

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Virtualizacijos technologijų pritaikymas debesyje (Cloud)
Virtualization in the Cloud

Magistro baigiamasis darbas

Atliko:	Jonas Mardosas	(parašas)
Darbo vadovas:	doc.dr. Saulius Minkevičius	(parašas)
Recenzentas:	doc.dr. Andrius Kurtinaitis	(parašas)

Vilnius – 2010

Turinys

Santrauka	6
Summary.....	7
Įvadas.....	8
1. Debesų kompiuterijos sąvoka.....	10
1.1 Debesų kompiuterijos charakteristikos.....	11
2. Debesies paslaugų lygmenys.....	13
2.1 Infrastruktūra, kaip paslauga	14
2.1.1 Programinė įranga bei kompanijos teikiančios infrastruktūros paslaugas.....	14
2.2 Platforma, kaip paslauga.....	16
2.3 Programinė įranga, kaip paslauga.....	16
3. Debesų kompiuterijos dislokacijos modeliai.....	18
3.1 Privatus debesis (angl. Private cloud)	18
3.2 Bendras debesis (angl. Community cloud).....	18
3.3 Viešas debesis (angl. Public cloud).....	19
3.4 Hybridinis debesis (angl. Hybrid cloud)	19
4. Virtualizacija debesyje	19
4.1 Virtualizacija	20
4.2 Virtualizacijos tipai.....	20
4.3 Virtuali mašina	22
4.4 Virtualūs įrenginiai	22
4.4.1 Virtualus procesorius	22
4.4.2 Virtuali atmintis	23

4.4.3	Virtualūs įvesties/išvesties įrenginiai.....	23
4.5	Virtuali infrastruktūra.....	24
4.5.1	Konsolidacija ir izoliacija.....	24
4.5.2	Įsibrovimų aptikimo sistema.....	24
4.5.3	Virtualių mašinų migracija.....	25
4.5.4	Aukštas patikimumas (High Availability).....	26
4.5.5	Dinaminis resursų paskirstymas.....	26
4.5.6	Energijos vartojimo valdymas.....	27
5.	Virtualizacijos produktai.....	27
5.1	VMware virtualizacija.....	27
5.2	Xen virtualizacija.....	28
6.	Virtualūs debesies resursai.....	28
7.	OpenNebula platforma.....	28
8.	Eucalyptus platforma.....	30
9.	Paketų palyginimas.....	31
10.	Eucalyptus paketas.....	32
10.1	Eucalyptus palaikomos virtualizacijos technologijos.....	32
10.2	Eucalyptus architektūra.....	32
10.3	Cloud Controller.....	34
10.4	Eucalyptus paslaugos.....	35
10.4.2	Skaičiavimo resursai.....	35
10.4.3	Storage Controller (walrus).....	36
10.4.4	Storage Controller (Block Storage).....	38
10.5	Eucalyptus duomenų tinklo infrastruktūra.....	39
11.	Debesų kompiuterijos sąsaja (angl. Cloud API).....	40

11.1	Eucalyptus (Amazon) API.....	41
11.2	Eucalyptus sistemos valdymo įrankiai (euca2tools).....	42
12.	Paslauga debesyje	43
13.	LAMP paslauga	45
13.1	Internetinių puslapių talpinimo paslauga.....	46
13.2	LAMP virtuali mašina	47
14.	Paslaugos duomenys.....	47
14.1	Servisų konfigūraciniai duomenys	48
14.2	Aplikacijų duomenys.....	49
14.3	MySQL duomenų bazė	49
14.4	Papildomi duomenys	50
15.	Paslaugos stebėjimas	50
16.	Sistemos valdymo įrankis.....	51
16.1	Sistemos valdymo modulis.....	51
16.2	Sistemos valdymo modulio funkcijos	52
16.2.1	Tinklalapio serverių valdymas.....	52
16.2.2	Objektinės talpos valdymas (walrus storage)	53
16.2.3	Virtualių diskų valdymas	53
16.3	Sistemos valdymas	54
16.4	Apskaita (angl. accounting).....	56
17.	Paslaugos perkėlimas į debesį	57
18.	Dinaminis plečiamumas pagal poreikį	59
18.1	Pagal apkrovimą	61
18.2	Pagal programinę įrangą.....	62
19.	Bendras PaaS paslaugos sukūrimo modelis.....	62

Rezultai ir išvados	64
Šaltiniai.....	65
Priedai.....	67
1) LAMP funkcijų paruošimas virtualios mašinos vaizde.	67
2) Pavyzdinis skriptas (Apache serviso konfigūracija, bei tinklapio katalogas).	68
3) Munin grafikai: tinklo apkrovimas, atminties naudojimas, CPU apkrovimas.	69

Santrauka

Šiame darbe aprašomos technologijos naudojamos debesų kompiuterijos platformose. Pilnai išanalizuojama nemokama debesies platforma Eucalyptus. Bandoma sukurti internetinių puslapių talpinimo paslaugą debesyje (PaaS paslauga), kuria naudotis galėtų daug vartotojų. Taip pat sudaromas planas kaip galėtų atrodyti panašių paslaugų perkėlimas į debesies infrastruktūras. Išnagrinėjus, kokios programinės įrangos reikia tokiai paslaugai teikti, paruošti pavyzdiniai instaliaciniai skriptai, nubraižytos schemos kaip tokia paslauga galėtų veikti ir kokias funkcijas, bei kokią naudą gauna galutinis vartotojas naudodamas tokią paslaugą. Suprojektuota sistema, kuri automatiškai turi rūpintis tokios paslaugos valdymu, bei stebėjimu. Pateikti tokios automatizuotos sistemos kodo pavyzdžiai.

Summary

This document describes the technologies used in cloud computing platforms. Also this work completely analyze cloud open free platform Eucalyptus. On this platform trying to create a web page hosting service in the cloud as a PaaS service, which could be used of many users. Also work describes the plan/scheme as it might be possible to transfer similiar services to the cloud infrastructure. Examination of which software must be provided the following services, preparing model system installation scripts, either as a scheme for such a service can operate and what functions and what benefits the final consumer gets using this service. Designed a system that automatically can provide such a service management and monitoring. Shows such an automated system code examples.

Įvadas

Tam, kad gerai ir suprantamai išnagrinėti darbo temą reikia suprasti naujus terminus, kurie vartojami debesų kompiuterijos srityje. Debesų kompiuterija (angl. Cloud Computing) apibrėžimas atsiradęs visai neseniai. Šiuo metu tai greičiausiai auganti IT sritis. Šį terminą bandoma aiškinti įvairiais būdais, kol kas nėra jokio standarto, kuris pasakytų, kaip teisingai apibrėžti šį terminą. Atsiranda vis daugiau nuomonių ir apibrėžimų debesų kompiuterijos tema.

Paslaugos debesyje dabar ypač populiarios, nes už jas vartotojai moka, tiek kiek jas naudoja. Taip pat lengvai išsprendžiamas naujų resursų praplėtimo klausimas. Tokios paslaugos kaip internetinių puslapių talpinimas ar panašios yra gan paplitusios. Tačiau jas perkelti į debesį nėra lengva, nes kiekviena debesies platforma turi savo įrankius, bei technologijas.

Dauguma debesies paslaugos komercinių tiekėjų naudoja savo programinę įrangą ir neatskleidžia kaip realizuota debesies paslauga. Cloud rinkoje yra ir nemokamų debesies platforminių produktų. Vienas iš jų, - Eucalyptus sistema, dažniausiai naudojama testavimo, mokslinio tyrinėjimo tikslais. Šis produktas nėra pilnai išdirbtas ir vis dar tobulinamas.

Vienas šio produkto privalumų tai, kad jis turi tokį pačią sąsają API (angl. Application programming interface) kaip ir Amazon komercinė Cloud paslauga, kuri šiuo metu yra labiausiai paplitusi tarp debesies IaaS (žr. 2.1) paslaugų.

Tačiau pati Eucalyptus sistema, turi tik branduolį, kuris reikalingas, kad suteikti pagrindines debesies IaaS funkcijas/paslaugas. Neturėdama jokių patogių įrankių skirtų palengvinti darbą IaaS paslaugų vartotojams, ši sistema gali pasirodyti sudėtinga valdyti ir naudotis – priešingai Amazon atžvilgiu, kur vartotojai turi daugybę patogių įrankių skirtų suvaldyti cloud paslaugas, tačiau tie įrankiai nėra platinami viešai.

Palyginus debesies produktus: Nebula ir Eucalyptus, buvo pasirinkta Eucalyptus platforma. Atliktas jos įdiegimas keliuose fiziniuose serveriuose. Skaitant Eucalyptus, taip pat ir Amazon paslaugų dokumentaciją ir išbandžius įvairius įrankius su realiai veikiančia sistema – aprašyta Eucalyptus sistemos architektūra.

Gerai susipažinus su eucalyptus platforma, nuspręsta paruošti PaaS (žr. 2.1.2) paslaugą debesyje, naudojantis eucalyptus teikiamamos IaaS paslaugomis. Pavyzdžiui pasirinkta internetinių puslapių talpinimo paslauga (angl. web hosting), kuri yra PaaS lygmens paslauga debesyje.

Magistro darbo tikslas, - išsiaiškinti, kaip paprasta internetinė paslauga gali būti perkelta į debesies platforma ir joje dirbti kaip PaaS paslauga, t.y. kaip jos analogas.

Tuo pačiu norint pasiekti šį tikslą, realiai įdiegti sistemą ir ją išbandyti, pritaikyti darbui, sukurti valdymo sistemą, kuri valdytų PaaS paslaugos funkcijas. Kad tai įgyvendinti, darbe atliekama detali Eucalyptus sistemos analizė, nustatoma sistemos komponentų paskirtis, išbandomos php bibliotekos, nustatoma kokia programinė įranga reikalinga tokiai PaaS paslaugai. Remiamasi Eucalyptus dokumentacija, bei realiu sistemos darbu dviejuose ITC išskirtuose fiziniuose tarnybinėse stotyse.

Toks tikslas iškeltas, norint išbandyti ir parodyti, kad debesyje tokios paslaugos tampa lankstesnės, patogesnės vartotojui, bei ekonomiškesnės tiek iš vartotojo pusės, tiek iš PaaS paslaugos tiekėjo pusės.

1. Debesų kompiuterijos sąvoka

Norint suprasti kas yra Debesų kompiuterija - debesu kompiuterija, reikia apibrėžti debesies (angl. Cloud) terminą. Išnagrinėti, kodėl atsirado toks terminas. Reikia akcentuoti taip pat ir finansinę naudą, kuri gaunama naudojantis debesies paslauga. Tokia paslauga dažniausiai yra tiekama iš taip vadinamų vidutinio arba didžiulių duomenų centrų (angl. medium-size, big-size datacenters).

„Debesų kompiuterija“ turi tokias savybes:

- Techninės įrangos, skaičiavimus atliekančių resursų ar taikomųjų programų suabstraktinimas;
- moki tik už tai ką naudoji, tik tol kol naudoji, jokių ilgalaikių investicijų;
- pilna paslaugų sukūrimo, plėtimo ar išjungimo automatizacija;
- galimybė greitai plėstis atsiradus naujiems poreikiams, bei lanksčiai sumažinti resursus ir mokėti mažiau;
- išnaudojamas resursų paskirstymo efektas, kai vienu klientų sumažėjusį resursų poreikį gali panaudoti kiti klientai;
- self-healing - sistema pati sureaguoja į tam tikrų komponentų gedimą ir perkelia resursus į kitą vietą;

Iš tokių savybių galime spręsti, jog „Debesų kompiuterija“ yra orientuota į paslaugas, arba kitaip tariant skirtas teikti paslaugas internetu.

Pirmiausia, norint suprasti debesies pagrindinius principus, reikia suprasti kaip šis terminas apibrėžiamas technologiškai. Architektūriškai ir terminologiškai debesų kompiuterija apibrėžiama kaip debesies (Cloud). Taip yra todėl, kad debesies terminas savyje talpina daugybę technologinių apibrėžimų, pvz: grid skaičiavimai, resursų paskirstymo įrankiai, virtualizacija, SOA, WEB 2.0 ir t.t. Tai reiškia, kad tos technologijos iš dalies, arba pilnai yra debesyje. Cloud gali būti visų šių technologijų bendras pavadinimas. Daugybė technologinių apibrėžimų, sistematikų ir architektūrų gali skambėti sudėtingai. Svarbiausia suprasti kaip visos šios technologijos gali bendrai tarnauti Debesų kompiuterijos paslaugai.

Taip pat reikia išsinagrinėti debesies paslaugos tiekėjų veikimo principus.

Iš dalies nagrinėjant, Amazon EC2 (Amazon Elastic Compute Cloud) paslaugas, kurios orientuotos į tam tikrų skaičiavimo resursų tiekimą (virtualių resursų) ir Google AppEngine paslaugas, kurios orientuotos į programuotoją – aplikacijai veikti reikalingas aplinkos tiekimas, galima suprasti kas yra Debesų kompiuterija.

Debesų kompiuterija – lankstus ir dinamiškas įvairiomis technologijomis virtualizuotų resursų pateikimas vartotojui internetu. Pagrindinė šios technologijos idėja yra tai, jog vartotojui nėra būtina turėti daug žinių, kaip susikurti vienokią ar kitokią paslaugą. Vartotojui pakanka mokėti naudotis skaičiavimų debesies teikiamomis paslaugomis, visiškai nesigilinant, koku būdu ir kokiomis technologijomis jam rūpima paslauga pateikiama. Toks skaičiavimo telkinys (debesis) skirtas sujungti įvairias technologijas į vieną bendrą visumą, ir pateikti vienodą sąsają, visoms teikiamoms paslaugoms.

1.1 Debesų kompiuterijos charakteristikos

Pagrindinės debesies charakteristikos yra dinamiškai, taupiai, optimizuotai išdalinti ir pateikti skaičiavimo resursus subjektams, kuriems tuo momentu tie resursai reikalingi. Dažniausiai tie subjektai tai, - klientas, galutinis vartotojas, organizacija, IT padalinys, IT sistema. Visais išvardintais resursais subjektas turi turėti galimybę pasinaudoti neišmanydamas sudėtingų technologinių principų, esančių įvairiose platformose ir sistemose. Debesies architektūra pati savaime gali būti privati (esanti organizacijos ribose), arba vieša (prieinama internete). Taigi pagrindiniai Debesų kompiuterija principai galėtų būti tokie [TRGooC108]:

- **Galimybė pakeisti skaičiavimo išteklius pagal poreikį (Scalability on Demand)**

Viena iš didžiausių debesų kompiuterija privilegijų - praplėsti, sumažinti sistemų skaičiavimo išteklius. Jeigu organizacijoje yra tam tikras laiko periodas, kai reikalingi labai dideli arba labai maži skaičiavimo resursai negu įprasta, cloud paslauga-technologija gali išspręsti šią problemą.

• **Duomenų centro/IT infrastruktūros modernizavimas**

Kiekviena organizacija investuoja didelias pinigų sumas į savo IT infrastruktūras. Tai apima tiek techninės, tiek programinės įrangos pirkimą, modernizavimą, palaikymą. Palaikymas reikalauja žmogiškųjų išteklių. Organizacija perkeldama savo sistemas į vidinį arba išorinį – virtualų duomenų centrą (Cloud), gali sumažinti išlaidas visoms šioms išvardintoms veikloms. Paprasčiau tariant, perkama paslauga/resursai vietoj to, kad programos/sistemos dirbtų organizacijos infrastruktūroje. Tuo pačiu debesies paslaugos tiekėjas pasirūpina, kad vartotojo duomenys išliktų saugūs, bei būtų visada pasiekiami. Tam naudojama aukšto patikimumo įrangą, technologijos, kurios fiziškai laikomos saugiuose duomenų centruose[RHPCA99].

• **Verslo procesų gerinimas**

Debesis (Cloud) suteikia galimybę pagerinti verslo procesus organizacijoje. Organizacija ir jos tiekėjai arba partneriai, gali dalintis visa informacija ir sistemos teikiamomis galimybėmis viename debesyje (Cloud). Kiekvienas iš jų, gali susitelkti ties verslo procesu, vietoj to, kad prižiūrėtų kiekvieną sistemą/IT infrastruktūrą atskirai.

• **Mažesni startavimo kaštai**

Organizacijos, kurios tik pradeda veiklą, savo sistemas gali pradėti diegti debesyje. Jeigu organizacija pradėtų veiklą su savo infrastruktūra, tokiu atveju laikas ir kiti resursai būtų skirti naujos IT infrastruktūros kūrimui, jos valdymui ir t.t. Cloud atveju infrastruktūra rūpinasi Cloud paslaugos tiekėjas. Tas pats galioja ir didelėms organizacijoms, kurios turi skubius arba laikinus projektus, o tuo metu nėra pakankamai resursų. Organizacijai nereikia nuomotis tarnybinių stočių ar duomenų talpyklų laikiniems projektams. Viskas paimama iš Cloud, ir mokama tikrai už išnaudotą laiką.[MARBVIC07]

Taigi galima daryti išvadą, kad debesies vartotojams visai nereikia žinoti kur ir kokią techninę įrangą naudoja jų programos. Programuotojas nesirūpina kaip valdomi skaičiavimo resursai. Debesyje yra visi reikalingi resursai ir juos naudodami mokame (dažniausiai Cloud paslaugos tiekėjui), tik už sunaudotą skaičiavimo resursų laiką. Dažniausiai organizacijos naudoja skaičiavimo telkinius stambiams moksliniams, ar verslo sistemų uždaviniams spręsti. Tokios užduotys gali būti skaidomos į smulkesnes ir eilės būdu sprendžiamos debesyje. Pasinaudojant debesies paslauga, organizacija, su specialiais uždaviniais, gali dirbti naudodama 1000 serverių vieną valandą, ir mokėti tiek pat, kiek naudodama vieną serverį 1000 valandų. Taigi milžiniški skaičiavimo resursai gaunami greitai, už tokią pačią kainą kaip ir naudojant mažus resursus ilgą laiką.

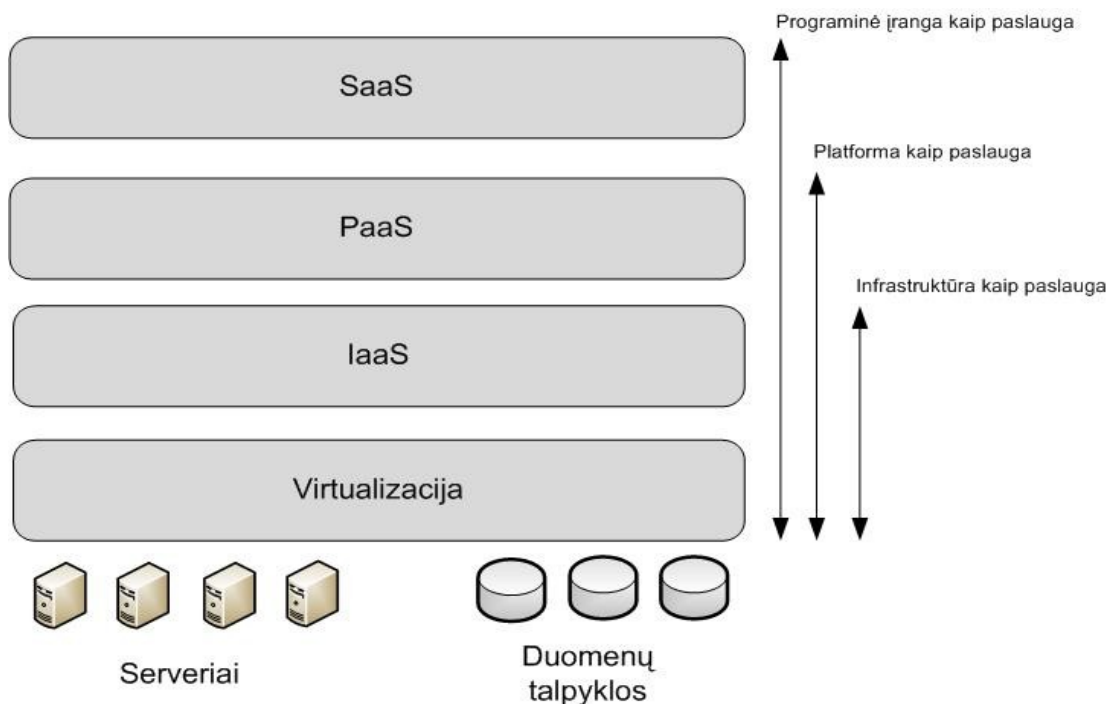
Visa tai įgyvendinama panaudojant resursų virtualizacijos technologijas. Cloud skaičiavimai turi ir trūkumų, kurie trukdo pritaikyti Cloud principus visoms IT sistemoms. Šios problemos gali atsirasti tiek programos sluoksniuose, tiek iš virtualių resursų sluoksnių, tiek iš saugumo pusės, bei taip pat egzistuoja daugybė teisinių problemų, nes duomenų fizinė vieta nėra svarbi programai, kuri dirba debesyje.

2. Debesies paslaugų lygmenys

Šiuo metu atsiranda įvairiausių debesies apibrėžimų ir kartais galima pasimesti, kai kuriais galima pasitikėti, kai kuriais ne, nes tai dažniausiai būna marketinginė medžiaga. U.S. National Institute of Standards and Technology (NIST) pateikia nedidelį dokumentą [MGNistv15], kuriame pažymimos pagrindinės cloud charakteristikos, bei debesies paslaugos tiekimo lygmenys. Dabartinė, naujausia dokumento versija v15.

Debesies kompiuterijos paslaugų lygmenys skirstomi į tokius tris pagrindinius lygius (Pav. 1):

- **Cloud Infrastructure as a Service (IaaS) (Infrastruktūra, kaip paslauga);**
- **Cloud Platform as a Service (PaaS) (Platforma, kaip paslauga);**
- **Cloud Software as a Service (SaaS) (Programinė įranga, kaip paslauga).**



Pav. 1

2.1 Infrastruktūra, kaip paslauga

Infrastruktūra, kaip paslauga (IaaS) – toks modelis leidžia išnuomoti/pateikti resursus (procesoriaus darbą, atmintį, talpą, tinklą ir t.t.) Paslaugos gavėjas (IaaS paslaugos gavėjas) pats kontroliuoja operacinę sistemą, tinklo nustatytumus, talpos dydį. Gavėjas pats gali, - kurti virtualias mašinas, diegti jam reikalingas operacines sistemas, kurti tinklo struktūras, reguliuoti resursų apkrovimo paskirstymus. Vienas iš esminių šios paslaugos pranašumų yra tai, kad klientas gali, bet kada padidinti ar sumažinti valdomų resursų kiekį. Naudojantis šia paslauga paslaugos gavėjas kontroliuoja patį žemiausią debesies paslaugų lygmenį – virtualiai aparatūrinį arba virtualius įrenginius. Čia paslaugos tiekėjas dažniausiai panaudoja virtualizacijos technologiją, kuri leidžia valdyti virtualius įrenginius (diskinė talpa, tinklo adapteriai, procesoriai, atmintis). Tokios paslaugos tiekėjų pavyzdžiai būtų: Amazon EC2 (Amazon Elastic Compute Cloud, S3 storage), RackSpace.com, Elastichosts.com. Šiame darbe plačiau bus nagrinėjamas debesies infrastruktūra, kaip paslauga, tipo modelis. Todėl, taip pat bus apžvelgiama virtualizacijos technologija, kuri atlieka svarbų vaidmenį šiame IaaS paslaugos lygyje.

IaaS sprendimai, įprastai turi šiuos komponentus:

- Atsiskaitymas už sunaudotus resursus (pavyzdžiui už vykdymo valandas);
- virtualizacijos įranga, galinti vykdyti kliento virtualias mašinas;
- aparatūrinė įranga, įprastai sujungta į aukšto patikimumo klasterius;
- kompiuterių tinklas, su ugniasienėmis, apkrautumo balansavimu, ir t.t.;
- interneto prieiga.

2.1.1 Programinė įranga bei kompanijos teikiančios infrastruktūros paslaugas

Yra sukurta nemažai programinės įrangos, teikiančios infrastruktūros paslaugas. Atviro kodo sprendimų pavyzdžiais gali būti: Eucalyptus, OpenNebula, Enomaly, Nimbus, GoGrid.

Pasaulyje gerai žinoma Amazon kompanija, iš daugelio teikiamų paslaugų, teikia ir infrastruktūros paslaugas. Tai „Amazon EC2“ paslauga (Amazon Elastic Computing Cloud). Plačiau panagrinėsime jų teikiamas paslaugas.

Amazon EC2

Amazon Elastic Compute Cloud (dažnai vadinamas EC2) yra komercinė web paslauga, kuri klientams nuomoja virtualius kompiuterius, kuriuose klientai gali paleidinėti savas aplikacijas. EC2 suteikia lankstų aplikacijų naudojimą, teikdamas web paslaugų sąsają (angl. API), kurios pagalba vartotojas gali susikurti virtualias mašinas, kuriose savo pasirinkimu įsikelia programinę įrangą. Vartotojas gali kurti, paleisti ir sustabdyti virtualias mašinas pagal savo poreikius, sumokėdamas už valandas, kiek serveris buvo aktyvus. Suteikiama galimybė susikurti virtualius serverius atskirose izoliuotose zonose, vieną virtualų serverį naudojant kaip pagrindinį, antrąjį kaip atsarginį serverį, iškilus nenumatytiems gedimams. Tokiu būdu įvykus gedimui, labai sumažinamas serverio nepasiekiamumo laikas, kai klaidos atveju atstatoma atsarginė kopija.

Virtualios mašinos

EC2 naudoja Xen virtualizacijos technologiją. Kiekviena virtuali mašina funkcioja kaip virtualus privatus serveris. Šie virtualūs serveriai gali būti trijų dydžių: mažas, didelis ir ekstra didelis. Amazon.com virtualių mašinų dydžius skirsto pagal „EC2 skaičiavimo vienetus“ – aparatūrinės įrangos CPU galia.

Apmokėstinimas

Amazon taiko dviejų tipų mokesčius:

- Valandinis mokestis už virtualią mašiną
- Duomenų persiuntimo mokestis

Valandinis mokestis yra fiksuotas, atsižvelgiant į virtualios mašinos talpą ir savybes. Amazon apibrėžia kainų politiką taip: „Jūs sumokate už resursus, kuriuos suvartojate“.

2.2 Platforma, kaip paslauga

PaaS tai IaaS ir programinės įrangos leidžiančios valdyti IaaS resursus junginys. Ji palengvina taikomųjų programų diegimą, išvengiant išlaidų sudėtingiems aparatūriniais bei programiniams sprendimams. PaaS pilnai palaiko visą infrastruktūrą, reikalingą web aplikacijų ir paslaugų kūrimui ir diegimui, kurios yra pasiekiamos internetu. Galutiniam vartotojui, IT vadybininkui ar programinės įrangos kūrėjui nereikia siųstis ar diegti jokios programinės įrangos, viską pateikia PaaS.

Šiame lygyje pats paslaugos gavėjas paleidžia savo aplikacijas parašytas kurią nors kalba (pvz. Java, python, .Net, Ruby) Jis negali kontroliuoti pačios debesies infrastruktūros – tinklo, virtualių mašinų, operacinės sistemos, sisteminės programinės įrangos. Tačiau jis gali pats kontroliuoti pačią aplikaciją, jos aplinką, naudoti specialius paslaugos tiekėjo siūlomus įrankius, kuriais gali kontroliuoti programos veikimą. PaaS sprendimai apima darbų paskirstymą projektuojant programinę įrangą, programinės įrangos kūrimą, testavimą, programinės įrangos diegimą ir talpyklos paslaugas. Taip pat kai kurios PaaS paslaugos palaiko komandinį bendradarbiavimą, web paslaugų integravimą, duomenų bazės panaudojimą, saugumą, keičiamumą, talpyklą, būklės stebėjimą, aplikacijų versijavimą, kūrėjų bendruomenės subūrimą. Visos šios paslaugos yra teikiamos kaip integruotas sprendimas į web aplinką. Tokios paslaugos komerciniai pavyzdžiai galėtų būti: AppJet, Google App Engine, Force.com.

2.3 Programinė įranga, kaip paslauga

Programinės įrangos kaip paslaugos lygmuo (SaaS – Software as a Service) įsivaizduojamas kaip programinės įrangos prieiga per internetą, kai programinė įranga saugoma duomenų centruose, o vartotojas interneto pagalba gali ja naudotis, nediegdamas įrangos savo kompiuteryje. Taip pat programinė įranga gali būti parsisiunčiama iš duomenų centrų, ir kliento kompiuteryje veikti ribotą laiką, pavyzdžiui tol, kol baigiasi sutartis. Tai gali būti pasiekama naudojant bendrą licenzijų serverį, kuris paleistas tiekėjo duomenų centre. Šitoks programinės įrangos panaudojimo modelis, turi keletą privalumų.

Naudojant SaaS, išsprendžia licenzijavimo problemas. Vartotojui nebereikia diegti programinės įrangos į kiekvieną kompiuterį, ir pirkti atitinkamą kiekį licenzijų. Naudojant paslaugų modelį, vartotojas moka tik už laiką, ir licenzijų kiekį, kiek tuo metu jų vartoja.

Taip pat SaaS sumažina administravimo kaštus, kai vartotojui nebereikia diegti programinės įrangos į kiekvieną kompiuterį, arba ją atnaujinti, pakanka paslaugų tiekėjams atnaujinti programinę įrangą serveriuose.

Nepaisant technologijos privalumų, ši technologija, kaip ir visos turi ir trūkumų. Programinė įranga, kurią teikia SaaS paslauga, turi būti pritaikyta dirbti internetu, būti pasiekiami per naršyklę, arba kitais būdais pasiekiami nuotoliniu būdu.

Kai kurie ekspertai, programinės įrangos kaip paslaugos (SaaS) lygmenį skirsto į keturis brandos lygius, kurių požymiai yra: konfigūravimo lankstumas, globalaus panaudojimo galimybė, keičiamumas. Kiekvienas brandos lygis skiriasi nuo ankstesniojo, pridėdamas po vieną iš šių požymių. (Lentelė 1)

1 Lygmuo – Specialus/Padarytas pagal užsakymą: pirmame brandos lygmenyje, kiekvienas klientas turi pagal savo poreikius pritaikytą aplikaciją, ir paleidžia ją duomenų centro serveriuose. Netinklinės arba kliento – serverio architektūros programinės įrangos perėjimas į šį SaaS lygmenį reikalauja mažiausiai išsivystymo pastangų, ir mažina eksploatacijos išlaidas, apjungdamas serverio aparatinę įrangą ir administravimą.

2 Lygmuo – Konfigūracija: Antrasis brandos lygmuo suteikia didesnę programos lankstumą panaudojant konfigūruojamus metaduomenis, kad daug klientų galėtų panaudoti atskirus to paties taikomojo kodo pavyzdžius. Tai leidžia paslaugų tiekėjui patenkinti skirtingus kliento norus, pateikiant išsamų konfigūracijos pasirinkimą, supaprastinant palaikymą ir bazinio kodo atnaujinimą.

3 Lygmuo – Konfigūracija, globalaus panaudojimo galimybė: trečiasis brandos lygmuo prideda globalaus panaudojimo galimybę prie antrojo lygmens, kai vienas programos pavyzdys aptarnauja visus klientus. Šis metodas įgalina efektyvesnę serverio išteklių panaudojimą, be jokio matomo skirtumo galutiniam vartotojui, tačiau šis būdas turi keičiamumo ribas.

4 Lygmuo – Konfigūracija, globalaus panaudojimo galimybė, keičiamumas: ketvirtas ir paskutinis SaaS brandos lygmuo prideda dydžio keičiamumą naudojant daugiapakopę architektūrą. Ši architektūra turi palaikyti apkrovimą paskirstantį identiškų taikomųjų pavyzdžių telkinį, paleistą ant kintamo skaičiaus serverių. Paslaugų tiekėjas gali padidinti arba sumažinti sistemos pajėgumus, keisdamas naudojamų serverių skaičių, nedarydamas jokių programinės įrangos keitimų.

	Pagal užsakymą	Konfigūracija	Globalus panaudojimas	Keičiamumas
1 lygmuo	+			

2 lygmuo		+		
3 lygmuo		+	+	
4 lygmuo		+	+	+

Lentelė 1

3. Debesų kompiuterijos dislokacijos modeliai

Priklausomai kaip naudojami debesies resursai, bei kur yra padėti duomenys, pagal tai galime apibrėžti dislokacijos modelį.

3.1 Privatus debesis (angl. Private cloud)

Šiame modelyje visa debesies infrastruktūra naudojami tikrai vienos organizacijos. Prižiūrėti tokią infrastruktūrą gali pati organizacija, arba organizacija iš trečiosios šalies. Fiziškai visa aparatūrinė įranga gali būti tiek organizacijos pastatuose, arba išorėje.

3.2 Bendras debesis (angl. Community cloud)

Tokia debesies infrastruktūra yra naudojama ne vienos, o kelių organizacijų. Dažniausiai tokių organizacijų veiklos glaudžiai susijusios. Jos turi bendras sistemų saugumo politikas, reikalavimus, todėl tokios sistemos gali dirbti vienoje bendroje infrastruktūroje. Valdymas, priežiūra taip pat, kaip ir privačiame debesyje gali būti vykdomas trečiosios šalies įmonės.

3.3 Viešas debesis (angl. Public cloud)

Tai visiems prieinama debesies paslauga. Tokia infrastruktūra dažniausiai priklauso organizacijai, kuri parduoda debesies kompiuterijos paslaugas. Tokio modelio debesyse geografinė duomenų vieta praranda prasmę, nes tiekėjo infrastruktūra gali būti išdėstyta per kelias šalis, žemynus. Viešuose debesyse gali dirbti įvairios skirtingos sistemos, dažniausiai izoliuotos nuo kitų sistemų darbo. Virtualių resursų panaudojimą kontroliuoja paslaugos gavėjas, o techninę įrangą, bei valdymo įrankius suteikia, bei prižiūri debesies paslaugos arba viešojo debesies tiekėjas. Tam, kad teikti viešo debesies paslaugas daugybei klientų, paslaugos tiekėjas turi turėti didelius techninius pajėgumus. Todėl tokios kompanijos, kaip Google ar Amazon, naudodamos savo milžiniškas, gerai sustyguotas infrastruktūras savo sistemų darbui, gali kartu parduoti resursus kitoms kompanijoms, kurioms tų resursų reikia. Gerai žinomas viešo debesies paslaugos pavyzdys galėtų būti, jau anksčiau minėta Amazon Elastic Compute Cloud (Amazon EC2) paslauga.

3.4 Hybridinis debesis (angl. Hybrid cloud)

Tai dviejų ar daugiau modelių rinkinys, kuris gali susidėti iš privataus, bendro ar viešo debesies modelių. Pvz: tokio tipo debesyse organizacija gali naudotis savo privatų debesį o poreikiams išaugus galima prijungti daugiau resursų iš viešai prieinamo debesies.

4. Virtualizacija debesyje

Norėdami paaiškinti, virtualizacijos svarbą debesies skaičiavimuose, reikėtų paminėti ir kitas skaičiavimo platformas.

- **Skaičiavimo telkinys (angl. Cluster)** – tai bendra, paskirstytų skaičiavimų sistema, susidedanti iš atskirų kompiuterių, serverių, kurie yra sujungti į vieną bendrą infrastruktūrą. Skaičiavimo telkinį galime interpretuoti, kaip vieną skaičiavimo resursų vienetą.[MARBVIC07]
- **Gridas (angl. Grid)** – lygiagrečių ir paskirstytų skaičiavimų tinklas, susidedantis iš daugybės skaičiavimo resursų vienetų (telkinių, galingų super kompiuterių). Skaičiavimo resursai prijungiami prie bendro grid tinklo, kuriuo gali naudotis grid autorizuoti vartotojai.

- **Debesies skaičiavimai (angl. Cloud computing)** - lygiagrečių ir paskirstytų skaičiavimų sistema, susidedanti iš virtualių kompiuterių (angl. Virtual machines), virtualios talpos (angl. Virtual storage), kuriuos pagal poreikius galima sukurti per kelias minutes.

Taigi debesies skaičiavimus galime apibūdinti, kaip telkinių ir grido kombinaciją, nes jis turi tiek skaičiavimo telkinių, grido charakteristikų. Reikia pridurti, taip pat virtualizacijos technologiją, kuri suteikia didelį lankstumą debesies infrastruktūrai, aukštą patikimumą, nepertraukiamo darbo vientisumą, dinamiškumą ir patogų centralizuotą valdymą.

4.1 Virtualizacija

Kas yra toji virtualizacija? Virtuali infrastruktūra? Virtualizacija sukuria sluoksnį (virtualizacijos sluoksnis) tarp aparatūrinės įrangos ir programinės įrangos/operacinės sistemos. Tokia tarpinė grandis leidžia įgyvendinti, tokius technologinius sprendimus programiškai, kokių neįmanoma arba labai sudėtinga įgyvendinti aparatūrinėje dalyje. [BDFXenArt03]

Virtualizacijos technologija gali būti pritaikyta, ne tik pvz. virtualiai atminčiai, bet ir kitiems infrastruktūros sluoksniams: operacinė sistema, tinklo komponentai, talpyklos, aplikacijos ir t.t. Visa tai vadinama - virtualia infrastruktūra (Virtual Infrastructure). Virtualizacija iš esmės nereikalauja kažkokių specifinių žmogaus veiksmų, norint suvaldyti tokias virtualias infrastruktūras, kadangi joje objektai atitinka realaus pasaulio objektus, kaip pavyzdžiui procesorius ar tinklo adapteris, tačiau čia jie yra virtualūs.

Virtualizacija leidžia vienoje fizinėje mašinoje paleisti kelias ar daugiau virtualių mašinų su skirtingomis operacinėmis sistemomis. Tos virtualios mašinos (angl. VM – virtual machine) naudoja virtualius įrenginius, virtualią atmintį, virtualius procesorius.

4.2 Virtualizacijos tipai

Šiuo metu yra trys virtualizacijos tipai: pilna virtualizacija (full virtualization), paravirtualizacija (paravirtualization), įrangos technologija (hardware assist). Pagal procesoriaus instrukcijų vykdymą, procesų sluoksniuose išskiriami trys virtualizacijos tipai. Pabrėžiama, kad toks tipas kaip paravirtualizacija, reikalauja, kai kurių pakeitimų operacinės sistemos branduolyje, kas suvaržo palaikomų operacinių sistemų aibę. Tačiau tai leidžia efektyviau išnaudoti turimus resursus, virtualizuojant operacines sistemas.

Pilnoji virtualizacija (angl. Full virtualization) turi viena nemažą privalumą, - operacinė sistema, kuri dirba virtualioje mašinoje neprivalo būti modifikuota ir tai leidžia virtualioje aplinkoje naudoti įvairias operacinių sistemų versijas. Tačiau pats virtualizacijos sluoksnis (angl. Hypervisor) naudojantis šią technologiją yra sunkiai programiškai įgyvendinamas.

Įrangos technologija, čia reikėtų paminėti žinomų procesorių gamintojų Intel Virtualization Technology (VT-x) ir kompanijos AMD - AMD-V technologijas. Procesoriai naudojantis šias technologijas įgalina operacinės sistemos aukšto rango iškvietimus į procesorių perleisti tiesiai į hypervisorį – vykdyti iš taip vadinamo root režimo, nenaudojant binary translation arba paravirtualizacijos principų. Kas labai sumažina virtualizacijos sluoksnio sudėtingumą.

Kai kurie VMware kompanijos pateikti dokumentai, teigia, kad “binary translation – full virtualization” virtualizacija išlieka geriausias sprendimas spartos, patogumo atžvilgiu.

Įrangos technologijomis paremta virtualizacija, ateityje tobulės, nes Intel ir AMD labai stipriai dirba tobulindamos procesorius, tam kad virtualizuojant sistemas būtų pasiektas didesnis efektyvumas.

Paravirtualizacija turi didelį paplitimą nemokamose Linux distribucijose. Xen vienas iš populiariausių paravirtualizacijos produktų, kurį dabar galima rasti beveik kiekvienoje populiarioje linux distribucijoje. Tiek VMware, tiek Xen produktai stengiasi naudoti, pritaikyti atvirus standartus, kai kurie iš jų jau taikomi įvairiose virtualiose aplinkose, keli pvz:

- Virtualios mašinos prieiga (Virtual Machine Interface) – API tarp hypervisoriaus ir virtualios mašinos operacinės sistemos.
- Management Interface – karkasas skirtas valdyti virtualias sistemas, virtualias mašinas ir t.t.
- Virtual Machine Disk Format – virtualių mašinų diskų formatas, kuris leidžia virtualių mašinų migraciją/pernešimą tarp skirtingų virtualizacijos platformų.

Tiek pilnosios virtualizacijos, tiek paravirtualizacijos atveju, tam tikrą kiekį resursų turi suvalgyti virtualizacijos sluoksnis. Pilnoji virtualizacija (Vmware) naudoja 7%-9% procentų resursų savo skaičiavimams. Paravirtualizacija sunaudoja tikrai apie 3% resursų savo reikmėms.

Kaip teigiame daugelyje dokumentų tiek full virtualization, tiek paravirtualizacija ateityje bus plačiai naudojama. Nereikia pamiršti ir techninės įrangos patobulinimų (hardware assist), kurie vis daugiau ir daugiau, praplečia virtualizacijos galimybes. [ITCVMSH06]

4.3 Virtuali mašina

Virtualią mašiną (virtualus kompiuteris) -galime įvairinti kaip vieną failą, kuris apibrėžia mašinos virtualius įrenginius. Toks failas gali būti nukopijuotas, perkeltas į kitą vietą. Atsiradus tam tikriems gedimams operacinėje sistemoje, galima tiesiog paimti virtualios mašinos kopija ir vėl ją paleisti virtualioje aplinkoje. Virtuali mašina visiškai atitinka realią fizinę mašiną. Ją galima įjunti, išjungti, perkrauti, įtraukti papildomus virtualius įrenginius ir t.t. viskas taip pat kaip su paprastu kompiuteriu, tik tai veiksmai atliekami programiškai.

4.4 Virtualūs įrenginiai

Kiekviena virtuali mašina turi savo virtualius įrenginius, kuriuos pateikia operacinei sistemai. Virtualius įrenginius galima skirstyti į tris pagrindines grupes: virtualus procesorius, virtuali atmintis, bei virtualūs įvesties/išvesties įrenginiai.

4.4.1 Virtualus procesorius

Tai vienas iš svarbiausių virtualių įrenginių, nes jis vykdo visas operacinės sistemos skaičiavimo instrukcijas. Kaip buvo minėta anksčiau, virtualizuojant procesorių šiuo metu labiausiai paplitusios trys virtualizacijos technologijos: pilna virtualizacija (full virtualization), paravirtualizacija (paravirtualization) ir įrangos technologija (hardware assist). Dažniausiai vienai virtualiai mašinai galima priskirti ir daugiau, nei vieną virtualų procesorių, tokiu atveju operacinė sistema naudos kelis procesorius vienu metu, taip pat kaip naudojant kompiuterius su daugiau nei vienu procesoriumi. Dar vienas didelis privalumas, kad virtualios mašinos gali dalintis fizinius procesorius. Tokiu atveju fiziniame serveryje su 4 fiziniais procesoriais (branduoliais), panaudojant virtualizacijos technologiją, galima dirbti su 20 virtualių procesorių, nuo kiekvieno fizinio procesoriaus. Taip pat virtualūs procesoriai yra dalinami (virtualizacijos sluoksnio) pagal apkrovimus virtualioms mašinoms. Taigi galime turėti daugybę virtualių mašinų, kurios naudoja virtualius procesorius, žinant, kad fiziškai turime tik 4 fizinius procesorius.

4.4.2 Virtuali atmintis

Kitas svarbus įrenginys, kuris yra virtualizuojamas, tai operatyvioji atmintis. Virtualizuojant atmintį siekiama, efektyviai, dinamiškai paskirstyti atmintį, visoms vieno fizinio serverio virtualioms mašinoms. Virtualizacija leidžia šį procesą valdyti ir rankiniu būdu, taigi atmintį paskirstyti virtualioms mašinoms galima kaip nori. Už visą atminties paskirstymą atsakingas virtualizacijos sluoksnis. Naudojant papildomas procesorių ypatybes, tokias kaip pvz: memory management unit (MMU) arba translation lookaside buffer (TLB), optimizuojamas virtualios atminties valdymas. Dar vienas virtualios atminties privalumas toks, kad nenaudojamą virtualią atmintį galima “paskolinti” kitai virtualiai mašinai, tai vadinama angl. Memory overcommitment. Pavyzdžiui turime virtualią mašiną su jai priskirtu 1 GB atminties, ši virtuali mašina dirba beveik nenaudodama resursų (nėra apkrovimo), kita virtuali mašina, su jai priskirtu 1 GB atminties turi didelius apkrovimus, išnaudojama visa 1 GB atmintis, nenurtraukiant darbo, virtualizuotą atmintį iš pirmos mašinos, galima paskolinti antrajai mašinai, tokiu būdu dinamiškai padidinti atminties kiekį, kol yra dideli apkrovimai.

4.4.3 Virtualūs įvesties/išvesties įrenginiai

Virtualizacijos sluoksnis taip pat turi virtualizuoti ir kitus įrenginius, tokius kaip tinklo adapteriai, diskai, išorinė talpa, optiniai įrenginiai ir kt. Taigi virtualizacijos sluoksnio užduotis siuntinėti komandas tarp virtualių įrenginių ir tam tikro vieno ar kelių fizinių įrenginių. Paprasčiausias pavyzdys galėtų būti tinklo adapteris,- jis fiziškai yra vienas, tačiau virtualioje aplinkoje mes galime sukurti daugybę virtualių tinklo adapterių, ir visa tai valdyti iš virtualizacijos sluoksnio. Tokiu būdu vienas fizinis įrenginys yra padalinamas kelioms virtualioms mašinoms. Taip pat reikėtų pabrėžti, kad visi virtualūs įrenginiai yra standartizuoti, vienodi, nesvarbu koks įrenginys dirba fizinėje aplinkoje, virtuali mašina operacinei sistemai pateikia tam tikro, vieno tipo įrenginį.

4.5 Virtuali infrastruktūra

Virtualizacijos serveriu pavadinsime vieną fizinį serverį, kuriame yra įdiegtas virtualizacijos sluoksnis, virtualių mašinų darbui. Toliau galime pereiti prie visos virtualios infrastruktūros, kur yra ne vienas serveris, o keli arba jų daugybė. Čia reikia paminėti išorines talpyklas, kur laikomos pvz virtualios mašinos, kiti sistemoms reikalingi duomenys. Tokia įranga yra didelės greitaveikos, aukšto patikimumo ir brangi. Trumpai tariant virtualios infrastruktūros leidžia padidinti serverių darbo efektyvumą, konsoliduoti duomenis, greitai sukurti reikiamus resursus, lengvai ir patogiai valdyti visą infrastruktūrą. Toliau bus remiamasi VMWARE kompanijos pateikta dokumentacija, visi terminai paiimti iš VMWARE dokumentacijos apie virtualią infrastruktūrą ir jos komponentus (angl. Virtual Infrastructure). Aptarsime kokias galimybes suteikia virtuali infrastruktūra.

4.5.1 Konsolidacija ir izoliacija

Dirbdami, su nevisą kompiuterių galią, išnaudojančiomis aplikacijomis, mes neefektyviai naudojame elektros energiją, reikalingą kelių serverių darbui. Pasinaudodami virtualizacija, visas tokias programas galime paleisti vienu fiziniu serveriu. Taigi virtualioje infrastruktūroje konsoliduojame duomenis.

Izoliacija leidžia kiekvieną virtualią mašiną laikyti kaip atskirą aplinką, t.y. viename fiziniame serveryje, atskirų vartotojų virtualios mašinos yra žemame lygyje izoliuotos viena nuo kitos. Tai leidžia saugiai turėti daug vartotojų viename ar keliuose fiziniuose serveriuose.

4.5.2 Įsibrovimų aptikimo sistema

Dar vienas virtualizacijos privalumas, toks, kad įsibrovimų aptikimo sistema gali dirbti izoliuota nuo virtualios mašinos. Jeigu tokia sistema dirba iš virtualizacijos sluoksnio, tai, bet koks virusas, kuris modifikuoja operacinės sistemos branduolį, niekaip neapgaus šios sistemos, kadangi ji dirba iš dar žemesnio lygio, nei operacinės sistemos branduolys – iš virtualizacijos sluoksnio ir yra izoliuota nuo išorės.

4.5.3 Virtualių mašinų migracija

Vienas iš didžiausių virtualizacijos privalumų – virtualių mašinų migracija iš vienos fizinės mašinos į kitą, kas yra lengviau įgyvendinama, negu tam tikrų programos procesų migracija. Kaip anksčiau buvo apibrėžta, virtuali mašina tai tam tikras paketas, kuriame yra operacinė sistema, vartotojo programos ir kiti virtualūs įrenginiai. Pasinaudojant virtualių mašinų migracija (vmware-vmotion technologija, Xen – live migration technologija) duomenų centruose, virtualiose infrastruktūrose, galime labai padidinti administravimo, resursų panaudojimo efektyvumą. Daugelis žinomų virtualizacijos produktų gamintojų teigia, kad tam reikalingas Ethernet tinklas, bei bendra talpykla, prie kurios prijungti fiziniai serveriai. Naudojant specialius algoritmus, virtualios mašinos atminties puslapiai, per tinklą, persiunčiami kitai fizinei mašinai, kuri perima virtualios mašinos valdymą. Kad migruoti, veikiančias virtualias mašinas, reikia greito tinklo, su mažais tinklo paketų užlaikymais, nes veikiančiose virtualiose mašinose, aplikacijos naudoja tam tikrus atminties puslapius, kuriuos reikia persiųsti kitam fiziniam serveriui. Tam naudojami specialūs algoritmai, kurie leidžia, beveik nstabdant aplikacijos darbo, migruoti virtualią mašiną. [ITCVMSH06]; [RRTh05]

Migruojant virtualias mašinas, atminties persiuntimas vyksta nuosekliai, tai reiškia, kad ir operacinės sistemos branduolio būseną taip pat persiunčiama nuosekliai (pvz. aktyvių tinklo susijungimų, TCP kontrolinis bloko paketas). Praktiškai, tai reiškia, kad pavyzdžiui migruojant tinklo žaidimų serverį ar vaizdo transliavimo serverį, vartotojams nereikia iš naujo prisijungti prie šių tinklo paslaugų. Tai būtų neįmanoma, jei šiam veiksmui naudotume aplikacijos sluoksnį ar OSI modelio 7 sluoksnio nukreipimą.

Čia taip pat didelį privalumą turi infrastruktūros prižiūrėtojas/tiekėjas. Vartotojų programos turi savo veikimo niuansus, kurie visiškai nesvarbūs infrastruktūros operatoriui, nes jam nereikia žinoti kaip tos programos veikia, jam nereikia priėjimo prie operacinės sistemos. Iš esmės operatoriui nereikia žinoti, kas dirba tam tikroje virtualioje mašinoje. Jis virtualią mašiną mato kaip vieną objektą, kuri gali migruoti tarp kelių fizinių serverių, o vartotojai to beveik nepajaučia. Priklausomai nuo programos atsiranda tam tikri mikro sekundžių sulaikymai, tačiau tai niekaip nepaveikia programos duomenų.

Taigi virtualių mašinų migracija, tai labai galingas įrankis duomenų centruose, virtualiose infrastruktūrose, leidžiantis atskirti programinę ir techninę įrangą, konsoliduoti visą techninę įrangą ir valdyti tai kaip vieną bendrą domeną.

4.5.4 Aukštas patikimumas (High Availability)

Kad užtikrinti aukštą patikimumą, virtualiose infrastruktūrose, virtualios mašinos pakraunamos iš bendros talpyklos, tai yra greitas prieigas turinčios duomenų talpyklos. Dažniausiai naudojami sprendimai yra (angl. Storage Area Network (SAN)) arba (angl. Network Attached Storage (NAS)). Visi fiziniai serveriai naudojami virtualioje infrastruktūroje, turi būti pajungti į tokio tipo talpyklas. Iš ten pasiimamas virtualios mašinos failas (paketas), kuris užkraunamas reikiamame serveryje. Jeigu vienas serveris sugenda, sistema tai pastebi ir reikalinga virtuali mašina yra užkraunama kitame fiziniame serveryje, tokios būsenos, kokios buvo prieš gedimą, tačiau aplikacijos būseną dažniausia neišsaugoma, nes nėra išsaugoma operacinės sistemos atmintis, todėl toks metodas dažniausiai tinka tokioms paslaugoms, kaip žiniatinklio serveris (angl. Web server), kur nereikia išsaugoti ar rašyti duomenų. Procesas vyksta automatiškai, nereikalaujant žmogaus įsikišimo. [ITCVMSH06]

4.5.5 Dinaminis resursų paskirstymas

Panaudojant virtualių mašinų migraciją, virtualiose infrastruktūrose, duomenų centruose, galima labai efektyviai paskirstyti skaičiavimo resursus virtualioms mašinoms. Kaip pavyzdį imkime VMware kompanijos technologija, pavadinimu VMware DRS (Distributed Resource Scheduler). Tai leidžia, pagal poreikius paskirstyti resursus visoje virtualioje infrastruktūroje, nepertraukiant virtualių mašinų darbo. Procesoriai, atmintis ir talpa apibrėžiami kaip resursų paketai (angl. Resource Pools) nepriklausomai, kur fiziškai jie yra. Virtualios mašinos gali būti suskirstytos į loginias grupes, taip pat nepriklausomai, kokiam fiziniame serveryje jos dirba. DRS sistema ne tik stebės visų mašinų apkrovimus (angl. Monitoring), tačiau pati automatiškai paskirstys visos infrastruktūros turimus resursus virtualioms mašinoms. Pavyzdžiui vienos virtualios mašinos apkrovimas labai padidėjo, tuomet sistema patikrina kokius resursų nustatymus yra sukurti šiai virtualiai mašinai. Pagal tai virtuali mašina gali būti migruojama į kitą fizinį serverį, kuris tuo metu nėra taip apkrautas. Šiuo keliu, galima užtikrinti labai svarbių aplikacijų darbo greitį, patikimumą. [ITCVMSH06]

4.5.6 Energijos vartojimo valdymas

Reikia paminėti DPM procesą (angl. Distributed Power Management), kuris leidžia duomenų centrums sutaupyti nemažai elektros energijos. Specialiems virtualizacijos įrankiams, tokiems kaip minėtas DRS, stebint virtualių mašinų apkrovimus, galimas fizinių serverių stabdymas. Jeigu virtualios mašinos naudoja labai mažus resursus, tai leidžia joms dirbti mažesniame fizinių serverių skaičiuje. Tokius fizinius serverius, kurie yra tuo momentu nereikalingi tiesiog išjungti/sustabdyti, ir visa tai gali būti atliekama automatiškai. Kada tai naudinga? Pats paprasčiausias pavyzdys galėtų būti, naktinis laikas, kai serveriai turi labai mažus apkrovimus, tuo momentu virtualios mašinos gali aptarnauti labai mažą skaičių klientų ir nereikalauti didelių resursų. Išaušus rytui, apkrovimas pastebimai padidėtų, sistema automatiškai suteiktų naujų resursų (fiziškai įjungtų serverius), bei migruotų virtualias mašinas į reikiamus serverius. [ITCVMSH06]

5. Virtualizacijos produktai

5.1 VMware virtualizacija

Tai vienas iš populiariausių komercinių virtualizacijos produktų. Su tokiais galingais įrankiais kaip Virtual Infrastructure, galima labai lengvai ir greitai susikurti savo virtualias infrastruktūras. VMware produktai dažniausiai turi patogias ir aiškias vartotojo sąsajas. VMware infrastructure turi daug virtualizacijos naudojamų privalumų:

- Virtualių mašinų migracija;
- aukštas patikimumas;
- dinaminis resursų paskirstymas;
- energijos vartojimo valdymas.

VMware produktuose naudojamas pilnos virtualizacijos principas, tai leidžia dirbti su įvairiomis operacinėmis sistemomis. Tačiau šiek tiek resursų tenka paskirti virtualizacijos sluoksniui, nes programiškai jis yra sudėtingas ir jam reikia atlikti tam tikrus sudėtingus skaičiavimus.

VMware virtualizacija leidžia taip pat naudoti paravirtualizacijos principą, įdiegus specialias tvarkykles į operacinę sistemą. [ITCVMSH06]

5.2 Xen virtualizacija

Xen virtualizacija labiausiai paplitusi tarp mokamų ir nemokamų Linux distribucijų. Tai atvirojo kodo produktas, kuris buvo pradėtas kurti Kembridžo universiteto kompiuterių laboratorijoje (computer laboratory of the University of Cambridge, UK.). Vėliau ši produktą tobulino ir kūrė Xen bendruomenė. Egzistuoja taip pat ir komercinė šio produkto versija – XenServer, prižiūrima kompanijos Citrix.

Anksčiau, kai dar nebuvo procesorių su VT-x (intel) ir AMD-V (AMD) technologijomis, Xen dirbo tikrai kaip paravirtualizacijos technologija, o tai reiškė, kad operacinė sistema dirbanti virtualioje mašinoje turėjo būti šiek tiek modifikuota. Procesorių gamintojams įdiegus šias technologijas, atsirado galimybė supaprastinti virtualizacijos sluoksnį, kas leido dirbti su nemodifikuotomis operacinėmis sistemomis.

Xen virtualizacijos pagrindinė šerdis yra Xen hypervisorius (angl. Xen hypervisor) – dar kitaip, tiesiog virtualizacijos sluoksnis. Greta yra Xen hypervisor API – skirtas, padedant įvairiems įrankiams valdyti virtualias mašinas, bei kitas virtualizacijos funkcijas. [BDFXenArt03]

6. Virtualūs debesies resursai

Taigi naudojant virtualizacijos technologiją skaičiavimo resursus galima laikyti loginiais objektais. Programiškai tokie loginiai objektai lengviau valdomi ir lengviau suprantami, nei techninė įranga. Taigi, jeigu pažvelgsime į skaičiavimų debesies (cloud computing) terminą, matome, kad resursai yra dinamiškai skirstomi, didinami, mažinami kai to reikia, o tai įgyvendinti leidžia, - virtualizacija. Kliento požiūriu tokie veiksmai turėtų būti atliekami automatiškai, arba su minimaliu žmogaus įsikišimu. Tačiau ne viskas taip lengvai padaroma. Tam reikalingi specialūs įrankiai leidžiantys kuo efektyviau subalansuoti sistemų darbą ir pritaikyti tas sistemas dirbti naudingai debesyje.

7. OpenNebula platforma

OpenNebula yra atviro kodo virtualios infrastruktūros variklis, kuris įgalina virtualių paslaugų kūrimą ir perkėlimą keletame skaičiavimo telkiniu. OpenNebula išplėčia virtualizavimo platformas nuo vieno fizinio resurso iki resursų telkinio, atskiriant serverį ne tik nuo fizinės infrastruktūros, bet ir nuo fizinės dislokacijos vietos. OpenNebula gali būti naudojamas kaip virtualizacijos įrankis

skirtas administruoti virtualią infrastruktūrą duomenų centre arba telkinyje (angl. Cluster). Šis produktas dažnai vadinamas privačiu debesies sprendimu, tačiau naudojantis OpenNebula galima dinamiškai išplėsti į keletą išorinių debesies sistemų, sukuriant hibridinį debesį.

OpenNebula pertvarko fizinių klasterių į lanksčią virtualią infrastruktūrą, kuri dinamiškai prisitaiko prie paslaugų apkrovimo poreikių. OpenNebula naudoja egzistuojančią virtualizacijos platformą, bei sukuria naują virtualizacijos lygmenį esantį tarp paslaugų bei fizinės infrastruktūros. Šis naujas lygmuo palaiko paslaugų vykdymą fiziniuose klasteriuose, išplečiant virtualių mašinų stebėjimo įrangą nuo vieno fizinio resurso, į resursų klasterį. OpenNebula sėkmingai atskirą serverį ne tik nuo fizinės infrastruktūros, bet taip pat ir nuo fizinės dislokacijos vietos.

OpenNebula išplėčia virtualių mašinų hypervizorių funkcionalumą, suteikdama žemiau aprašytą naudą daug serverių turinčioje aplinkoje:

Infrastruktūros sąvinkui (sistemos administratoriui):

- Centralizuotas virtualių mašinų apkrautumo administravimas;
- apkrautumo balansavimas, siekiant padidinti efektyvumą bei sistemos išnaudojimą;
- serverių apjungimas, taip sumažinant fizinių serverių skaičių, dėl to sumažėja energijos bei šaldymo įrangos sąnaudos
- dinaminis fizinės infrastruktūros plėtimas pridėdant naujus serverius;
- dinaminis klasterių skirsniavimas norint paleisti skirtingas paslaugas;
- heterogeninių užduočių palaikymas, norint paleisti skirtingą programinę įrangą;
- dinaminis privačios infrastruktūros plėtimas.

Vartotojui (paslaugos administratoriui):

- Virtualių mašinų kūrimas pagal reikalavimą, patenkinant galutinių paslaugos vartotojų poreikius.

Sistemos integratoriui:

- Atvira bei lanksti architektūra, bei sąsajos;
- atviro kodo programinė įranga, naudojama Apache licenzija;
- integracija su bet kokia virtualizacijos sistema, tokia kaip hypervizorius, virtualių mašinų administravimo įrankiai, planuotojai.

OpenNebula trūkumai:

Kadangi šis produktas yra visdar kūrimo stadijoje, todėl turi nemažai klaidų.

8. Eucalyptus platforma

Eucalyptus (angl. Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems) - tai atvirojo kodo infrastruktūra, leidžianti įdiegti IaaS (Infrastructure as a Service) tipo debesų kompiuterijos sistemas skaičiavimo telkiniuose (angl. clusters), atskiruose serveriuose ir pan. Tokia platforma leidžia vartotojams paleisti ir valdyti daugybę virtualių mašinų išsidėsčiusių įvairiuose fiziniuose įrenginiuose. Fiziškai, dažniausiai, tai duomenų centro ribose, kartais ir kelių duomenų centrų ribose. Dažniausiai naudojama, kaip Privataus tipo Cloud sistema. Tai leidžia tyrimų organizacijoms, universitetams turėti debesų kompiuterija sistemas, išnaudojant turimus resursus.

Ši platforma buvo pradėta kurti, The Virtual Grid Application Development Software (VGrADS) projekto iniciatyva, Computer Science departamente Kalifornijos universitete, Santa Barbaroje, profesoriaus Rich Wolski. Vėliau įkurta kompanija *Eucalyptus systems*, kuri tesia šio produkto tubulinimus, bei prižiūri pačią platformą. Šiuo metu egzistuoja du sprendimo variantai, tai nemokama eucalyptus versija ir Eucalyptus sistema su komerciniu palaikymu. [EucaS09]

Pats produktas sudarytas iš įvairių Linux operacinės sistemos nemokamų įrankių. Iš vartotojo pusės gali būti naudojamas Web interfeisas, palengvinantis priėjimą prie sistemos resursų. Taigi platforma apima šiuos debesų kompiuterija aspektus:

- virtualių mašinų kūrimas/valdymas;
- suplanuotas resursų paskirstymas;
- virtualių mašinų ir vartotojų duomenų talpinimas;
- debesų kompiuterija infrastruktūros valdymo prieiga;
- virtualių tinklų planavimas/valdymas;
- debesų kompiuterijos vartotojų prieiga;
- debesies paslaugų valdymas naudojant tam skirtą specialią sąsają.

Labai svarbu paminėti, kad Eucalyptus platforma naudoja tokius pačius interfeisus kaip komercinės Amazon EC2 (angl. Elastic Compute Cloud) ir S3 (angl. Storage) paslaugos, kurios dar vadinamos de facto standartu debesų kompiuterijos paslaugų srityje, iš esmės Eucalyptus duplikuoja komercinės paslaugos Amazon EC2 ir S3 funkcionalumą. Tai leidžia vartotojams, kurie yra gerai susipažinę su Amazon paslaugomis, pereiti prie Eucalyptus sistemos. Taip pat pavyzdžiu naudojant

vietinę Eucalyptus sistemą, reikalui esant pasinaudoti komercinėmis Amazon paslaugomis (atsiradus didesniai resursų poreikiui) t.y. aplikacija gali dirbti abiejose platformose. Kitas pavyzdys galėtų būti, kad prieš paleidžiant tam tikrą aplikaciją į darbinę būseną, ją išbandyti arba kurti privataus tipo cloud'e (Eucalyptus) ir tik ištestavus visas funkcijas, paleidinėti sistemą komercinėje, patikimoje, tačiau mokomoje platformoje (mokama už išnaudotus resursus/ laiką). [OpGr09]

Šiame darbe aprašoma 1.6 Eucalyptus versija. Eucalyptus kūrėjai išleidžia daug atnaujinimų/pataisymų, produktas dar nėra pilnai stabilus. Todėl dažniausiai Eucalyptus naudojamas tyrinėjimo tikslais, laikinoms kompiuterijos debesims sukurti, debesų sistemoms išbandyti. [OpGr09]

9. Paketų palyginimas

Bendrai išnagrinėjus abudu nemokamus paketus, galime palygtinti juos.

Funkcionalumas	Eucalyptus	OpenNebula
Grafinė vartotojo sąsaja	-	-
Komandinės eilutės sąsaja	+	+
XEN hypervizoriaus palaikymas	+	+
KVM hypervizoriaus palaikymas	+	+
Amazon EC2 įrankių palaikymas	+	+ (kaip įskiepis)
API palaikymas	+ (Amazon API)	+ (savo)
Nuotolinė virtualių mašinų talpykla	-	-

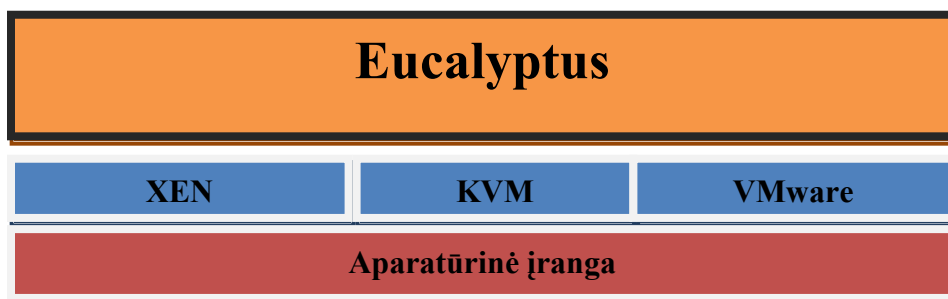
Lentelė 2

Atsisžvelgiant, kad OpenNebula yra skirtas daugiau virtualizacijos infrastruktūroms administruoti, taip pat šis produktas vis dar turintis nemažai klaidų, pilnai nepalaikoma Amazon API sąsaja, todėl nuspręsta toliau detaliai išnagrinėti Eucalyptus paketą, kuris bus naudojamas PaaS paslaugai sukurti.

10. Eucalyptus paketas

10.1 Eucalyptus palaikomos virtualizacijos technologijos

Vienas iš Eucalyptus tikslų, turėti debesų kompiuterijos sistemą nepriklausomai kokia virtualizacijos technologija yra naudojama – tiekti IaaS paslaugą. Pirmoje versijoje Eucalyptus sistema palaikė tikrai XEN hypervaizorių. Vėliau, bendruomenei tobulinant produktą, atsirado KVM, bei VMware hypervaizoriaus palaikymas (Pav. 2). Kol kas Microsoft Hyper-v hypervaizorius nepalaikomas. [DRCGLDEuca09]



Pav. 2

Naudojant XEN hypervaizorių debesies sistema gali užtikrinti:

- Virtualių mašinų migraciją;
- aukštą patikimumą;
- dinaminį resursų paskirstymą;
- energijos vartojimo valdymą.

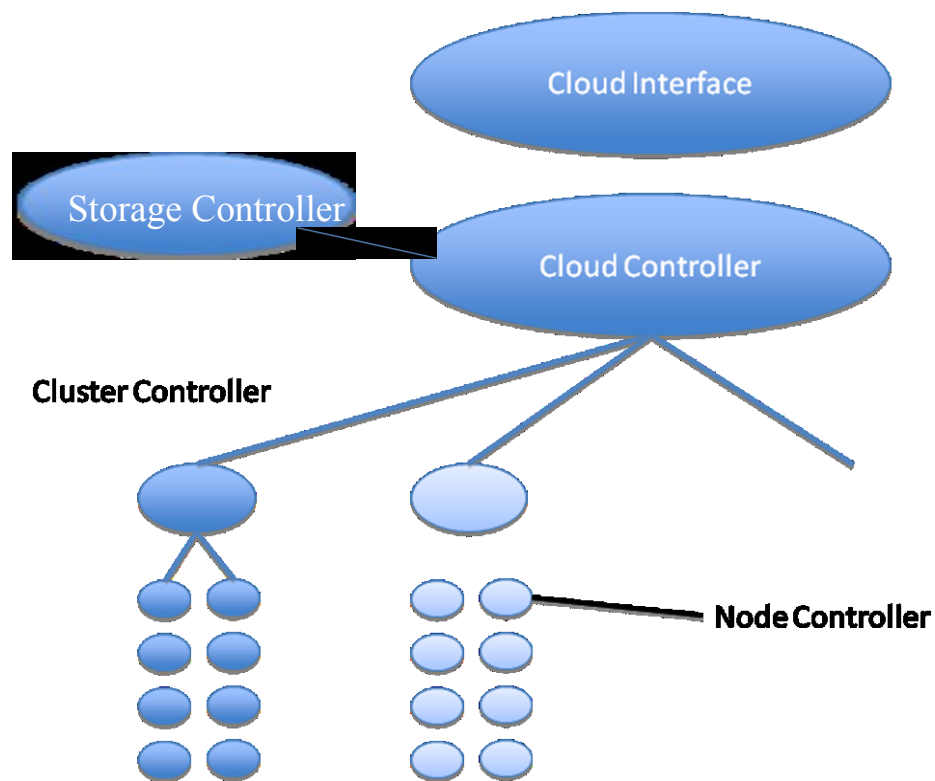
10.2 Eucalyptus architektūra

Pati sistemos architektūra nėra sudėtinga ir praktiškai lengvai valdoma. Naudojama hierarchinė sistemos architektūra. Sistema yra modulinė, kiekvienas modulis turi savo sąsają. Iš esmės sistema vartotojams leidžia startuoti, kontroliuoti, bei išjungti virtualias mašinas, naudojant emuliuotus komercinės paslaugos Amazon EC2 SOAP ir “Query” interfeisus. Tokiu būdu Eucalyptus sistemos vartotojai ir jų aplikacijos gali prieiti prie debesų kompiuterijos paslaugų valdymo, taip pat kaip prieitų dirbdami su Amazon EC2 sistemos paslaugomis. [DRCGLDEuca09]

Kiekvienas aukšto lygio sistemos komponentas-cloud paslauga dirba kaip savarankiškas Web servisas. Tai leidžia kiekvienam komponentui turėti savo, nepriklausomai nuo programavimo kalbos, API interfeisą. API formai pasirinktas WSDL apibrėžimas. Bendravimui naudojamas SOAP protokolas. Taip pat reikia paminėti, kad SOAP užklausos ir atsakymai atitinka bendrus standartus, tai leidžia naudoti visas programavimo kalbas, šiuos standartus palaikomas bibliotekas.

Eucalyptus sistema susideda iš keturių, aukšto lygio komponentų (Pav. 3). Kur kiekvienas komponentas turi savo Web interfeisą, leidžiantį juos suvaldyti:

- Node Controller – skaičiavimo mazgas, kuris kontroliuoja resursus (procesorius, atmintis, tinklo adapteriai, vieta kietajame diske) ir virtualių mašinų darbą fiziniame serveryje. Skaičiavimo mazgo valdiklis duoda komandas hypervaizoriui, kai reikia atlikti tam tikrus veiksmus su virtualiomis mašinomis virtualizacijos sluoksnyje. Node valdiklis gali gauti komandą iš klasterio valdiklio, kad paleistų ar sustabdytų tam tikrą virtualią mašiną. Maksimalus skaičiavimų mazgų skaičius vienam klasteriui yra 256 mazgai;
- Cluster Controller – komunikuoja su klasterio node kontroleriais, renka informaciją apie veikiančias virtualias mašinas, taip pat valdo virtualių mašinų Ethernet tinklą. Dažniausiai išskiriama viena fizinė mašina, dar kitaip pagrindinis klasterio valdiklis (front-end);
- Storage Controller – duomenų talpos valdiklis, naudojamas Amazon S3 interfeisas. Suteikia prieigą prie virtualių mašinų, vartotojų duomenų;
- Cloud Controller – vartotojų ir administravimo valdiklis. Iš šio komponento siunčiamos komandos klasterio valdikliams, jie savo ruožtu skaičiavimo mazgams (node controller).



Pav. 3

10.3 Cloud Controller

Visų virtualizuotų resursų priėjimo, bei valdymo komandos vykdomos iš pagrindinio Cloud valdiklio (angl. Cloud Controller). Realiai Cloud valdiklis, tai web servisų grupė. Kiekvienas servisas gali būti išskirtas pagal savo vaidmenį Cloud sistemoje:

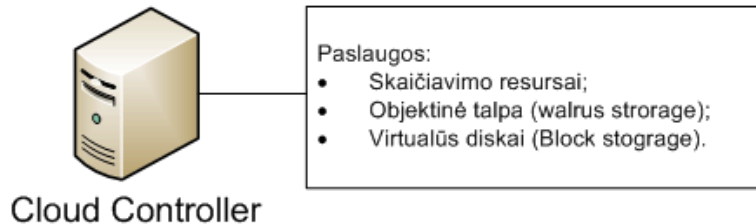
- Resursų servisas – valdo visus Cloud sistemos resursus. Leidžia vartotojams valdyti jiems prieinamas virtualias mašinas ir tinklus. Stebi sistemos komponentų darbą ir virtualius resursus;
- duomenų servisas – valdo visus vartotojų ir sistemos duomenis. Suteikia duomenų konfigūravimo paslaugą, reikalingų resursų paskirstymo paslaugą;
- Interfeiso servisas – suteikia priėjimą prie įvairių sistemos interfeisų. Valdo autentifikaciją ir protokolų transliaciją. Leidžia konfigūruoti sistemą, pasinaudojant sisteminiiais valdymo įrankiais.

10.4 Eucalyptus paslaugos

Šiuo metu Eucalyptus sistemos 1.6.1 versija suteikia tokias tris pagrindines paslaugas:

- Skaičiavimo resursai (virtualios mašinos);
- objektinė talpa (Walrus Storage);
- virtualūs diskai (blokinė talpa – block storage).

Visos paslaugos prieinamos per Cloud Controller valdiklį – visos debesies sistemos valdymo tašką (Pav. 3).



Pav. 4

10.4.2 Skaičiavimo resursai

Skaičiavimo resursus debesyje galime vadinti virtualias mašinas su operacine sistema (angl. instance). Pagal debesies apibrėžimą, tai resursai, kuriuos gauna debesies vartotojas. Kiekviena virtuali mašina priskiriama tam tikram Eucalyptus sistemos vartotojui, jis ją kontroliuoja. Taigi pagrindinė/pirminė Eucalyptus sistemos paslauga yra virtualių mašinų – skaičiavimo resursų pateikimas. Vartotojai nežino, kur fiziškai yra resursai, Eucalyptus sistema pati automatiškai parenka, kokiam fiziniame serveryje paleisti virtualią mašiną.

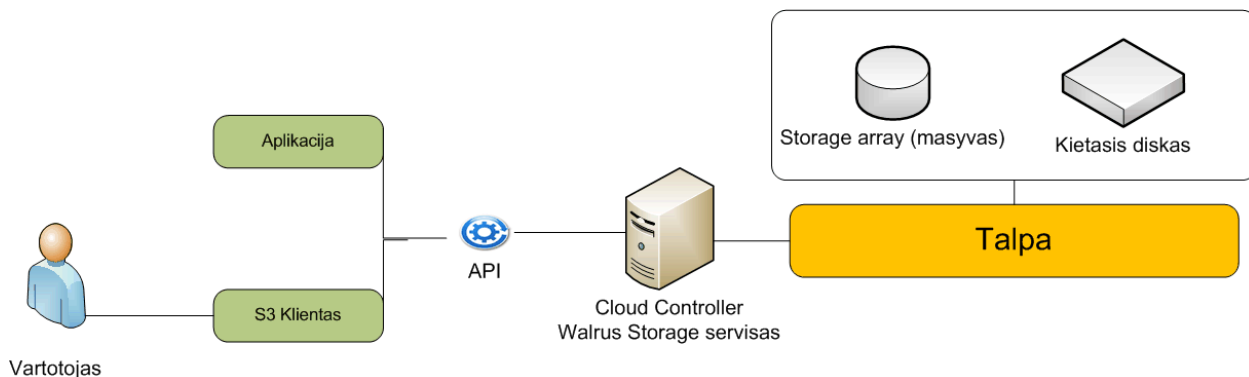
Vartotojams nereikia rūpintis serverio darbo paleidimu, priežiūra ir t.t. Vartotojai gauna priėjimą prie virtualios mašinos ir prie, kai kurių jos valdymo parametrų. Parametrai gali būti: įjungimas, išjungimas, atminties kiekio keitimas, procesorių skaičius padinimas/sumažinimas, ip adreso nustatymas. Realiai tie parametrai negali būti keičiami veikiančiai virtualiai mašinai. Parametrai keičiami pasirinkus tam tikrą virtualios mašinos tipą pvz: vm1-256mb-1CPU. Tai reiškia, kad virtuali mašina gaus 256mb atminties ir vieną virtualų procesorių. [EucaAdG]

Svarbu paminėti, kad virtualios mašinos pakraunamos iš taip vadinamų vaizdų (angl. images), iš vieno vaizdo gali būti pakrautos ir kelios virtualios mašinos. Šiame žingsnyje pasirenkamas mašinos tipas. Vaizde patalpinta operacinės sistemos failų sistema. Išjungus virtualią mašiną failų sistema sunaikinama, todėl, bet kokie duomenys įrašyti į tokią failų sistemą yra prarandami, tokią failų sistemą galima vadinti laikinąja failu sistema. Duomenų saugojimui naudojamos kitos debesies sistemos paslaugos, - tokios kaip virtualūs diskai, objektinė talpa (žiūrėti skyrius: 10.4.4 Block storage, 10.4.3 walrus).

10.4.3 Storage Controller (walrus)

Eucalyptus sistema turi taip vadinamą Walrus duomenų talpos paslaugą, kuri naudoja standartines Web technologijas (Axis2, Mule) ir kurios sąsaja yra iš dalies suderinta su Amazon Simple Storage Service S3 paslauga, ši paslauga gali būti apibūdinama, kaip lengvai valdoma interneto duomenų talpa. Programuotojas nesirūpina, kur ir kaip fiziškai šie duomenys yra talpinami, jo pareiga rūpintis programine įranga, o ne duomenų talpinimu (Pav. 4). Walrus galimybės[EucaUsG]:

- Šioje talpoje gali būti laikomi, bet kokie duomenys;
- tie duomenys gali būti perskaityti/pakeisti daugybę kartų;
- paslaugos tiekėjas užtikrina duomenų prieinamumą:
 - tokia talpa dažniausiai apsaugota nuo duomenų praradimo atvejų;
 - užtikrinama greita skaitymo/rašymo operacijų prieiga;
 - paprastą priėjimą (API);
 - lankstumą (galima kelti mažus, didelius failus);
 - komerciniu požiūriu (amazon s3 paslauga) ekonomiška paslauga - moki tik už tai, ką naudoji;



Pav. 5

Walrus paslauga yra realizuota REST interfeisu (naudojamas HTTP protokolas) ir taip pat SOAP interfeisu. Prie duomenų galima prieiti ir iš išorės, ir iš vidinės cloud infrastruktūros.

Walrus talpoje katalogai (angl. Directory) vadinami “bucket”, failai – objektais (angl. Objects). Kai sukuriamas “bucket”, tuomet jame galima talpinti failus. Pagal Amazon S3 specifikaciją:

- Failų dydis gali būti nuo 1kb iki 5GB;
- failų kiekis kataloguose neribotas;
- kiekvienas failas turi unikalų raktą, pagal kurį failas gali būti pasiekiamas;
- failai gali būti prienami visiems (vieši) arba tik tai tų failų šeimininkui;

Dažniausiai sistemos-aplikacijos dirbančios debesyje ar jo išorėje naudoja S3 API, tam, kad padėtų arba paiimtų duomenis iš virtualios talpos. Programuotojas paprastai gali dirbti su didelio talpumo talpa. Pateikiamas paprastas Java kalbos pavyzdys, kuriame, bandoma sukurti katalogą (pagal walrus - Bucket) (Pav. 5) [EucaUsG].

```

AWSAuthConnection conn =
    new AWSAuthConnection("[aws-access-key-id]", "[aws-
secret-access-key-id]");

Response response = conn.createBucket("[bucket-name]",
null);

if (response.connection.getResponseCode() == 200) {
    // bucket was created
} else {
    // something bad happened

```

Pav. 6

Įprastai tokia virtuali walrus paslauga naudojama:

- Atsarginėms kopijoms laikyti (angl. Backup);
- aplikacijų laikymui, valdymui;
- media duomenų laikymui (vaizdo medžiaga, garso medžiaga);
- dideliems failams laikyti;
- programinės įrangos pateikimui interneto vartotojams.

Kaip ir Amazon S3 komercinė duomenų talpos paslauga, Walrus palaiko lygiagrečių ir nuoseklių priėjimą prie duomenų.

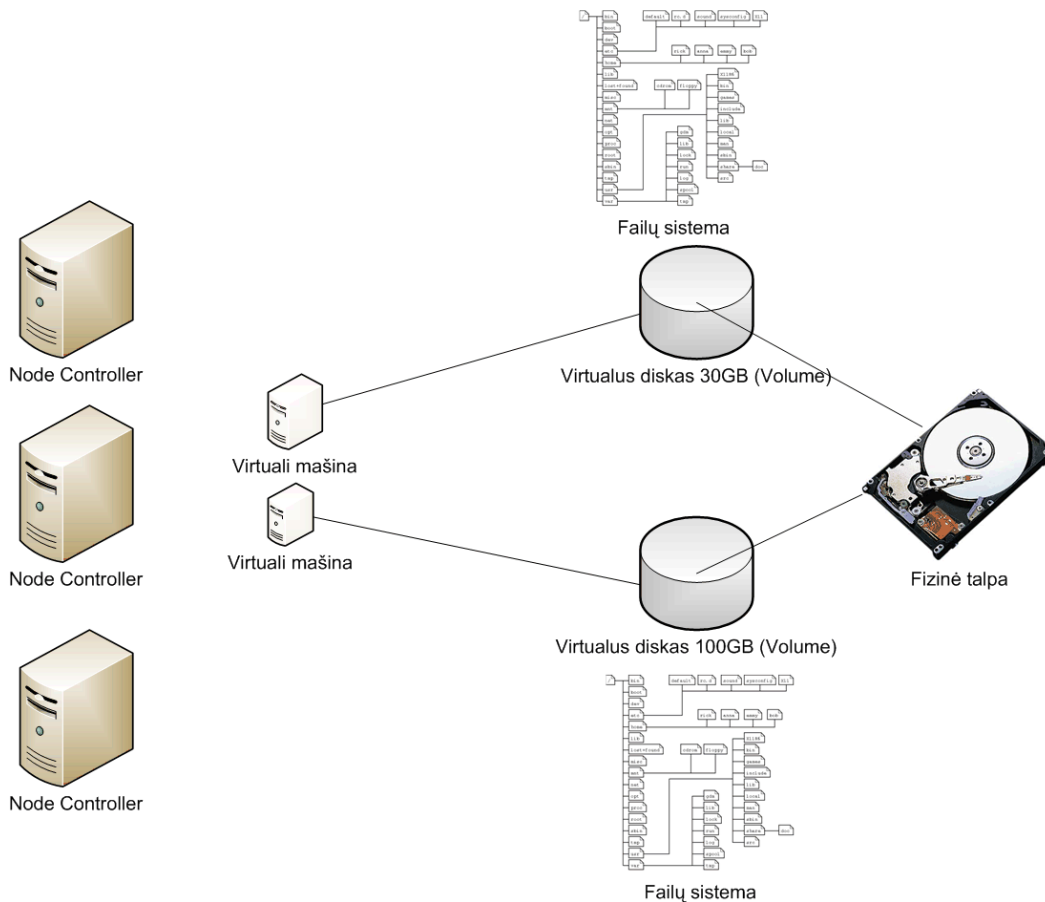
10.4.4 Storage Controller (Block Storage)

Eucalyptus sistema taip pat gali dirbti su block tipo duomenų failų sistemomis, tai dažnai naudojamos failų sistemos, tokiuose įrenginiuose kaip kietieji, optiniai diskai, flash atmintis ir pan.

Eucalyptus sistemoje, tai vadinama Block Storage Cloud tipo paslauga – analogas komercinei Amazon's Elastic Block Store paslaugai. Programiškai, tai apibrėžiama kaip virtualus diskas - talpa (angl.volumes), su viena failų sistema jame. Operacinė sistema prie tokios talpos prieina naudodama failų sistemos procesus. Prireikus tam tikro dydžio duomenų talpos, cloud vartotojai virtualius diskus gali lengvai susikurti. Šie virtualūs diskai gali būti prikabinami (angl.mount) prie virtualios mašinos, operacinė sistema virtualioje mašinoje juos matys kaip savo valdomus diskus su failinėmis sistemomis (Pav. 7) [EucaUsG].

Taip pat daug tokių diskų gali būti prikabinata prie vienos virtualios mašinos. Realiai Eucalyptus sistema sukuria virtualaus disko failą-vaizdą (angl. image) fizinėje talpykloje.

Tuo pačiu galimas tokios vaizdo klonavimas. Tam naudojama funkcija snapshot, kuri gali padaryti virtualaus disko kopiją. Vėliau iš tokios kopijos galima paprastai atsatyti reikiamus duomenis.



Pav. 7

10.5 Eucalyptus duomenų tinklo infrastruktūra

Virtualių mašinų tinklo infrastruktūra (virtualus tinklas) gali būti pritaikyta prie įvairių fizinio tinklo aplinkų. Eucalyptus sistemoje yra 4 aukšto lygio tinklo tipai. Kiekvienas iš jų, su savo konfigūraciniais parametrais, galimybėmis, privalumais, apribojimais yra pritaikomas prie vietinio fizinio cloud telkinio (angl. cluster) struktūros. Cloud administratorius turi pasirinkti vieną iš šių tipų, prieš paleidžiant pagrindinį valdymo mazgą (front-end). Virtualaus tinklo tipai yra šie [EucaUsG]:

- Sisteminis tipas (angl. System mode) – tai paprasčiausias (iš valdymo pusės) tinklo tipas, tačiau suteikiantis mažiausiai galimybių iš tinklos pusės. Naudojant šį tipą, Eucalyptus sistema paprasčiausiai parenka, bet kokį MAC adresą ir jį priskiria virtualios mašinos virtualiam tinklo adapteriui. Virtualios mašinos tinklo adapteris prikabinamas prie realaus, fizinio serverio adapterio naudojant XEN bridge technologiją. Tinklo adresas (angl. IP address) gali būti pateikiamas naudojant automatinį DHCP protokolą.
- Statinis tipas (angl. Static mode) – naudojant šį tinklo tipą, cloud administratorius turi didesnes galimybes kontroliuoti IP adresų paskirstymą virtualioms mašinoms. Administratorius gali sukurti tam tikrą lentelę, kurioje nurodomos MAC ir IP adresų poros. Kai inicializuojama nauja virtuali mašina, jai priskiriama viena iš lentelėje esančių adresų pora. Virtualios mašinos tinklo adapteris prikabinamas prie realaus, fizinio serverio adapterio naudojant XEN bridge technologiją.
- Valdomas tipas (angl. Managed mode) – tai labiausiai valdomas/konfigūruojamas virtualaus tinklo tipas. Tačiau, šis tipas nepatyrusiam vartotojui, gali pasirodyti per daug sudėtingas. Administratorius gali nurodyti skirtingas IP adresų grupes, norėdamas atskirti (iš tinklo pusės) vienas virtualias mašinas nuo kitų virtualių mašinų. Tokiu būdu gali būti sukuriami atskiri, izoliuoti virtualūs tinklai, padidinamas saugumas. Administratorius taip pat gali sukurti grupę viešų IP adresų, kuriuos galima priskirti virtualioms mašinoms.

11. Debesų kompiuterijos sąsaja (angl. Cloud API)

Tam, kad optimaliai išnaudoti cloud teikiamus resursus, programos dirbančios cloud sistemose, turi mokėti valdyti tuos resursus. Komerciniuose debesų kompiuterijos paslaugose, resursai dažniausiai apmokami valandiniais tarifais, todėl svarbu ekonomiškai efektyviai juos suvaldyti. Taip pat ir iš paslaugos tiekėjo pusės svarbu turėti patogius įrankius sistemai valdyti. Tokius veiksmus galima atlikti tiek rankiniu, tiek automatiniu būdu. Stengiamasi kurti tokias sistemas, kurios dirbdamos debesyje šiuos veiksmus atlieka automatiškai. Todėl prie debesų kompiuterijos paslaugų tiekėjų resursų valdymo, programos/bibliotekos prieina per specialią sąsają (Application programming interface) [AmzEC2G] [RCAmz09].

Didžiausi cloud paslaugos tiekėjai dažniausiai turi savo API pvz:

- Amazon EC2 API;
- ElasticHosts API;
- FlexiScale API;
- GoGrid API;
- Sun Cloud APIs.

Kaip pavyzdį galime imti virtualios mašinos valdymo funkcijas, kurios prieinamos pas didžiausius cloud paslaugų tiekėjus:

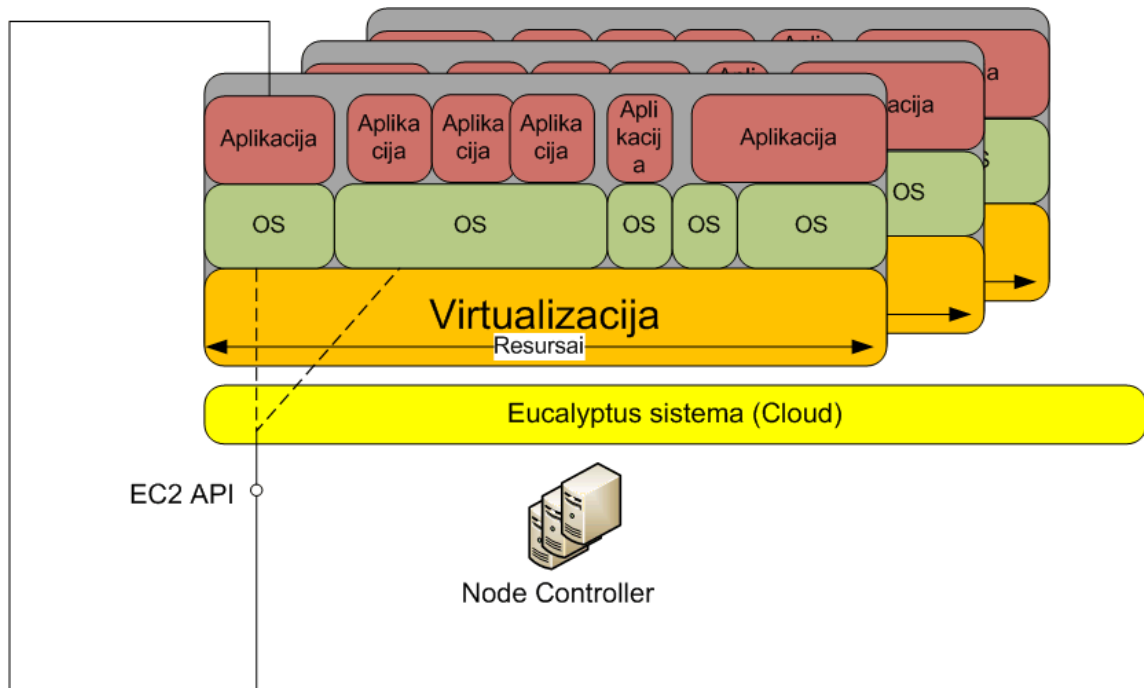
- Paleisti virtualią mašiną (angl. Deploy);
- išjungti virtualią mašiną (angl. Shutdown);
- išjungti/sunaikinti virtualią mašiną (angl. Cancel);
- padaryti virtualios mašinos kopiją išsaugant jos būseną (angl. snapshot);
- atkurti virtualią mašiną iš padarytos kopijos (angl. restore);
- gauti informaciją apie virtualią mašiną – informaciją apie jos naudojamus resursus.

Labiausiai paplitęs Amazon API standartas. Tačiau nėra gerai, kai toks standartas yra valdomas komercinės organizacijos. Todėl buvo įkurta nekomercinė organizacija OCCI (Open Cloud Computing Interface Working Group), dirbanti prie atviro API cloud paslaugų standarto. Tikslas pateikti API specifikacijos reikalavimus, bei pačią atvirą API specifikaciją [OGF10]. Šiuo metu nėra galutinės šio standarto versijos. Tačiau egzistuoja dokumentas, kuris apibrėžia API specifikacijos reikalavimus. Taip pat galima peržiūrėti, dar vis tobulinamą specifikacijos versiją. Tuo tarpu nėra visiškai išbaigtos cloud sistemos atitinkančios specifikacijos reikalavimus. Eucalyptus artimai bendradarbiauja su šia organizacija kuriant specifikaciją. Šiuo metu Eucalyptus sistema nepalaiko pagal šią specifikaciją sukurto API.

11.1 Eucalyptus (Amazon) API

Kaip minėjome, Eucalyptus paslaugos pasiekiamos/valdomos per API (Pav. 8). Tam gali būti naudojamas REST dar kitaip žinomas kaip HTTP Query interfeisas, arba SOAP interfeisas. Šiuo metu labiau paplitęs REST interfeisas.

Programos taip pat gali naudoti jau paruoštas bibliotekas darbui su cloud paslauga. Dažniausiai tokios bibliotekos teikia tokias pagrindines funkcijas: autentifikacijos užklausa, užklausų persiuntimas ir klaidų aptikimas. Čia kalbama tiek apie Eucalyptus, tiek apie amazon cloud paslaugas – abi naudoja tą patį API, taigi ir bibliotekos/aplikacijos teoriškai turėtų dirbti abiejose sistemose. Daug įvairių kodo pavyzdžių ir įvairioms kalboms (python, php, ruby, .NET ...) skirtų bibliotekų, galima rasti Amazon „Sample Code & Libraries“ duomenų bazėje [AmzEC2G].



Pav. 8

11.2 Eucalyptus sistemos valdymo įrankiai (euca2tools)

Skaičiavimo resursų, bei duomenų talpos paslaugas Eucalyptus sistemoje galima valdyti naudojant komandinės eilutės įrankius - Euca2ools. Programiškai, tai įrankiai bendraujantys su web prieigomis Eucalyptus sistemoje naudojant REST/Query API interfeisą, kuris iš dalies atitinka Amazon EC2 ir S3 paslaugų API. Todėl šie įrankiai gali būti naudojami tiek su Amazon cloud paslaugomis, tiek su Eucalyptus paslaugomis. Įrankiai sukurti Python kalba, pasinaudojant Boto biblioteka ir Python kalbos įrankiu M2Crypto. Šie įrankiai suteikia tokią veiksmų aibę:

- SSH autentifikacijos raktų valdymas (pridėti, ištrinti, gauti sąrašą);
- virtualių mašinų valdymas (start, list, stop, reboot, get console output);
- saugumo grupių valdymas;

- virtualių diskų valdymas (attach, list, detach, create, bundle, delete);
- virtualių mašinų vaizdų valdymas (bundle, upload, register, list, deregister);
- IP adresų valdymas IP (allocate, associate, list, release)

Taigi vartotojas, pasinaudodamas šiais įrankiais, iš savo kompiuterio, cloud sistemos virtualios mašinos, rankiniu būdu gali valdyti savo resursus. Šie įrankiai atitinka Amazon pateiktus įrankius tokius kaip (api-tools, ami-tools) ir dažniausiai valdomi panašiai kaip Amazon pateikti įrankiai (parametrų pateikimas, aplinkos kintamieji) [EucaAdG]. Pvz:

Informacija apie veikiančią virtualią mašiną.

(amazon api-tools):

ec2-describe-instances i-ae0bf0c7

(Eucalyptus tools):

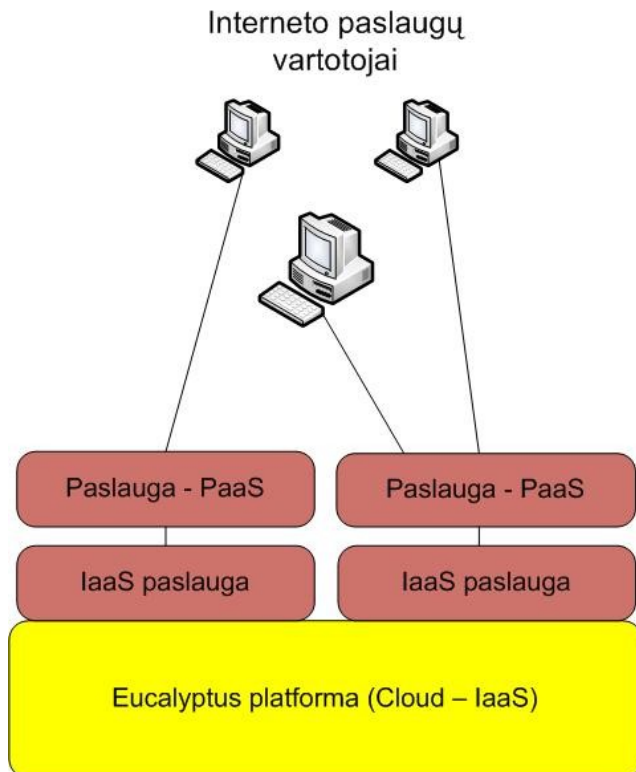
euca-describe-instances i-43035890

Tokie įrankiai nėra labai patogūs. Vartotojui nepratusiam kaip dirbti iš komandinės eilutės gali būti sunku atlikti resursų valdymo veiksmu cloud sistemoje.

12. Paslauga debesyje

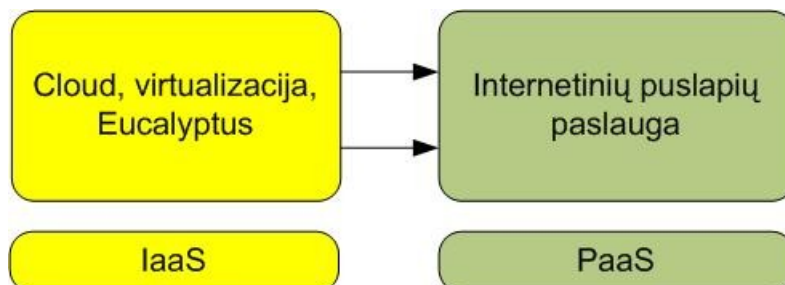
Išnagrinėjus įvairias virtualizacijos technologijas, bei Eucalyptus debesies infrastruktūrą, galima daryti išvadą, kad tai geriausias ir nemokamas sprendimas bandomajai debesies sistemai. Išsiaiškinus, kaip veikia eucalyptus platforma, bei kokius įrankius reikia naudoti, kad suvaldyti tokią sistemą, nuspręsta pabandyti perkelti internetinę paslaugą į debesies platformą.

Naudojant tokią internetinę paslaugą, vartotojui nereikia rūpintis kokios aparatūrinės įrangos jam reikės, vartotojui nereikia mokėti pilną aparatūrinės įrangos kainą, nereikia rūpintis kitais duomenų centro procesais (administravimas, aušinimas, sisteminių problemų sprendimas ir kt.) Vartotojo paslauga dirbs debesyje (angl. cloud). Tuo pačiu išsiaiškinsim, kaip galėtų būti perkeltos ir kitos, panašios, interneto paslaugos į debesies platformą.



Pav. 9

Norint aiškiau suprasti, kas yra debesies paslauga, kas yra internetinė paslauga, reikia apibrėžti tam tikrus debesies terminus. Debesies infrastruktūrą (mūsų atveju Eucalyptus platforma) vadinsime infrastruktūra kaip paslauga (angl. IaaS - Infrastructure as a Service) (žr. 2.1 skyrių). Debesies vartotojas, tai subjektas, kuris naudojasi IaaS paslauga. Iš tiesų IaaS vartotojas, tai internetinės paslaugos tiekėjas, kuris pateikia internetinę paslaugą (mūsų atveju internetinių puslapių talpinimo paslauga) šios internetinės/PaaS paslaugos vartotojams. Pagal debesies paslaugų apibrėžimus (žr. 2 skyrių), galime teigti, kad internetinių puslapių talpinimo paslauga tai platforma kaip paslauga (angl. PaaS - Platform as a service). Taigi turime tokią struktūrą:



Pav. 10

Pabandysime perkelti paprastą internetinių paslaugų paslaugą į debesies platformą t.y.sukursime PaaS paslaugą.

13.LAMP paslauga

Kaip pavyzdį pabandysime įgyvendinti interneto puslapių talpinimo paslaugą (angl. web hosting) eucalyptus sistemoje. Tradicinė interneto puslapių talpinimo paslauga (LAMP) siūloma su tokiomis funkcijomis (LAMP):

- L – Linux operacinė sistema;
- A – Apache web serveris;
- M – MySQL duomenų bazė;
- P – PHP plačiai paplitusi dinaminė interpretuojama programavimo kalba (angl. Hypertext Preprocessor) kartais, raidė P gali reikšti ir kitas web programavimo kalbas Perl arb Python.

Tokių web serverio funkcijų dažniausiai pakanka paprastiems, nedideliems interneto puslapiams. Visos talpinimo paslaugos funkcijos yra sudaromos iš nemokamų produktų, kuriuos galima naudoti komerciniams tikslams. Eucalyptus platformoje, tai būtų virtuali mašina, su įdiegtomis šiomis funkcijomis.

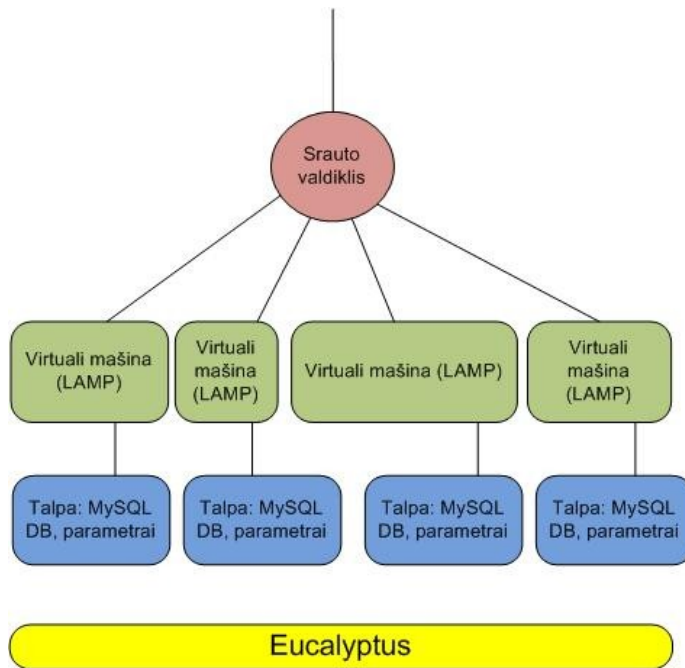
Internete standartinės LAMP paslaugos yra labai populiarios. Tai dažniausiai būna bendro naudojimo web talpinimo paslaugos, kai vartotojų puslapiai dalinasi vienu fiziniu serveriu, paskirstant resursus operacinės sistemos lygyje. Taip pat ypač paplitę virtualūs dedikuoti serveriai. Dažniausiai tam naudojama virtualizacija, kuri leidžia viename fiziniame serveryje turėti kelis ar daugiau virtualių serverių (virtualūs serveriai - virtualios mašinos). Vartotojas tokiuose serveriuose viską iš operacinės sistemos pusės gali administruoti savo nuožiūra. Virtualizacijos atžvilgiu tai labai panašu į debesies platformą, kurioje taip pat resursai yra virtualizuoti, tačiau naudojant virtualius dedikuotus serverius, vartotojas moka už juos mėnesinį mokestį. Debesies atveju, vartotojas už resursus moka, tik tiek, kiek juos naudoja (pvz: 10 valandų) t.y. vartotojas gali ir pats administruoti virtualius resursus, nenaudoti jų, kai jie jam nereikalingi. O tuo pačiu naudotis debesies platformos teikiamomis galimybėmis.

13.1 Internetinių puslapių talpinimo paslauga

Eucalyptus sistema suteikia vartotojui virtualius resursus. Mūsų atveju tai bus virtualios mašinos, kuriose bus įdiegtos pagrindinės LAMP funkcijos. Toliau tokia virtuali mašina bus atiduodama galutiniam vartotojui, kuris ir naudosis PaaS paslauga. Tokios paslaugos įjungimas galutiniam vartotojui turi veikti be žmogaus įsikišimo t.y. reikia įgyvendinti automatinį procesą, kuris reikiamus resursus galutiniam vartotojui paruošia automatiškai naudodamas debesies platformos valdymo įrankius.

Kad gerai suprastume, kaip veiks internetinių puslapių talpinimo paslauga debesyje, reikia apsibrėžti sistemos dalis:

- Virtualios mašinos (virtualūs resursai);
- talpa (žiūrėti blokinio tipo talpa);
- valdymas;
- srauto valdiklis;



Pav. 11

(virtualios mašinos pavaizduotos skirtingų dydžių, tai parodo, galingesnius ar mažesnio galingumo virtualius resursus.)

13.2 LAMP virtuali mašina

Standartiniai virtualių mašinų vaizdai (angl. Eucalyptus Machine Image(EMI)), kurie yra naudojami Eucalyptus platformoje, neturi jokios papildomos programinės įrangos. Tam, kad paruošti virtualios mašinos vaizdą, mums reikės sudiegti LAMP funkcijas į vaizdą, kad ateityje paleidus naujas virtualias mašinas naudojančias šį vaizdą, nereikėtų diegti LAMP funkcijų iš naujo. Tam naudosime specialius programinės įrangos instaliacinius skriptus, kurie automatiškai įdiegs mums reikalingas funkcijas, tuo pačiu tai bus pati naujausia stabili programinė įranga. Skriptų pavyzdžiai, virtualios mašinos vaizdo su CentOS operacine sistema, paruošimui darbui su LAMP (priedas). Tokie skriptai vėliau gali būti modifikuojami, išskylant papildomų bibliotekų poreikiui. Instaliavus reikiamą programinę įrangą, paruoštas vaizdas gali būti įtrauktas į Eucalyptus platformą, bei paleistas kaip virtuali mašina (serveris).

Taip pat reikia paminėti, kad Eucalyptus sistemoje, virtualios mašinos (angl. instance) duomenų talpa nėra pastovi (žr. 10.4.2 skyrių). Tam, kad kiekvieną kartą perkrovus virtualią mašiną, vartotojas galėtų prieiti prie, prieš tai buvusių duomenų, naudojama nekintanti/pastovi talpa. Tam pasirinkta blokinio tipo talpa, kurioje bus laikomi reikalingi duomenys. Tokiu atveju, jeigu ateityje prisireiks galingesnės virtualios mašinos, pvz: atlikti procesoriaus galios padidinimą. Startavus kitai (su didesne procesoriaus galia) virtualiai mašinai, tiesiog prikabinti, prie naujos mašinos, anksčiau naudotą blokinio tipo talpą su joje esamais vartotojo duomenimis. Tuo pačiu, galima lengvai įgyvendinti, tokių duomenų replikavimą, naudojant snapshot funkciją (žr. 10.4.4 skyrių).

14. Paslaugos duomenys

Tinklapių talpinimo atveju, reikia apibrėžti, kokie duomenys yra nuolatiniai t.y. vartotojo/sistemos duomenys, kurie turi likti talpykloje, kol jų neištrynė pats vartotojas. Duomenis suskirstome į tokias grupes:

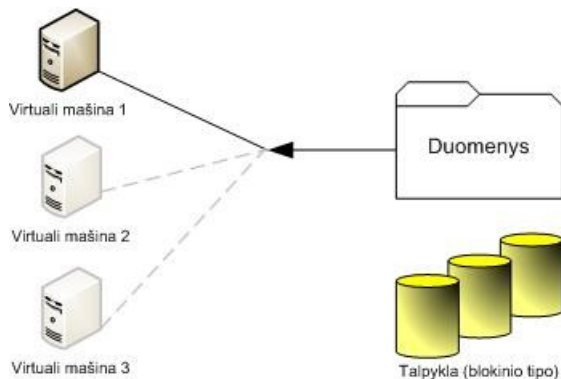
- Servisų (Apache, php, MySQL) konfigūraciniai duomenys;
- aplikacijos t.y. WWW tinklapiai, php aplikacijos;
- MySQL duomenų bazė;
- papildomi duomenys;

Tokia talpa sukuriama, pasinaudojus blokinio tipo talpos paslauga eucalyptus platformoje. Tokioje talpoje laikyti duomenis yra naudingiau, nes:

- Talpa atskirta nuo virtualios mašinos leidžia atkurti duomenis nepriklausomai ar virtuali mašina „nulūžo“, ar buvo išjungta, ištrinta iš sistemos;
- tokia talpa gali būti prijungta prie kitos virtualios mašinos;
- blokine talpa yra atskira debesies platformos paslauga, jina gali būti dubliuota, veikti aukšto patikimumo įrangoje ir pn.
- tokia talpa gali naudoti naujausias technologijas, kurios leidžia pasiekti labai didelius greičius skaitymo/rašymo operacijose;
- talpa gali būti plečiama iki 1TB dydžio;
- lengva padaryti atsarginę kopiją.

Šiems veiksmams atlikti, gali būti naudojami administravimo įrankiai arba tam tikras valdymo modulis, kuris automatiškai, prijungia naują talpą, prie virtualios mašinos, pasinaudodamas Eucalyptus sąsaja, kuri leidžia valdyti tokią talpą (žr. 11.1).

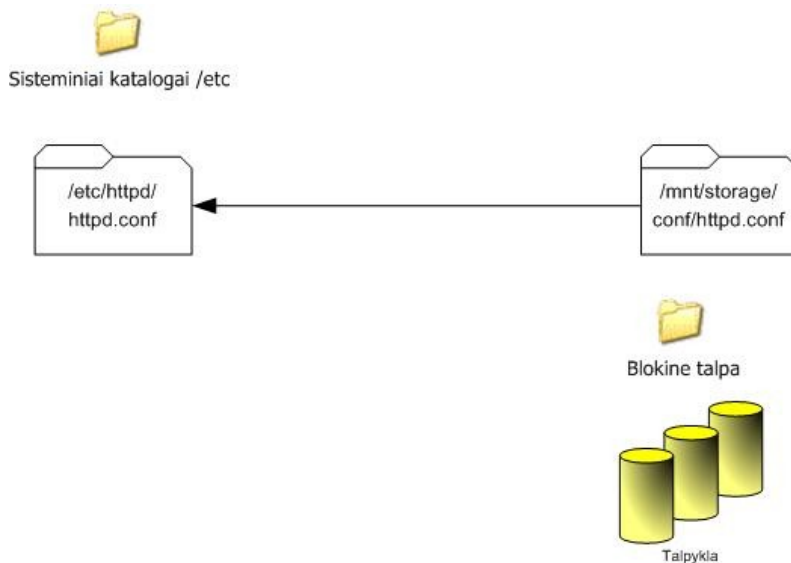
Kad veiktų viena bendra PaaS sistema, kurioje yra daug vartotojų, reikės taip pat išsaugoti kokiam vartotojui, kokia talpa priklauso, kokia talpa prie kokios virtualios mašinos priklauso, visą šią informaciją laikysime duomenų bazėje (Pav. 12).



Pav. 12

14.1 Servisų konfigūraciniai duomenys

Visų pagrindinių tinklapio talpinimo servisų konfigūracija taip pat turi likti nekintanti perkrovus virtualią mašiną, tam naudojama blokinio tipo talpa. Apache, php, mysql servisų konfigūracijos laikomos atskirame kataloge, į kurį rodo simbolinė (angl. symbolic link) nuorodą iš sisteminių katalogų (Pav. 13).



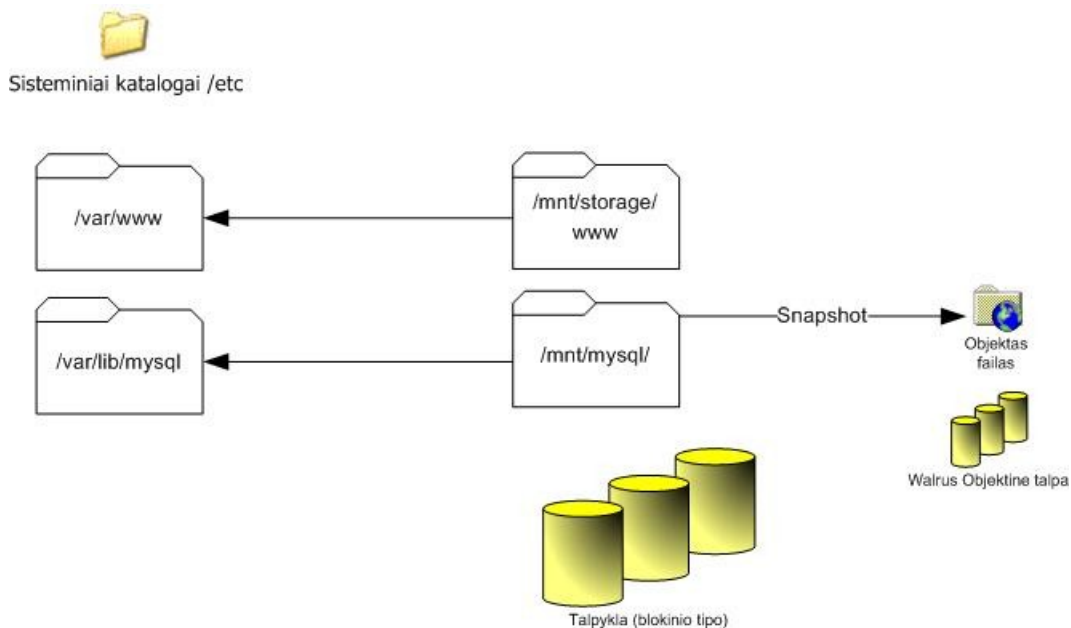
Pav. 13

14.2 Aplikacijų duomenys

Tinklapių talpinimo paslaugoje, aplikacijos, tai pačio tinklapio programinis kodas (HTML, PHP, CSS). Tokius duomenis taip pat svarbu išsaugoti. Todėl juos dedame į blokinę talpą, kurioje tie duomenys nepražus, perkrovus virtualią mašiną. O kai tai būtina, - galimas greitas duomenų perkėlimas į kitą virtualią mašiną. Pasinaudojant simboliu nuoroda, visos aplikacijos pasiekiamos iš blokinės talpos. (Pav. 14)

14.3 MySQL duomenų bazė

Duomenų bazės duomenys mums yra svarbūs ir reikalingi po virtualios mašinos perkrovimo. Jie taip pat patalpinami blokinėje talpoje (priedas). Kaip ir paminėta, atsarginės kopijos gali būti daromos visai blokinei talpai. Tačiau, kad atskirti mysql duomenų bazės duomenis nuo kitų, visus duomenis dedame į atskirą blokinę talpą (viena virtuali mašina gali turėti 5 virtualius diskus). Tam sukuriama atskira blokinės talpos diskas ir prikabinamas prie virtualios mašinos (Pav. 14).



Pav. 14

Šiuo atveju, norint padaryti duomenų bazės atsarginę kopiją užtenka įvykdyti snapshot funkciją naujai talpai, kuri prijungta prie bendros failų sistemos adresu „/mnt/mysql“. Kaip žinome snapshot funkcija pagamina vieną disko vaizdą (angl. image) – vieną failą, kuris toliau gali būti patalpinimas, kitoje debesies IaaS paslaugoje - walrus objektinėje talpoje (žr. 10.4.3 skyrių).

14.4 Papildomi duomenys

Visi kiti papildomi duomenys taip pat gali būti laikomi blokinėje talpoje, tam vartotojui reikia susikurti reikalingas simbolines nuorodas į blokinės talpos katalogą.

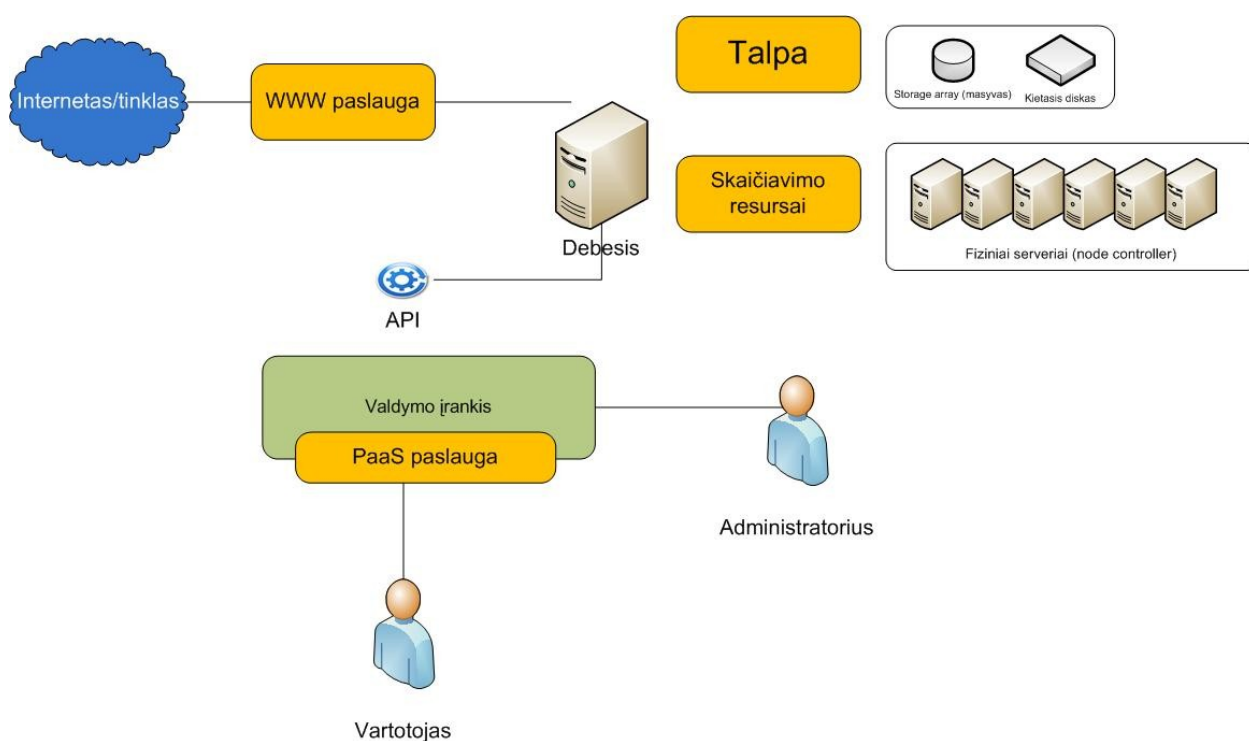
15. Paslaugos stebėjimas

Tiek iš administravimo pusės, tiek iš paslaugos vartotojo pusės, reikalingas paslaugos stebėjimas ar neįvyko sistemos „lūžimas“, ar nėra didelio apkrovimo ir t.t. Į tai įeina: procesų stebėjimas, virtualių resursų stebėjimas (virtualus procesorius, virtuali atmintis, virtuali talpa). Paslaugos stebėjimui įgyvendinti pasirinktas Munin įrankis. Kuris veikia Serverio/kliento režimu. Kiekvieno vartotojo virtuali mašina (LAMP serveris) turi savyje įdiegtą Munin klientą (tai galima atlikti paruošiant LAMP serverių vaizdus žr. 13.2), kuris siunčia visus reikalingus duomenis į pagrindinį Munin serverį. Paslaugos tiekėjas gali nustatyti ką vartotojas gali stebėti, ko negali.

Informacinės diagramos gali būti pateikiamos į kliento valdymo įrankio grafinę prieigą. Pavyzdžiai kaip atrodo Munin grafikai (3 priedas) [MuninT].

16. Sistemos valdymo įrankis

Atsižvelgiant į tai, kad paslauga vienu metu tiekama dideliame skaičiuje vartotojų, reikia turėti įrankį, kuris automatizuotų, bei leistų suvaldyti tokią paslaugą, kontroliuotų visą sistemą. Tam tikslui apibrėžiame, sistemos valdymo modulį. Kuris turi valdyti paslaugą (valdyti LAMP/paslaugos serverį, padidinti/sumažinti serverio galią, padidinti talpą), ir taip pat patalpinti visą informaciją apie vartotojo duomenis, virtualias mašinas į pagrindinę sistemos duomenų bazę, esant reikalui ją keisti. Bendroje infrastruktūroje šis valdymo įrankis atliktų PaaS paslaugos valdymą.



Pav. 15

Toks PaaS paslaugos valdymo modulis, suteiktų galimybę vartotojams valdyti naudojamą paslaugą, o administratorius galėtų prižiūrėti/stebėti PaaS sistemą (Pav. 15).

16.1 Sistemos valdymo modulis

Sistemos valdymas galimas naudojant Eucalyptus sąsają, kuria naudojantis galima atlikti tokius veiksmus:

- Virtualių mašinų valdymas;
- objektinės duomenų talpos valdymas;
- virtualių diskų valdymas;
- IP adresų valdymas.

Šiam moduliui sukurti pasirinkta PHP programavimo kalba [PHPDev10], leidžianti lengvai programuoti tinklalapių aplikacijas. Šiuo metu internete yra keletas php bibliotekų/klasijų palaikančių Amazon API. Resursų valdymo modulio sukūrimui pasirinkta pilnai palaikanti Eucalyptus platformą CloudFusion PHP biblioteka.

CloudFusion 2.5 versija pilnai palaiko (Eucalyptus platformoje) [CIFu10]:

- Virtualių mašinų valdymą;
- objektinės talpos valdymą (walrus);
- virtualių diskų valdymą.

16.2 Sistemos valdymo modulio funkcijos

16.2.1 Tinklalapio serverių valdymas

Sistema turi leisti valdyti virtualius vartotojo serverius, jiems paruoštus LAMP vaizdus (angl. images). Turi būti realizuotos tokios funkcijos, kai kurios iš šių funkcijų gali būti prieinamos ir PaaS paslaugos vartotojams, visos funkcijos gali būti pateiktos sistemos administratoriui:

- Startuoti virtualų serverį;
- gauti veikiančių virtualių serverių sąrašą;
- peržiūrėti detalesnę informaciją apie kiekvieną virtualų serverį;
- stabdyti virtualų serverį;
- perkrauti virtualų serverį;
- valdyti duomenų talpų priklausomumą tam tikriems virtualiems serveriams;
- valdyti tinklo adresų (angl. IP address) priskyrimą virtualiems serveriams.

Tokios funkcijos kaip „Peržiūrėti detalesnę informaciją apie kiekvieną virtualų serverį“ gali susidėti ne tik iš informacijos, kurią galima gauti pasinaudojant Eucalyptus sąsaja, bet ir informaciją iš Munin įrankio apie serverio apkrovimus.

16.2.2 Objektinės talpos valdymas (walrus storage)

Sistema turi leisti matyti kokio dydžio talpa prisikirta vartotojui, kiek bendrai objektų yra duomenų saugykloje (failų), kiek failų yra kataloguose (angl. Buckets), sistema turi rodyti koks yra vieno katalogo dydis. Kartu turi atlikti šias funkcijas:

- Vienoje vietoje rodyti visus katalogus ir informaciją apie juos;
- leisti sukurti/ištrinti katalogą;
- leisti peržiūrėti kiekvieno katalogo turinį;
- leisti vartotojui nusiųsti į saugyklą failą ir priskirti vienam iš katalogų;
- leisti ištrinti esančius failus;
- leisti parsisiųsti failus iš katalogų;
- Sugeneruoti viešo priėjimo nuorodą.

Objektinės talpos valdymas pilnai gali būti prieinamas ir paprastam PaaS vartotojui, nes tokioje talpoje laikomi tikrai jo duomenys. Tokia paslauga gali būti kaip priedas prie pilnos www serverio paslaugos, nes nevisada tokios talpos reikia, vietoj objektinės talpos galima naudoti blokinę talpą.

16.2.3 Virtualių diskų valdymas

Virtualūs diskai (žr. 10.4.4). Sistemos valdymo modulis turi leisti atlikti šiuos veiksmus su virtualiais diskais:

- Pateikti informaciją apie esamus virtualius diskus;
- leisti sukurti naują diską;
- leisti ištrinti valdomus diskus;
- leisti priskirti (angl.mount) reikalingus diskus, tam tikroms virtualioms mašinoms;
- surinkti informaciją kokie diskai, kokiems virtualiems serveriams priklauso;
- leisti daryti virtualių diskų atsargines kopijas.

Virtualius diskus kontroliuoja sistemos valdymo modulis. Reikia nepamiršti, kad sistemos duomenų bazėje turi būti informacija, kur kokia blokinė talpa, priklauso kuriam virtualiam serveriui. Iš PaaS paslaugos vartotojo pusės sistemos valdymo modulis turi leisti daryti virtualių diskų atsargines kopijas, kas leidžia pačiam vartotojui nuspręsti, kada atliekamas atsarginės kopijos darymas.

16.3 Sistemos valdymas

Apibrėžiame, kokias funkcijas gali atlikti paprastas PaaS paslaugos vartotojas ir kokias administratorius (Lentelė 3).

Funkcija	Vartotojas	Administratorius
Startuoti virtualų serverį	+	+
Stabdyti virtualų serverį	+	+
Peržiūrėti detalesnę informaciją apie kiekvieną virtualų serverį	+ (iš dalies)	+
Perkrauti virtualų serverį	+	+
Gauti veikiančių virtualių serverių sąrašą	+ (informacija tikrai apie vartotojo serverius)	+ (informacija apie visų vartotojų serverius)
Valdyti duomenų talpų priklausomumą tam tikriems virtualiems serveriams;		+
Valdyti tinklo adresų (anlg. IP address) priskyrimą virtualiems serveriams	+	+
Vienoje vietoje rodyti visus katalogus ir informaciją apie juos (objektinėje talpoje)	+	+
Leisti sukurti/ištrinti katalogą	+	+
Leisti peržiūrėti kiekvieno katalogo turinį	+	+

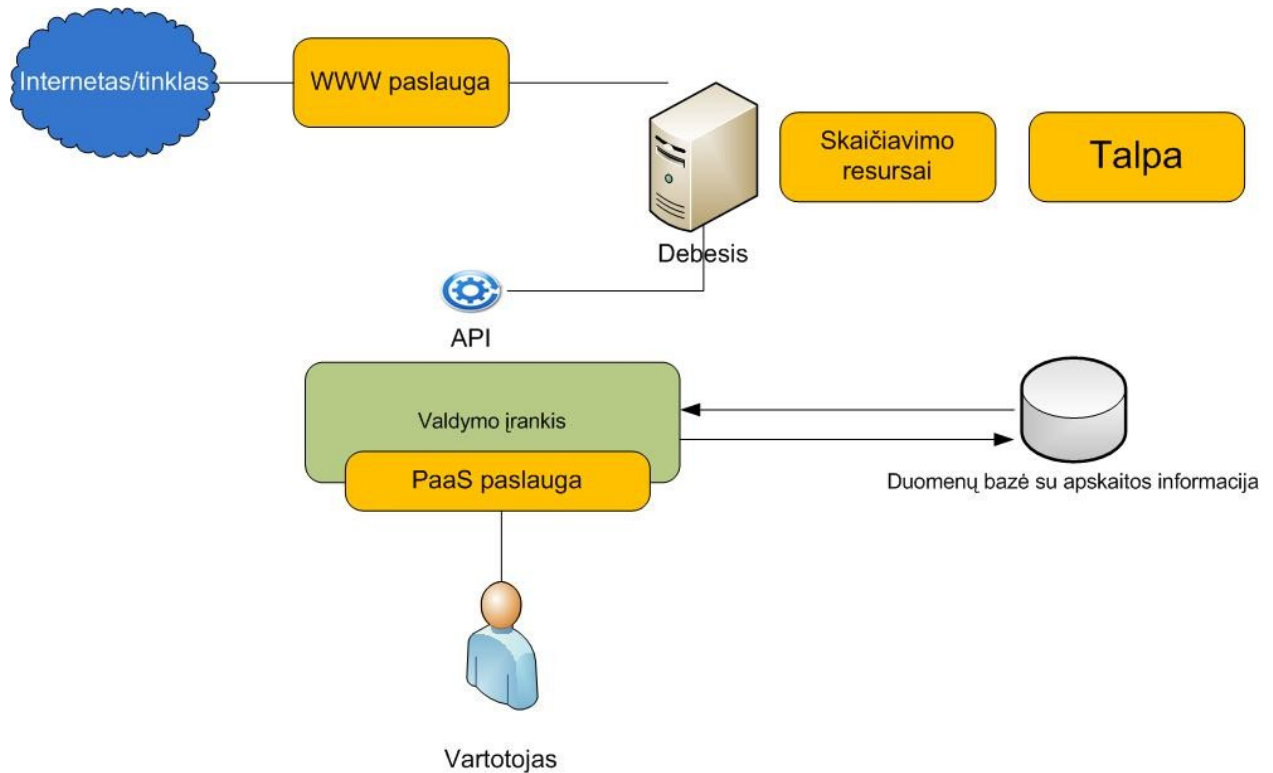
Leisti vartotojui nusiųsti į saugyklą failą ir priskirti vienam iš katalogų		
Leisti ištrinti esančius failus	+	+
Leisti parsisiųsti failus iš katalogų	+	+
Sugeneruoti viešo priejimo nuorodą	+	+
Pateikti informaciją apie esamus virtualius diskus		+
Leisti sukurti naują diską		+
Leisti ištrinti diskus		+
Leisti priskirti (anlg.mount) reikalingus diskus, tam tikriems virtualiems serveriams	+ (tiktai to vartotojo virtualiems serveriams)	+
Leisti daryti virtualių diskų atsargines kopijas	+	+

Lentelė 3 Sistemos vartotojo ir administratoriaus funkcijos

Vartotojo, administratoriaus valdymas prieinamas per Web sąsają – vartotojo kompiuteryje įdiegtoje naršyklėje. Informacijos pateikimas nėra pati svarbiausia šio įrankio dalis, tai gali būti paprastas tekstas, paveikslėliai ir nesudėtingas html kodas. Svarbiausia, kad reikalingos funkcijos veiktų, tiek kiek leidžia Eucalyptus API, bei sistemoje įdiegti įrankiai.

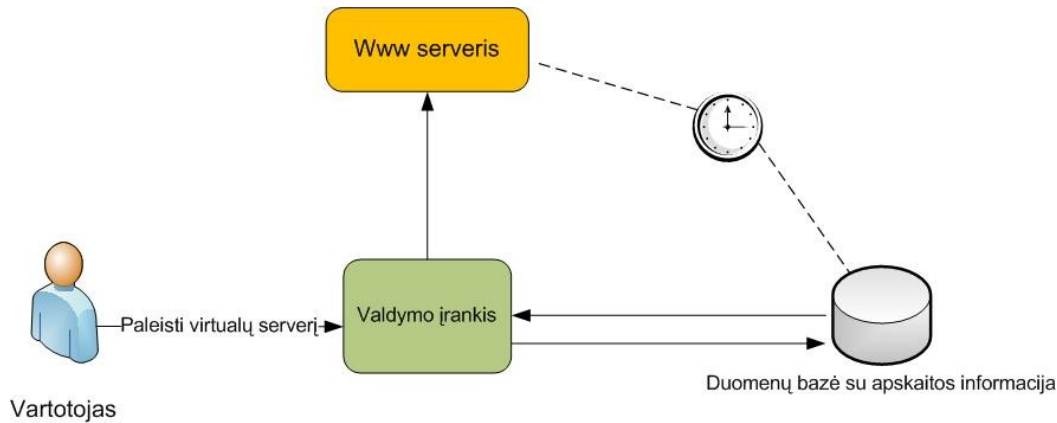
16.4 Apskaita (angl. accounting)

Tokiai PaaS paslaugai, kaip internetinių puslapių talpinimas (angl. web hosting) reikalinga detali apskaita, kuri parodytų kokių resursų, kiek kas sunaudoja. Vienas iš debesies privalumų, mokėk už tiek, kiek naudoji. Todėl ir įvairias PaaS paslaugas galima būtų apmokestinti, tik už tą dalį, kurią vartotojas naudoja. Internetinių puslapių talpinimo paslaugos atveju, tai galėtų būti mokestis už įvairaus galingumo virtualias mašinas (serveris), už duomenų talpą (mokestis už gigabaitus), už veikiančio serverio laiką ir panašiai.



Pav. 16

Apskaitą turi sudarinėti pati sistema (Pav. 16). Paprasčiausiai paleidus virtualų serverį, sistema turi įrašyti informaciją, kad toks serveris buvo paleistas tam tikru laiku. Išjungus serverį sistema turi atžymėti, kada serveris buvo sustabdytas (Pav. 17). Tokiai ir panašiai informacijai reikalinga duomenų bazė, kurioje bus saugomi visų vartotojų apskaita.

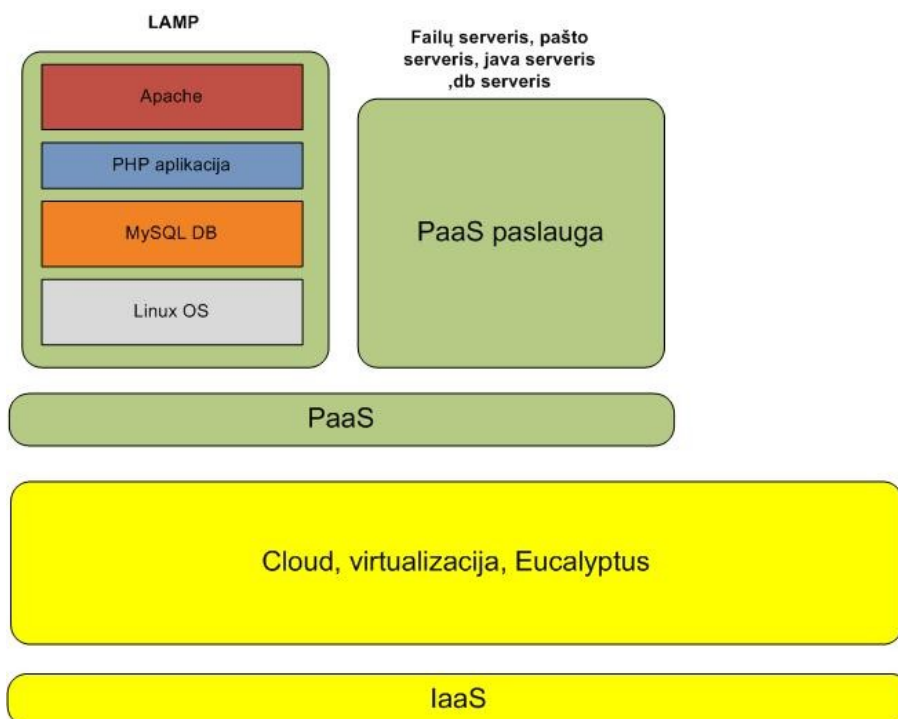


Pav. 17

Eucalyptus walrus objektinė talpa neturi jokio apskaitos mechanizmo, failai gali būti ištrinami/įdedami į talpą naudojantis tiesiogiai eucalyptus sąsaja. Už naudojamą talpą niekas neatsako ir jokių duomenų apie tai nepateikia, todėl nuspręsta nedaryti apskaitos vartotojams, kurie naudojami walrus talpa. Jeigu yra didelis poreikis šioje vietoje atlikti apskaitą (pvz: už sunaudotus megabaitus), tuo metu PaaS paslaugos vartotojai turėtų nenaudoti tiesiogiai eucalyptus sąsajos, o duomenų įkėlimą atlikti turėtų PaaS paslaugos valdymo sistema.

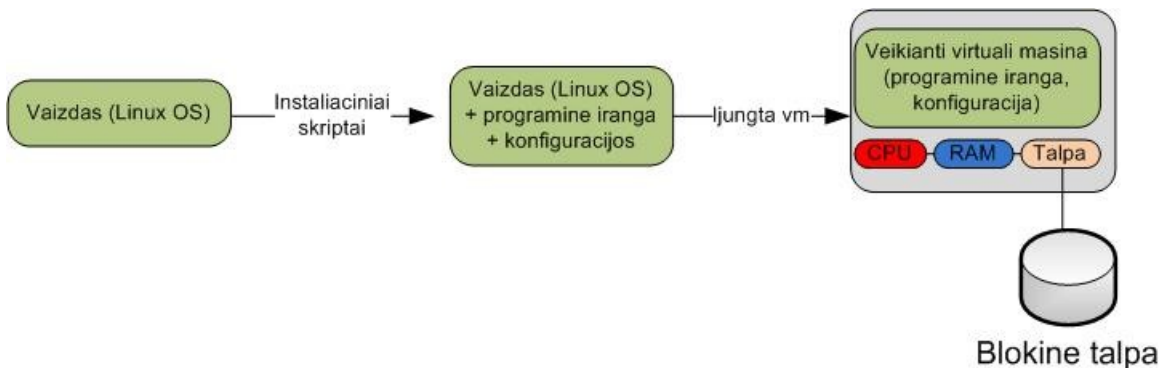
17.Paslaugos perkėlimas į debesį

Pateikiame pavyzdį, kaip įgyvendinti , ir kaip atrodytų LAMP paslauga debesyje, kurio pagrindas yra nemokama platforma Eucalyptus (pav. 18).



Pav. 18 PaaS paslauga Eucalyptus platformoje

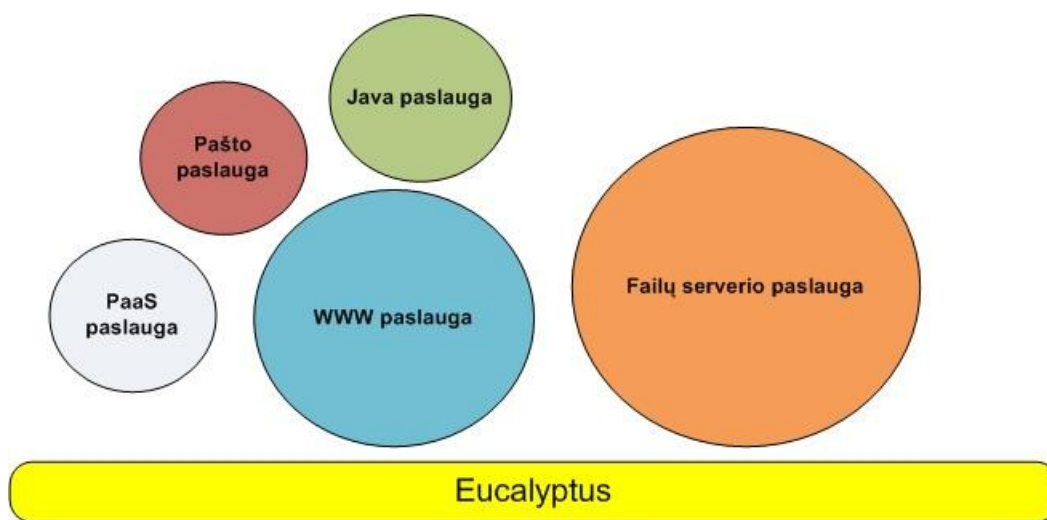
Rezimuojant galima teigti, kad paruošti tam tikrą paslaugą debesyje, mums pirmiausia reikia apsibrėžti, kokio virtualios mašinos vaizdo reikės. Visais atvejais naudojama Linux operacinė sistema, Eucalyptus platforma, vis dar nepalaiko virtualių mašinų su Windows operacine sistema.



Pav. 19 Diegiamos serverio funkcijos (paruošiamos PaaS paslaugos funkcijos)

Vaizde naudojame Linux operacinę sistemą, toliau pasinaudodami instaliaciniais skriptais, sudedame visą reikalingą programinę įrangą. Šioje vietoje nustatoma, kokias paslaugas teiks virtuali mašina (Pav. 19). Prieš tai nagrinėtame pavyzdyje buvo instaliuojama visa programinė įranga reikalinga LAMP funkcijoms. Modifikuojant instaliacinius skriptus galime nustatyti, kad virtualioje mašinoje būtų pvz failų serveriui reikalinga programinė įranga arba pašto serveriui reikalinga įranga. Paruošiant pašto serverio vaizdą, reikėtų atlikti daugiau konfigūracijos darbų. Tam galima naudoti tuos pačius instaliacinius skriptus. Tuo tarpu reikėtų nepamiršti, kad visa informacija ir konfigūracijos failai turi būti padėti į blokinę talpą, kurioje jie išlieka nežiūrint į tai kaip veikia virtuali mašina (žr. 14).

Bendroje debesies infrastruktūroje galime turėti įvairių PaaS paslaugų, kurios viena kitos gali ir neįtakoti, nematyti, bet taip pat gali viena kitą papildyti (Pav. 20).



Pav. 20 Vienoje debesies platformoje galime turėti daug PaaS paslaugų

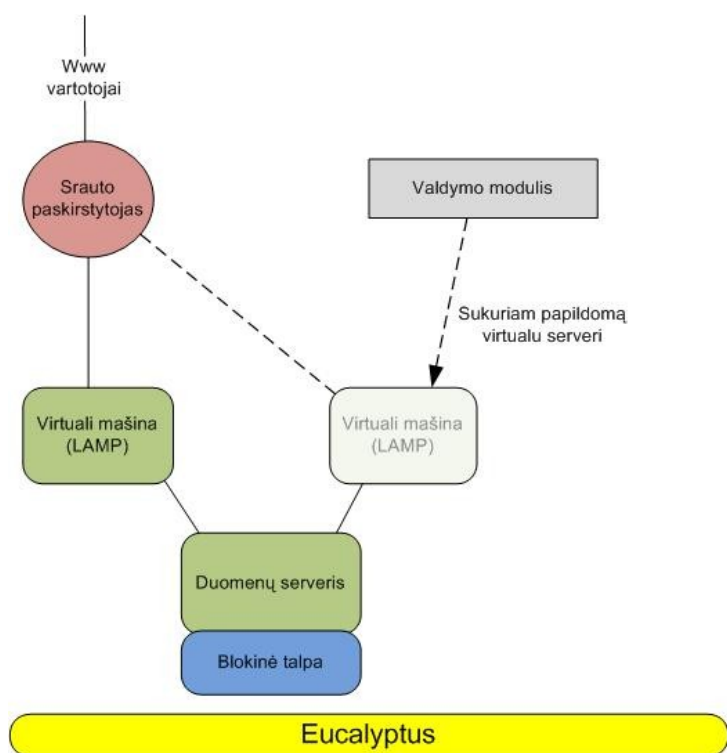
Kad suvaldyti tokias paslaugas, taip pat reikia duomenų bazės, kur laikoma papildoma informacija apie virtualiu serverius, jų vartotojus, duomenų talpos naudojimą, apskaitą.

18. Dinaminis plečiamumas pagal poreikį

Prieš tai aptartas www paslaugos tiekimas apsiriboja tikrai virtualių mašinų (virtualių serverių) pateikimu galutiniam vartotojui. Tačiau didėjant resursų poreikiui, vartotojai gali norėti turėti ne vieną virtualų serverį (vienai aplikacijai), bet jų telkinį, jeigu tai leidžia tiekiamos paslaugos programinė įranga. Žinant, kad debesis naudoja virtualizacijos technologiją, tai įgyvendinama labai paprastai. Mūsų atveju tinklalapių talpinimo paslauga, gali būti praplečiama iki kelių serverių ar

daugiau, taip paskirstant apkrovimą keliems virtualiems serveriams. Tačiau telkinio atveju, reikalingas vienas ar keli duomenų serveriai, kurie paskirytų bendrus duomenis visiems aptarnaujantiems serveriams. Visas plečiamumas vykėtų naudojantis valdymo modulių, kuris kartu atliekta naudojamų resursų apskaitą (atžymimas laikas, kada koks virtualus serveris buvo įjungtas arba išjungtas).

Jeigu prisiminti, kaip veikia blokinė talpa debesyje, tada lengva suprasti, kad visiems aptarnaujantiems virtualiems serveriams (priklauso vienam telkiniui), neišeitų prikabinti vieno duomenų diskų (db duomenys). Todėl telkinio atveju reikia, kurti vieną atskirą duomenų serverį, o kiti serveriai per tinklą pasiimtų tuos duomenis iš šio serverio (Pav. 21). Tam gali būti panaudotas paprastas failų serveris su jame įdiegta failų serverio programine įranga (angl. NFS – network file system).

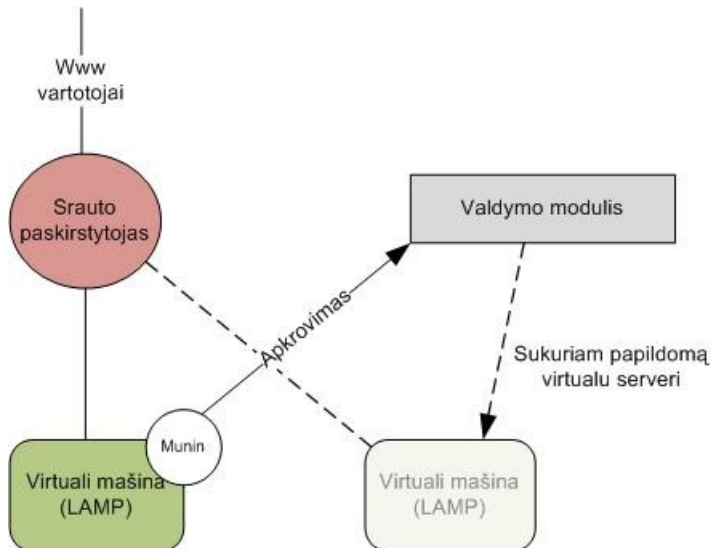


Pav. 21 Paslaugos plečiamumas skaičiavimo resursų atžvilgiu

Aptarsime kelis būdus, kaip tai įgyvendinti tinklalapių talpinimo paslaugoje.

18.1 Pagal apkrovimą

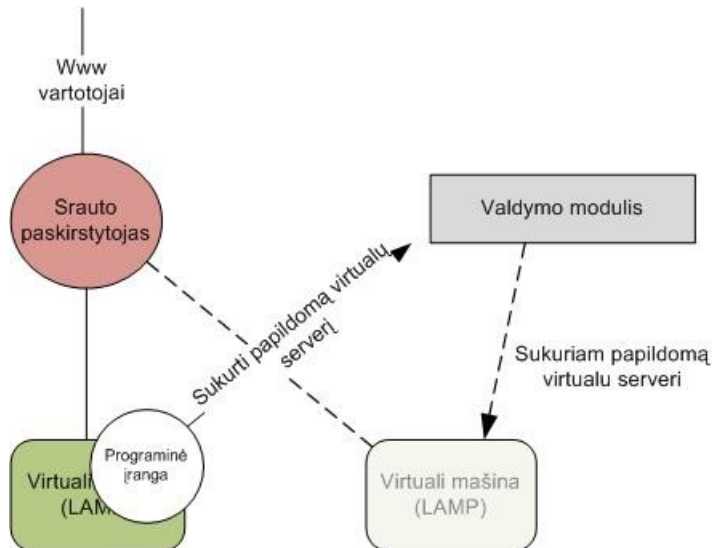
Žinant virtualių serverių apkrovimą, PaaS paslaugos sistema gali į tai reguluoti ir pagal apkrovimą sukurti reikiamą virtualių serverių skaičių. Apkrovimo duomenis galima perskaityti iš pavyzdyje pateikto apkrovos stebėjimo įrankio Munin. Taigi turime turėti tam tikrą užklausų paskirstytoją (angl. load balancer). Kuris žinodomas, kad egzistuoja virtualių serverių telkinys, paskirstytų užklausas visiems serveriams (Pav. 22).



Pav. 22 Paslaugos plečiamumas pagal apkrovimą

18.2 Pagal programinę įrangą

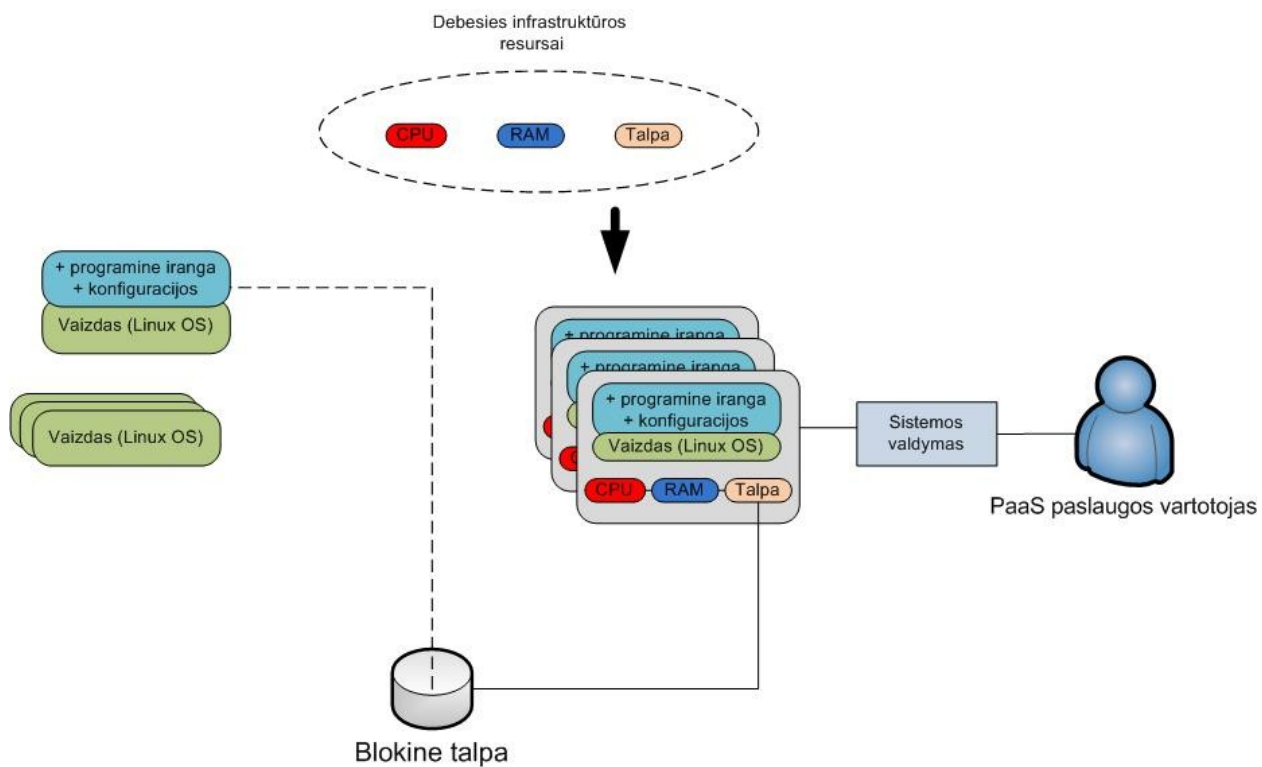
Resursus galima plėsti taip pat programiškai. Tiksliau pasakius, tai pati aplikacija nuspręstų, kada jai reikia papildomų skaičiavimo resursų. Tarkim atsiradus sudėtingo uždavinio sprendimui. Tam būtų sukuriamas papildomas virtualus serveris. Pati programinė įrangą gali ir panaikinti virtualų serverį, kuris tuo metu nereikalingas (Pav. 23).



Pav. 23 Paslaugos plečiamumas pagal uždavinį

19. Bendras PaaS paslaugos sukūrimo modelis

Pagal anksčiau pateiktą pavyzdį, galime parodyti, kaip atrodytų PaaS paslaugos modelis. Šiame modelyje pasinaudojant IaaS paslauga, sukuriama PaaS paslauga. Tokios PaaS paslaugos vartotojas gauna jam reikalingą aplinką, kurioje gali dirbti su savo programiniu kodu. Jam nereikia rūpintis aplinkos kūrimu, talpos valdymu – PaaS paslauga/ PaaS valdymo sistema už jį tą darbą atlieka. Kartu gaunama resursų panaudojimo statistika, atsarginių kopijų funkcija, yra resursų praplėtimo galimybė. Pagal tokį modelį sukuriamas sistemos prototipas (Pav.24).



Pav. 24 Taip atrodo PaaS paslaugos sukūrimas debesyje.

Taigi, pagrindinė PaaS paslaugos (internetinių puslapių talpinimo ar panašios paslaugos) funkcija yra suvaldyti skaičiavimo resursus pasinaudojant virtualizacijos technologija, bei pateikti tuos resursus, paruoštus darbui, galutiniam vartotojui.

Rezultai ir išvados

Šiame darbe detaliai išanalizavus Eucalyptus platformą ir realiai ją išbandžius, pabandyta sukurti internetinių puslapių talpinimo paslaugą debesyje (PaaS paslauga), kuria naudotis galėtų daug vartotojų. Išnagrinėjus, kokios programinės įrangos reikia tokiai paslaugai teikti, paruošti pavyzdiniai instaliaciniai skriptai, nubraižytos schemos kaip tokia paslauga galėtų veikti ir kokias funkcijas, bei kokią naudą gauna galutinis vartotojas naudodamas tokią paslaugą. Sukurtas tokios sistemos prototipas, kuris automatiškai turi rūpintis tokios paslaugos valdymu, bei stebėjimu. Pateikti tokios sistemos prototipo kodo pavyzdžiai.

Reziumuojant galima teigti, kad tokia PaaS paslauga kaip internetinių puslapių talpinimas ar panašios paslaugos, tikrai gali veikti debesies platformoje, bei išnaudoti debesies teikiamus privalumus (moki tik už tai kiek naudoji, resursų plečiamumas, bendra infrastruktūra, aukštas patikimumas). Svarbu pažymėti, kad tai įgyvendinama naudojantis nemokamais įrankiais ir platformomis. Ir tuo tarpu visas paslaugos procesas gali būti automatizuojamas, kad reikėtų kuo mažiau žmogiškųjų resursų paslaugos valdymui.

Šaltiniai

- [AmzEC2G] Amzon EC2 developer guide, <http://docs.amazonwebservices.com/AWSEC2/2009-11-30/DeveloperGuide/>
- [AmzWeb09] Amazon Web Services home page. <http://aws.amazon.com/> ,2009
- [BDFXenArt03] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the Art of Virtualization. In Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP 2003), Bolton Landing, USA, Oct. 2003
- [CIFu10] CloudFusion home page. <http://getcloudfusion.com/> , 2010
- [DRCGLDEuca09] Daniel Nurmi, Rich Wolski, Chris Grzegorzczuk, Graziano Obertelli, Sunil Soman, Lamia Youseff, Dmitrii Zagorodnov; The Eucalyptus Open-source Cloud-computing System, 2009
- [EucaAdG] Eucalyptus Administrator's Guide (1.6), <http://forum.Eucalyptus.com/wiki/Eucalyptus-administrators-guide-16>
- [EucaS09] The Eucalyptus Story, <http://forum.Eucalyptus.com/about/story> ,2009
- [EucaUsG] Eucalyptus User's Guide (1.6) <http://open.Eucalyptus.com/wiki/EucalyptusUserGuide> ,2009
- [FSARTGrid09] FINAL STATE OF THE ART D2.1.3 v2 WP2.1 - Technological Background, Standards and Reference Models, Advanced Grid Technologies, Systems and Services; 2009
- [ITCVMSH06] IT in the Cloud: Using VMware vCloud for Reliable, Flexible, Shared IT Resources, 2006
- [MARBVIC07] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia Above the Clouds: A Berkeley View of Cloud, 2007
- [MGNistv15] Peter Mell and Tim Grance. Draft NIST Working Definition of Cloud Computing v15
- [MuninT] Munin tool documentation: <http://munin-monitoring.org/wiki/Documentation>

- [OGF10] OGF Open Cloud Interface Working Group, <http://occi-wg.org/doku.php>, 2010
- [OpGr09] Slides from [Open Source Grid & Cluster](#) conference
<http://forum.Eucalyptus.com/wiki/presentations> ,2009
- [PHPDev10] PHP Developer Center, <http://aws.amazon.com/php/>, 2010
- [RCAmz09] Reuven Cohen, Amazon to Open Source Web Services API's, May. 29, 2009
- [RHPCA99] R. Buyya (ed.), High Performance Cluster Computing: Architectures and Systems, vol. 1, Prentice Hall, Upper Saddle River, USA, 1999
- [RRTh05] Rami Rosen on Thu, www.thelinuxdoctor.org/Articles/Xen_Explained.html, 2005
- [TRGooCl08] T. V. Raman Google Inc. Cloud Computing And Equal Access For All,2008

Priedai

1) LAMP funkcijų paruošimas virtualios mašinos vaizde.

Pirmiausia įdiegiamas yum įrankis, skirtas programinės įrangos paketų instaliavimo valdymui. Visa programinė įranga instaliuojama, specialioje failų sistemoje prikabintoje prie kompiuterio, kuriame ruošiamas reiklingas mums virtualios mašinos vaizdas.

```
echo "Installing Yum 3.0"
wget http://linux.duke.edu/projects/yum/download/3.0/yum-3.0.5.tar.gz
tar -xvzf yum-3.0.5.tar.gz
cd yum-3.0.5
make DESTDIR=/ install

## Sukuriam yum konfigūracija"
mkdir -p /mnt/ec2-fs/sys/block
mkdir /mnt/ec2-fs/var/
mkdir /mnt/ec2-fs/var/log/
touch /mnt/ec2-fs/var/log/yum.log

## Toliau įtraukiama, reikalinga yum konfigūracija, kuri pasako, iš kur
parsisiųsti programines įrangos paketus (galima naudoti lokalu yum repository
serverį, kas sumažintų tinklo apkrovimus)

cat <<EOL > /mnt/image/yum.conf
[main]
cachedir=/var/cache/yum
debuglevel=2
logfile=/var/log/yum.log
exclude=-debuginfo
gpgcheck=0
obsoletes=1
pkgpolicy=newest
distroverpkg=redhat-release
tolerant=1
exactarch=1
reposdir=/dev/null
metadata_expire=1800

[base]
name=CentOS - $basearch - Base
baseurl=http://mirror.ubiquityservers.com/centos/5.4/os/i386/
enabled=1
.....
.....
EOL
## Instaliuojame mums reikalingus servisus, bibliotekas
```

```

        yum -c /mnt/image/yum.conf --installroot=/mnt/ec2-fs -y
install wget httpd mysql php nano logrotate ruby* rubygems syslog-ng postfix
openssl openssh openssh-askpass openssh-clients openssh-server curl gcc* zip
unzip bison flex compat-libstdc++-296 cvs subversion autoconf automake libtool
        yum -c /mnt/image/yum.conf --installroot=/mnt/ec2-fs -y
clean packages

## sukuriama reikalinga tinklo konfigūracija, geriausia naudoti automatinį ip
adresų nustatymą, kas leis juos kontroliuoti naudojant pagrindinį dhcpd serverį.

cat <<EOL > /mnt/ec2-fs/etc/sysconfig/network
NETWORKING=yes
HOSTNAME=localhost.localdomain
EOL

cat <<EOL > /mnt/ec2-fs/etc/sysconfig/network-scripts/ifcfg-eth0
ONBOOT=yes
DEVICE=eth0
BOOTPROTO=dhcp
EOL

## automatiskai prikabiname blokinę talpą
cat <<EOL > /mnt/storage /dev/sdb      none bind" | tee -a /etc/fst

```

2) Pavyzdinis skriptas (Apache serviso konfigūracija, bei tinklapio katalogas).

```

service httpd stop
if [ -d /mnt/www ]; then
echo "Apache Base Files Exist"
else
## tinklapio katalogo perkėlimas
mv /var/www /mnt
ln -nsf /mnt/www /var/www
fi
## Apache ivykiu irasų perkėlimas
rm -fr /var/log/httpd
mkdir -p /mnt/log/httpd
ln -s /mnt/log/httpd /var/log/httpd
## Administratoriaus pastos adreso pakeitimas
perl -p -i -e "s/root@localhost/$ADMIN_EMAIL/" /etc/httpd/conf/httpd.conf
##jei reikia pataisome konfigūraciją
perl -p -i -e 's/ServerSignature On/ServerSignature Off/'
/etc/httpd/conf/httpd.conf
perl -p -i -e 's/#ExtendedStatus On/ExtendedStatus On/'

```

```
/etc/httpd/conf/httpd.conf
```

```
## mysql duomenų bazės perkėlimas į blokinę talpą  
/sbin/service mysqld stop  
mv /var/lib/mysql /ebs1/mysql  
ln -s /mnt/storage/mysql /var/lib  
/sbin/service mysqld start
```

3) Munin grafikai: tinklo apkrovimas, atminties naudojimas, CPU apkrovimas.

