апликација и пласирањем истих на тржиште одлучује се да пласира своју .Net платформу у облак. Azure је Microsoft – ова PaaS стратегија која је објављена 2008. године . Microsoft дефинише Azure платформу као „услуга у облаку интернет домена коју хостује Microsoft у својим центрима са подацима, који пружа оперативни систем и скуп развојних услуга које се могу користити заједно или појединачно“ .

Microsoft-ова општа стратегија је комбинација локалног рачунарства са услужним . Идеја је да програмери који желе да граде апликације које постоје на неком месту, док се остале компоненте налазе на „облаку“. Microsoft назива ову стратегију „software plus services“ . Центар стратегије Microsoft – а је да омогући програмерима да користе њихову технологију која се заснива на истим принципима и за локалне и апликације у „облаку“.

Док је главни фокус Azure технологије да познати Microsoft-ов модел програмирања у „облак“, Microsoft такође намерава да подржи друге програмске језике и моделе , укључујући Ruby on Rails и Python. За интероперабилност (способност за заједнички рад различитих система) , Microsoft подржава разне интернет протоколе , укључујући HTTP, REST, SOAP, и XML .

Azure платформа се састоји од следећих компоненти:

- Windows Azure : ова апликација покреће Windows окружење и локално чува одговарајуће податке о серверима унутар Microsoft центра са подацима.
- Microsoft .NET услуге : то су исте услуге које корисник има у Windows окружењу . Он је проширен тако да подржи апликације засноване на „облаку“ као и на локалу.
- Microsoft SQL Services : ове услуге база података Microsoft нуди на локалу само што су проширене на „облак“. Међутим ако желите да искористите способности које Azure нуди за скалабилност, морате преписати SQL код . Циљ SQL услуга је да обезбеди средства за чување

података у „облаку“. За услуге преноса података се користе оба REST и SOAP интерфејса као методе за приступ подацима. За складиштење података Azure не користи релациони модел.

- Live Services : Овај сет услуга омогућава програмерима да се повежу код развијен у Windows Live платформи са „облаком“. Ови сервиси обухватају оквир у коме треба да интегришу , повежу и синхронизују код. Платформа такође укључује подршку за различите програмске језике и омогућава да сви ресурси садрже интернет адресу.

Када саберемо све аспекте Microsoft је креирао платформу која омогућава програмерима да користе познате алатке као што су Microsoft's Visual Studio и .Net сервиси , а затим да сместе податке у њихову платформу услужног рачунарства. Због тога програмери који су упознати са развојем .Net апликација неће бити проблем да се прилагоде услугама у „облаку“.

Као и Google, Microsoft има исти модел наплате услуга по принципу плати колико потрошиш за своје PaaS услуге. И као и Google, Microsoft не наплаћује фазу развоја и тестирања. Почиње са наплатом клијентима када су апликације распоређене.

6. Закључак

Модел рачунарства који је кренуо као скуп гласина, неодобравања, потврда, исмевања и узвика подршке, дефинитивно је имао буран почетак. Неки су тврдили да је то већ виђена ствар, упакована у другачију амбалажу која се маркетингом продаје људима који желе нешто ново. Истина је да је услужно рачунарство скуп многих претходних модела и технологија које су развијане још од половине прошлог века, али није истина да је већ виђен у модерном информатичком друштву.

Основа овог модела је приступ чија је главна одлика испорука различитих типова ресурса на неконвенционалан начин, у виду услуге. Ти ресурси се огледају у хардверу, софтверу, одржавању, итд. На овај начин, крајњи корисници ослобођени су непотребних брига које им одвлаче пажњу од њиховог примарног посла или делатности.

Ово је модел о коме ће бити још доста приче, али је чињеница да све више људи улази у филозофију обављања разних типова послова у облаку. Не само што људи све више прихватају овај модел пословања, већ се и све више новца улаже у услуге у облаку, а многи истраживачи и визионари му предвиђају светлу будућност.

7. Литература

1. Weiss, Aaron. Computing in the clouds. *netWorker* 11, 4 (Dec. 2007),
2. http://en.wikipedia.org/wiki/Cloud_computing
3. <http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031>
4. Pollette, Chris. "How the Google-Apple Cloud Computer Will Work." *Howstuffworks.com*
5. Introduction to Cloud Computing Architecture ,Sun Microsystems June 2009
6. J. Barr, „Host your web site in the cloud, Amazon web services made easy“, Amazon Web Services, 2010.
7. M. Chu-Carroll, „Code in the Cloud Programming Google App Engine“, Pragmatic Programmers, LLC, 2011.
8. S. Fiore, G. Aloisio, „Grid and Cloud Database Management“, Springer-Verlag Berlin Heidelberg, 2011