

ՀԱՅԱՍՏԱՆԻ ԱԶԳԱՅԻՆ ՊՈԼԻՏԵԽՆԻԿԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ

ԼԵՎՈՆ ԳԱՍՊԱՐԻ ԿԻՐԱԿՈՍՅԱՆ

**ՏԵՂԵԿԱՏՎԱՏԵԽՆՈԼՈԳԻԱԿԱՆ ԵՆԹԱԿԱՌՈՒՑՎԱԾՔՆԵՐԻ ԵՎ
ԾԱՌԱՅՈՒԹՅՈՒՆՆԵՐԻ ՏՐԱՄԱԴՐՄԱՆ ԱՎՏՈՄԱՏԱՑՎԱԾ
ՀԱՄԱԿԱՐԳԻ ՄՇԱԿՈՒՄԸ**

Ե.13.02 – «Ավտոմատացման համակարգեր» մասնագիտությամբ տեխնիկական
գիտությունների թեկնածուի գիտական աստիճանի հայցման

ԱՏԵՆԱԽՈՍՈՒԹՅՈՒՆ

Գիտական ղեկավար,
տ.գ.դ., պրոֆ. Ա.Գ. Ավետիսյան

Երևան 2017

ԲՈՎԱՆԴԱԿՈՒԹՅՈՒՆ

ՆԵՐԱԾՈՒԹՅՈՒՆ	4
ԳԼՈՒԽ 1. ԳՐԱԿԱՆՈՒԹՅԱՆ ԱԿՆԱՐԿ ԵՎ ԱՏԵՆԱԽՈՍՈՒԹՅԱՆ ՆՊԱՏԱԿՆԵՐԸ	10
1.1. Ամպային տեխնոլոգիաների զարգացման միտումները	10
1.1.1. Ամպային տեխնոլոգիան ըստ գտնվելու վայրի	12
1.1.2. Վիրտուալացման դերը մասնավոր ամպային տեխնոլոգիայում	16
1.1.3. Ամպային տեխնոլոգիան ըստ տրամադրված ծառայության	21
1.1.4. Տվյալների կառավարումն ամպային տեխնոլոգիաներում	32
1.3. Եզրակացություններ 1-ին գլխի վերաբերյալ և ատենախոսության նպատակները	36
ԳԼՈՒԽ 2. ՏԵՂԵԿԱՏՎԱԿԱՆ ԾԱՌԱՅՈՒԹՅՈՒՆՆԵՐԻ ՏՐԱՄԱԴՐՄԱՆ ԱՎՏՈՄԱՏԱՑՎԱԾ ՀԱՄԱԿԱՐԳԻ ՄԱԹԵՄԱՏԻԿԱԿԱՆ ՈՒ ՏԵՂԵԿԱՏՎԱԿԱՆ ԱՊԱՀՈՎՈՒՄՆԵՐԸ	38
2.1. Համակարգի մաթեմատիկական ապահովումը.....	38
2.2. Համակարգի տեղեկատվական ապահովումը.....	55
2.2.1. Ceph և GlusterFS համակարգերը	55
2.2.2. CloudStack համակարգը	58
2.3. Եզրակացություններ 2-րդ գլխի վերաբերյալ.....	65
ԳԼՈՒԽ 3. ՏԵՂԵԿԱՏՎԱԿԱՆ ԾԱՌԱՅՈՒԹՅՈՒՆՆԵՐԻ ՏՐԱՄԱԴՐՄԱՆ ԱՎՏՈՄԱՏԱՑՎԱԾ ՀԱՄԱԿԱՐԳԻ ՏԵԽՆԻԿԱԿԱՆ ԱՊԱՀՈՎՈՒՄԸ	67

3.1. Ավտոմատացված համակարգի ամպային ենթակառուցվածքն ու դրա տեղադրումը.....	67
3.1.2. pfSense համակարգի տեղադրումը vmWare-ի վրա	74
3.1.4. Cloudstack համակարգի շահագործումը	80
3.1.5 Գաղտնի բանալու և հավաստագրի ստեղծումը վեբ սերվերի համար.....	82
3.2. Եզրակացություններ 3-րդ գլխի վերաբերյալ	85
ԳԼՈՒԽ 4. ՏԵՂԵԿԱՏՎԱԿԱՆ ԾԱՌԱՅՈՒԹՅՈՒՆՆԵՐԻ ՏՐԱՄԱԴՐՄԱՆ ԱՎՏՈՄԱՏԱՑՎԱԾ ՀԱՄԱԿԱՐԳԻ ԾՐԱԳՐԱՅԻՆ ԱՊԱՀՈՎՈՒՄԸ	
86	86
4.1 «ՀԱՊՀ ամպային լուծումներ» ավտոմատացված համակարգի աշխատանքային միջավայրը.....	86
4.2 Եզրակացություններ 4-րդ գլխի վերաբերյալ	93
ԵԶՐԱՀԱՆԳՈՒՄ	95
ԳՐԱԿԱՆՈՒԹՅԱՆ ՑԱՆԿ.....	97

ՆԵՐԱԾՈՒԹՅՈՒՆ

Կառավարման և ավտոմատացման բնագավառներում մեծ դեր ունի արդիական տեղեկատվական տեխնոլոգիաների՝ ամպային տեխնոլոգիաների ներդրումը: Մի շարք աշխատանքների [8, 9, 14, 17, 52], ինչպես նաև գործող համակարգերի վերլուծության հիման վրա կարելի է պնդել, որ տեղեկատվական ռեսուրսներ օգտագործող ընկերությունների համար, որտեղ առկա են ռիսկային տեղեկատվական հոսքեր, նպատակահարմար է շահագործել հիբրիդային տիպի ամպային համակարգեր՝ ընկերության աշխատունակությունը բարձրացնելու և ռիսկերից խուսափելու նպատակով: Նման համակարգերի աշխատանքի արդյունավետության բարձրացման նպատակով անհրաժեշտություն է առաջանում լուծել բազմաթիվ խնդիրներ, մասնավորապես ֆինանսական միջոցների նվազեցում [6, 29, 44, 58], համակարգի ծանրաբեռնվածության նվազեցում [18, 32, 34, 50, 68], տեղեկատվական ռեսուրսների օպտիմալ բաշխում [69, 47, 4, 28, 24, 30] և այլն: Յուրաքանչյուր կազմակերպության համար տեղեկատվական ռեսուրսների՝ ենթակառուցվածքների ու ծառայությունների տրամադրման ավտոմատացված համակարգերի մշակման ժամանակ պետք է ընտրվեն համապատասխան տեխնոլոգիաներ, մշակվեն դրանց համատեղ աշխատանքի սկզբունքները, ինչպես նաև համակարգի աշխատանքի բարձր արդյունավետությունն ապահովող միջոցներ:

Թեմայի արդիականությունը պայմանավորված է տեղեկատվատեխնոլոգիական ենթակառուցվածքների ու ծառայությունների տրամադրման ավտոմատացված համակարգի մշակման անհրաժեշտությամբ՝ հիբրիդային ամպի ստեղծման ու շահագործման ճանապարհով:

Ատենախոսության նպատակը տեղեկատվատեխնոլոգիական ենթակառուցվածքների ու ծառայությունների տրամադրման ավտոմատացված համակարգի մշակումն ու ներդրումն է:

Նպատակին հասնելու համար առաջադրվել են հետևյալ խնդիրները՝

1. Ուսումնասիրել ամպային տեխնոլոգիաների տարատեսակներն ու ընտրել ստեղծվող համակարգի շահագործման տեսանկյունից առավել նպատակահարմարը:
2. Մշակել հիբրիդային ամպերում ռեսուրսների բաշխման ավտոմատացված համակարգ՝ հիմնվելով օգտատերերի նախապես տրամադրած պարամետրերի և համակարգի ընթացիկ աշխատանքի ընթացքում հավաքագրված տեղեկատվության վերլուծության վրա:
3. Մշակված մեթոդների հիման վրա ստեղծել շահագործողական վեբ ինտերֆեյս, որը հնարավորություն կընձեռի օգտատերերին ստանալ համակարգի կողմից տրամադրվող ծառայությունները:
4. Համակարգում տեղակայվող տվյալների անվտանգ ու հուսալի պահուստավորման համար ընտրել, տեղադրել ու շահագործել օբյեկտային տվյալների պահպանման համակարգ:
5. Կիրառելով Հայաստանի ազգային պոլիտեխնիկական համալսարանի տեղեկատվական ռեսուրսները՝ ներկառուցել հիբրիդային ամպային համակարգ:

Գիտական նորույթը:

- մշակվել է տեղեկատվատեխնոլոգիական ենթակառուցվածքների ու ռեսուրսների տրամադրման ավտոմատացված համակարգ հիբրիդային ամպի կառուցման ու շահագործման ճանապարհով:

- մշակվել է հիբրիդային ամպի ռեսուրսների օպտիմալ շահագործման նպատակով ամպային տարածքի ընտրման մեթոդ՝ ինտեգրալ գնահատականների ձևավորման նպատակով կշռային գործակիցների օպտիմալ արժեքների որոշման ճանապարհով:
- հիմնավորվել ու ընտրվել են ավտոմատացված համակարգի ենթահամակարգերի գործունեության տեսանկյունից նպատակահարմար տեխնոլոգիաներն ու ծրագրային համակարգերը տեխնիկական, տեղեկատվական ու ծրագրային ապահովման ենթահամակարգերի ստեղծման ու համատեղ արդյունավետ աշխատանքի համար:

Գործնական արժեքը: Տեղեկատվատեխնոլոգիական ենթակառուցվածքների ու ծառայությունների տրամադրման «ՀԱՊՀ ամպային լուծումներ» ավտոմատացված համակարգը տեղադրվել է Հայաստանի ազգային պոլիտեխնիկական համալսարանում (ներդրման ակտը ներկայացված է աշխատանքում) և հնարավորություն է տալիս հիբրիդային ամպի տրամադրած ռեսուրսներով ընդլայնել մասնավոր ամպի հնարավորությունները՝ չսահմանափակվելով համալսարանի ֆիզիկական ռեսուրսներով: Մշակված վեբ ինտերֆեյսը թույլ է տալիս օգտատերերին գրանցվել համակարգում և ստեղծել նախագծեր ցանկալի պարամետրերով:

Աշխատանքում մշակված ավտոմատացված համակարգի մոդելը կարելի է կիրառել այլ կազմակերպություններում՝ առկա ֆիզիկական ռեսուրսների պարամետրերի հաշվառմամբ:

Պաշտպանության դուրս բերվող հիմնական դրույթները.

- տեղեկատվատեխնոլոգիական ենթակառուցվածքների ու ռեսուրսների տրամադրման ավտոմատացված համակարգը՝ հիբրիդային ամպի կիրառմամբ,
- հիբրիդային ամպի ռեսուրսների օպտիմալ շահագործման նպատակով ամպային տարածքի ընտրման մեթոդը՝ ինտեգրալ գնահատականների

ձևավորման նպատակով կշռային գործակիցների օպտիմալ արժեքների որոշմամբ,

- ավտոմատացված համակարգի տեխնիկական ապահովման մոդելը,
- ավտոմատացված համակարգի տեղեկատվական ապահովման մոդելը,
- ավտոմատացված համակարգի ծրագրային ապահովումը՝ օգտատերերի համար նախատեսված հարմարավետ ու պարզ վեբ ինտերֆեյսով:

Աշխատանքի տեսական և գործնական արդյունքները զեկուցվել են՝

- Հայաստանի ազգային պոլիտեխնիկական համալսարանի (նախկին ՀՊՃՀ) տարեկան գիտաժողովներում (2016թ.);
- «Տեղեկատվական տեխնոլոգիաներ և ավտոմատացում», «Քոմպյուտերային համակարգեր և ցանցեր» ամբիոնների գիտական սեմինարներում (2014-2017թթ.):

Հրատարակումները: Ատենախոսության հիմնական արդյունքները հրատարակված են 4 գիտական աշխատանքներում, որոնք ներկայացված և ընդգծված են գրականության ցանկում:

Ատենախոսության կառուցվածքը և ծավալը: Ատենախոսությունը բաղկացած է ներածությունից, 4 գլուխներից, եզրահանգումից և գրականության ցանկից, որը ներառում է 109 անվանում: Հիմնական նյութը շարադրված է 106 տպագիր էջի վրա, որոնք ներառում են 68 նկարներ և 6 աղյուսակներ:

Ներածությունում հիմնավորված է ատենախոսության թեմայի արդիականությունը, ներկայացված են ատենախոսության նպատակները, պաշտպանության դուրս բերվող հիմնական դրույթները, աշխատանքի հակիրճ նկարագրությունը և հիմնական գիտական արդյունքները:

Առաջին գլխում ներկայացված են ուսումնասիրություններ ամպային տեխնոլոգիաների արդիականության մասին, վերջինիս տարբեր տիպերի և տրամադրվող ծառայությունների տարբերությունների մասին: Ուսումնասիրվել են նաև օբյեկտային

եղանակով տվյալների պահպանման հիմնական մեթոդները և իրականացման եղանակները: Ներկայացված են աշխատանքներ, տարբեր կազմակերպությունների, նաև բուհական, տեղեկատվական ենթակառուցվածքներում հիբրիդային ամպի կիրառումների մասին: Վերլուծված են աշխատանքներ, նվիրված հիբրիդային ամպի շահագործման ժամանակ առաջացող խնդիրներն՝ ծախսերի նվազեցում, տեղեկատվական հոսքերի օպտիմալ բաշխում և այլն:

Երկրորդ գլուխը նվիրված է ավտոմատացված համակարգի մաթեմատիկական ու տեղեկատվական ապահովումներին:

Ձևակերպված է համակարգի ընթացիկ վիճակով պայմանավորված կշռային գործակիցների օպտիմալ արժեքների որոշման խնդրի մաթեմատիկական մոդելը: Գլխում ներկայացված են համակարգում վճիռների կայացման ալգորիթմը, մասնավորապես հիբրիդային ամպում ռեսուրսների բաշխման մոդելը ու դրա կիրառմամբ մի քանի սցենարներով ստացված արդյունքները:

Ներկայացված են օբյեկտային եղանակով տվյալների պահպանման համակարգերն ու դրանցից առաջադրված խնդրի տեսանկյունից առավել նպատակահարմար տարբերակի ընտրության հիմնավորումները: Ներկայացված են նաև ուսումնասիրված վիրտուալացման մեթոդներն ու դրանց վերլուծության հիման վրա ընտրված վիրտուալացման համակարգը, որի օգնությամբ աշխատելու են ավտոմատացված համակարգի մնացած բոլոր ենթահամակարգերը՝ հիբրիդային ամպի շահագործման համար:

Երրորդ գլուխը նվիրված է համակարգի տեխնիկական ու ծրագրային ապահովումներին: Գլխում ներկայացված է համակարգի ցանցային ճարտարապետությունը, ինչպես նաև ավտոմատացված հիբրիդային ամպի տեղադրման համար անհրաժեշտ ցանցային կարգավորումներն ու համակարգերի տեղադրման քայլերը: Ներկայացված են նաև համակարգի տեղեկատվական անվտանգության ապահովման համար առաջարկվող միջոցներն ու դրանց տեղադրման քայլերը:

Չորրորդ գլխում ներկայացված է մշակված «ՀԱՊՀ ամպային լուծումներ» ավտոմատացված համակարգի շահագործողական վեբ ինտերֆեյսը, օգտատերերի կողմից շահագործման ենթակա էջերը: Ներկայացված են համակարգից օգտվելու քայլերն ու օգտատերերին ընձեռնված հնարավորությունները:

Եզրահանգման մեջ ընդհանրացված են աշխատանքում ստացված գիտագործնական արդյունքները:

ԳԼՈՒԽ 1. ԳՐԱԿԱՆՈՒԹՅԱՆ ԱԿՆԱՐԿ ԵՎ ԱՏԵՆԱԽՈՍՈՒԹՅԱՆ ՆՊԱՏԱԿՆԵՐԸ

1.1. Ամպային տեխնոլոգիաների զարգացման միտումները

Ամպային տեխնոլոգիայի յուրաքանչյուր պահին և յուրաքանչյուր վայրից հասանելի լինելու առանձնահատկությունն ապահովում է դրա արագ զարգացումը: Ամպային տեխնոլոգիաների կիրառման արդյունքում նույն ֆիզիկական տարածքի շահագործումը հազարավոր օգտատերերի կողմից, ապահովում է նաև դրանց կիրառման ֆինանսական շահավետությունը: Սակայն, միաժամանակ մեծ քանակով օգտատերերի սպասարկումն առաջացնում է նոր խնդիրներ, որոնք պահանջում են լուծման նոր մոտեցումներ ինֆորմացիայի գրանցման/ընթերցման համար: Ամպային տեխնոլոգիաների պարագայում առաջնային է դառնում ոչ թե ֆիզիկական սարքերի հնարավորությունների մեծ լինելը, այլ այն, թե ինչպես են այդ միջոցները շահագործվում ծրագրային մակարդակում: Այդ իսկ պատճառով մատուցվող ծառայությունների որակի ապահովումն է, որ մեծ դեր ունի այս տիպի համակարգերի աշխատանքում: Ամպային համակարգը նախկինում ստեղծված նմանատիպ ծառայություններից տարբերակելու համար անվանում են համակարգ, որը մատուցում է համակարգչային ռեսուրսներ ու ծառայություններ ցանցի միջոցով: Ամպային համակարգի արագագործությունը, հուսալիությունն ու ֆիզիկական ռեսուրսների շահավետ կիրառումն ապահովելու տեսանկյունից մեծ դեր ունի վիրտուալացումը: Օգտատերերի քանակի, ինֆորմացիայի ծավալի աճի հետ կապված հասկանալի դարձավ, որ տվյալների պահպանման ընդունված տարբերակը, երբ ինֆորմացիան պահպանվում է սերվերների կոշտ սկավառակների վրա ընդունելի չի լինի ինչպես անվտանգության, այնպես էլ ֆինանսական տեսանկյուններից: Այդ իսկ պատճառով

ամպային համակարգում պահպանվող տվյալները գրանցվում են մեծ քանակով ավելի էժան կոշտ սկավառակների վրա, որոնք կենտրոնացված համակարգի կողմից կառավարվելով՝ ստեղծում են միջավայր, որտեղ մեկ կրիչի վնասման դեպքում խնդիր չի կարող առաջանալ, քանի որ այդ կրիչի վրա պահպանվող տվյալներն արդեն իսկ գրված են լինում մեկ այլ կրիչի վրա, իսկ շարքից դուրս եկածը հնարավորինս արագ փոխարինվում է նորով: Այսպիսով՝ ամպային տեխնոլոգիայի տվյալների կենտրոնացված համակարգի կառուցվածքի շահագործումն ունի մի շարք առավելություններ, ինչպիսիք են՝ համակարգչային ռեսուրսների ֆինանսական շահավետությունը, դրանց կառավարման հարմարավետությունը:

Ամպային ծառայություն մատուցող կազմակերպության համար մեկ այլ կարևոր առավելություն է այն, որ այն տարածքը, որը չի օգտագործվում մեկ օգտատիրոջ կողմից, կարող է տրամադրվել մեկ այլ օգտատիրոջ: Այսինքն ստացվում է, որ համակարգում առկա ռեսուրսները կարող են օգտագործվել օպտիմալ՝ բաշխման օպտիմալ ալգորիթմների կիրառմամբ [64]:

Յուրաքանչյուր ընկերության համար շատ կարևոր է ունենալ բարձր թողունակությամբ և ճկունությամբ ամպային համակարգ: Այս երկու չափանիշները միակը չեն, որ պետք է հաշվի առնվեն նման համակարգեր շահագործելիս: Այլ շատ կարևոր հանգամանքներ են՝ անվտանգությունը, անխափան աշխատանքը [60]:

Ըստ Ստանդարտների և տեխնոլոգիաների ազգային ինստիտուտի սահմանման «Ամպային համակարգը մոդել է համատարած, հարմար, ըստ պահանջի համատեղ օգտագործման տիրույթում կարգավորվող համակարգչային ռեսուրսներին (ցանցերի, սերվերների, հիշողության, ծրագրերի և սերվիսների) ցանցային հասանելիություն ապահովելու համար, որը կարող է տրամադրվել արագ և տրամադրողի կողմից մինիմալ կարգավորումներով» [19]:

Ամպային տեխնոլոգիաների անընդհատ զարգացումն ապահովեց դրա մուտքը տարբեր բնագավառների կազմակերպություններ: Դրանք սկսեցին կիրառվել ինչպես

տարբեր ոլորտների մեծ ընկերություններում, այնպես էլ կրթական ու գիտական հաստատություններում:

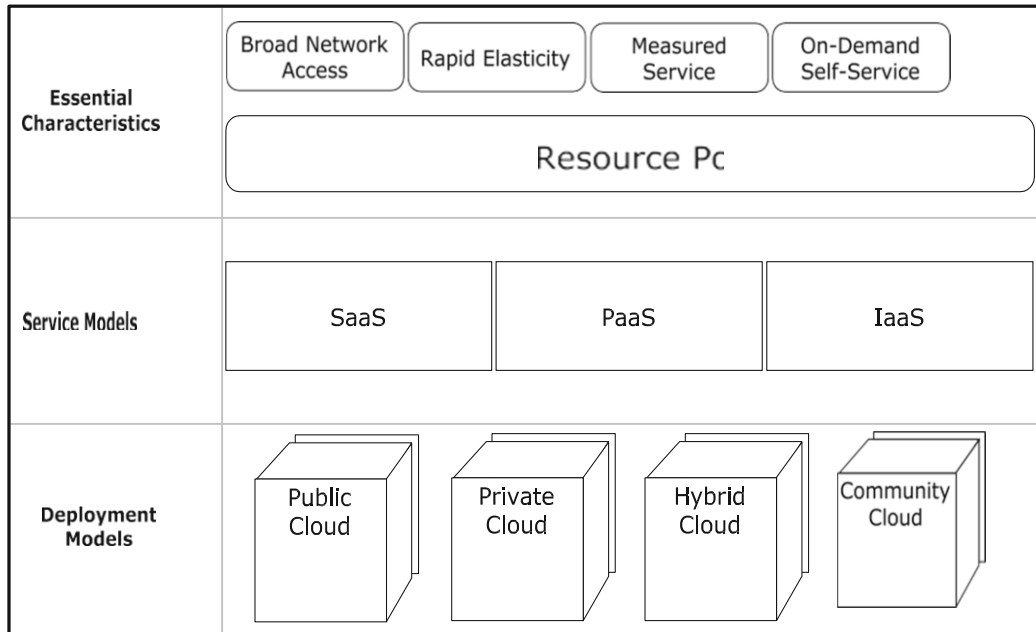
Գոյություն ունեն ամպային տեխնոլոգիաների մի քանի տիպեր: Նախ անհրաժեշտ է դրանք բաժանել երկու հիմնական խմբերի՝

- ըստ գտնվելու վայրի (ամպային մոդել)
- ըստ ծառայության (ամպային ծառայության մոդել)

1.1.1 Ամպային տեխնոլոգիան ըստ գտնվելու վայրի

Ըստ գտնվելու վայրի գոյություն ունեն ամպային տեխնոլոգիայի չորս հիմնական տիպեր [51] (նկ.1.1)՝

- Հանրային ամպ (Public cloud),
- Մասնավոր ամպ (Private cloud),
- Հիբրիդային ամպ (Hybrid cloud),
- Համայնքային ամպ (Community cloud):



Նկ. 1.1 Ամպային տեխնոլոգիայի տիպեր [51]

Հանրային ամպ (Public cloud): Հանրային տիպի ամպային տեխնոլոգիան տրամադրվում է որևէ ընկերության կողմից իր հաճախորդներին [51]: Այս դեպքում հաճախորդը ֆիզիկական մուտք չունի դեպի իրեն հատկացված ռեսուրսները: Այս տարբերակն ամենապարզն է և մեծ կիրառում ունի ամբողջ աշխարհում: Այս տիպի ամպային տեխնոլոգիա տրամադրում են մի շարք ընկերություններ, որոնցից են օրինակ Dropbox-ը [70], Microsoft-ը [71], Amazon-ը [72], Google-ը [73] և այլն: Ունենալով մի շարք առավելություններ՝ անվճար լինելը, ապահովությունը և այլն, այս տիպի ամպային տեխնոլոգիան ունի նաև թերություններ՝ անվճար տարբերակում հիշողության ծավալի սահմանափակումները որոշ ընկերությունների կողմից և տեղադրված տվյալների անվտանգության մակարդակը: Տվյալների անվտանգության հարց կարող է առաջանալ օրինակ այն դեպքում, երբ շատ օգտատերեր համակարգում գրանցվելիս օգտագործում են նույն էլ-փոստը և/կամ գաղտնաբառը, որը կիրառել են նաև այլ համակարգերում գրանցվելիս: Այս պարագայում, եթե նույնիսկ ամպային տեխնոլոգիան տրամադրող ընկերությունն իրականացնում է բոլոր ջանքերը իր համակարգի անվտանգությունն ապահովելու համար, հաքերները այլ պլատֆորմներից կորզելով օգտատերերի տվյալները կարող են փորձեր կատարել օգտատերերի ամպում տեղադրված ֆայլերը դիտելու համար: Այս խնդիրները հաշվի առնելով, ծառայությունը մատուցող ընկերություններն արդեն իսկ առաջարկում են երկքայլանի վերիֆիկացիայի մեթոդ, երբ նոր համակարգչից մուտք գործելու դեպքում համակարգը կարճ հաղորդագրություն է ուղարկում օգտատիրոջ հեռախոսահամարին, որը նա նախապես գրանցում է իր տվյալներում:

Հանրային ամպն ունի հետևյալ առավելությունները [74].

- ընդլայնման մեծ հնարավորություն - համապատասխան գումարի վճարման դիմաց հանրային ամպը կարելի է անընդհատ մեծացնել, ի վերջո ունենալով անսահմանափակ ծավալի հիշողություն,
- ծախսերի արդյունավետություն,

- Utility style costing – հանրային ամպերը մեծամասամբ կիրառում են վճարի օգտագործելիս տարբերակը, երբ հաճախորդը վճարում է ծառայության համար այն ժամանակ, երբ այն պետք է կիրառի: Այս հատկանիշը հնարավորություն է տալիս լրացուցիչ գումար չվատնել,
- հուսալիություն – հաշվի առնելով օգտատերերի քանակը, սերվերների հսկա քանակը ապահովում է հուսալիություն, որ եթե համակարգի որևէ մասնիկ խափանվի, դրան այլ տարրերն արագ կփոխարինեն:
- ճկունություն,
- անկախություն գտնվելու վայրից՝ համակարգը հասանելի է յուրաքանչյուր վայրից ի շնորհիվ համացանցի:

Մասնավոր ամպ (Private cloud): Այսպիսի ամպը ստեղծվում է մասնավոր ընկերության կողմից: Չնայած նրան, որ ամպը կարող է հասանելի լինել արտաքին ցանցից, դրանից կարող են օգտվել միայն ընկերության աշխատակիցները/օգտատերերը [75]: Մասնավոր ամպը կարող է ստեղծվել ինչպես ընկերության ուժերով, այնպես էլ այլ ընկերության կողմից, համապատասխան թույլտվության տրամադրմամբ միայն ծառայությունը պատվիրող կողմին: Մասնավոր ամպը հնարավորություն է տալիս ընկերություններին կատարել խնայողություններ, քանի որ ամպի շահագործման պարագայում կրճատվում են սարքերի քանակը՝ միաժամանակ ապահովելով ավելի մեծ ճկունություն և անվտանգություն: Ունենալով շատ առավելություններ՝ մասնավոր ամպն ունի նաև խնդիրներ, որոնք սակայն լուծելի են: Մասնավոր ամպի շահագործման համար անհրաժեշտ է նախագծել դրա անընդհատ վերահսկվող ճարտարապետությունը: Ճարտարապետության նախնական կազմակերպումը ամպային համակարգի համար անհրաժեշտ է այնքանով, որ եթե առաջիկայում անհրաժեշտ լինի այն ընդլայնել կամ կատարել այլ փոփոխություններ, ապա դրանք արվեն արագ և հեշտ:

Մասնավոր ամպի շահագործումից առաջ ընկերությունը պետք է որոշի տեղադրել այն իր միջոցներով, թե ծառայությունը ստանա այլ ընկերությունից: Այլ ընկերության կողմից տրամադրված մասնավոր ամպը հաճախ անվանում են կառավարվող մասնավոր ամպ: Սակայն, հաշվի առնելով, որ մասնավոր ամպի շահագործումն ավելի նախընտրելի է հանրային ամպից անվտանգության ապահովման նկատառումներով, այլ ընկերության կողմից դրա մատուցումը ևս փոքր ինչ հակասական կարող է լինել: Այս պարագայում առաջնային խնդիր է ընկերության համար գաղտնի ինֆորմացիայի տիրապետումը ծառայությունը մատուցող ընկերության կողմից: Առաջին հայացքից թվում է թե, որ այդ ընկերությունը ուղղակի դրա իրավունքը չունի, սակայն լիցենզավորման և պայմանագրային կետերի փոփոխման դեպքում, եթե պատվիրատուն այլևս չցանկանա օգտվել այդ ծառայությունից, մեկ այլ ընկերություն տվյալների տեղափոխումը կարող է երկարատև ու բարդ գործընթաց դառնալ: Հարցեր են առաջանում նաև այն դեպքերում, երբ հաքերները ձեռք են բերում ընկերության տվյալները: Չնայած ամպային ծառայություն մատուցող ընկերությունները մեծամասամբ ապահովում են բարձր անվտանգության, սակայն չի բացառվում հաքերների հարձակման արդյունքում պատվիրատու ընկերության ինֆորմացիայի կորուստը: Այս պարագայում խնդիրը նրանում է, թե արդյոք ծառայություն մատուցող ընկերությունն ամբողջությամբ կկրի հետևանքների պատասխանատվությունը: Այսպիսով, հարց է առաջանում, թե ինչու ընկերությունը պետք է իր մասնավոր ամպը մատուցի որևէ այլ ընկերության միջոցով: Պատասխանը, հավանաբար, ֆինանսապես շահավետ լինելն է: Շատ ընկերություններ միջոցներ չունեն ամպային ծառայություններ մատուցող առանձին սերվերային հանգույց ստեծելու համար: Մասնավոր ամպը տեղադրելուց առաջ կարևոր է հաշվի առնել, մեկ վիրտուալ մեքենայի միջինացված արժեքը, որը կարող է խիստ տատանվել՝ տարբեր հանգամանքներից կախված: Cloud Technology Partners-ի գլխավոր ճարտարագետ և տեխնոլոգիաների մասնագետ Էրիկ Սեբեստանի կատարած հետազոտությունների արդյունքում պարզվել է, որ եթե ընկերությունը իր ուժերով է տեղադրում մասնավոր

ամալը, ապա յուրաքանչյուր վիրտուալ մեքենայի միջինացված արժեքը կարող է հասնել մինչև 1000 ԱՄՆ դոլլարի, իսկ ամպային ծառայություններ մատուցող ընկերության դեպքում մեկ վիրտուալ մեքենայի արժեքը կարող է լինել միջինում 150-400 ԱՄՆ դոլլար: Իհարկե, շուկայական գները հասկանալու համար անհրաժեշտ է ուսումնասիրել բոլոր առաջարկները և հետո կայացնել վերջնական որոշում՝ կառուցել մասնավոր ամպն ինքնուրույն, թե օգտվել ծառայությունը մատուցող ընկերություններից: Ֆինանսապես լավագույն լուծումը միշտ էլ հանդիսանում է օգտվել բաց ծրագրային կոդով տարբերակներից՝ հաշվի առնելով վերջիններիս անվճար լինելը: Չնայած առաջին հայացքից կարող է թվալ, թե դրանք չեն կարող տրամադրել այն բոլոր ծառայությունները, որոնք անհրաժեշտ են կամ չունեն անվտանգության պահանջվող նորմերը, սակայն դրանց բաց հասանելիությունը ապահովում է խնդիրների արագ հայտնաբերմանն ու լուծմանը:

1.1.2. Վիրտուալացման դերը մասնավոր ամպային տեխնոլոգիայում

Հաճախ վիրտուալացումը համարում են մասնավոր ամպի շահագործում: Սակայն իրականում դա այդպես չէ: Վիրտուալացումը տեխնոլոգիա է, որտեղ ֆիզիկական սարքերը տարանջատվում են օպերացիոն համակարգերից և դրանց վրա տեղադրված ծրագրերից: Այս տեխնոլոգիան վերջին ժամանակների գուցե և ամենակարևոր ներդրումներից է, որը հեղափոխեց տեղեկատվական ոլորտը: Վիրտուալացման ամենակարևոր հատկանիշներից մեկն այն է, որ հնարավորություն է տալիս կատարել հսկայական խնայողություններ սարքավորումների ձեռքբերման ու շահագործման ոլորտում: Մինչ վիրտուալացման կիրառումը սերվերների շահագործման օգտակար գործողության գործակիցը շատ ավելի ցածր էր, որը պայմանավորված էր այն հանգամանքով, որ յուրաքանչյուր սերվերի վրա

տեղադրված էր լինում մեկ օպերացիոն համակարգ: Ստացվում էր, որ յուրաքանչյուր նոր համակարգի համար գնվում էր նոր սերվեր: Այստեղ կարևոր է հաշվի առնել նաև էներգիայի օգտագործման ծախսերը յուրաքանչյուր նոր սերվերի շահագործման ժամանակ:

Վիրտուալացումն ապահովելով ֆիզիկական սերվերների արդյունավետ կիրառում, հնարավորություն է տալիս շահագործել մեկից ավելի օպերացիոն համակարգեր մեկ ֆիզիկական սարքի վրա: Վիրտուալացման միջոցով սերվերի վրա ստեղծվում են վիրտուալ մեքենաներ, որոնք իրականում հանդիսանում են ծրագրային միջավայր օպերացիոն համակարգի շահագործման համար [57]: Բոլոր վիրտուալ մեքենաներն աշխատում են իրարից առանձնացված միջավայրերում:

Սակայն ինքն իրեն վիրտուալացումը մասնավոր ամպ չէ, այն մասնավոր ամպի կարևորագույն հիմնարար բաղադրիչ է: Յուրաքանչյուր ընկերություն, որը ցանկանում է ունենալ մասնավոր ամպ պետք է ունենա հստակ պատկերացում հետևյալ խնդիրների մասին՝

- տեխնիկական հնարավորությունները,
- ամպի շահագործման համար նոր սարքավորումների անհրաժեշտությունը,
- նորակառույց ամպի տրամադրող ծառայությունները

Այս երեք հիմնական խնդիրներից բացի անհրաժեշտ է հանգամանորեն վերլուծել ևս մի քանիսը, որոնցից առաջնայինը անվտանգությունն է: Ընդունված է համարել, որ մասնավոր ամպերը հանրայիններից անվտանգ են: Սակայն կան նաև այլ կարծիքներ: Հանրային ամպի անվտանգության նկատմամբ կասկածները պատճառաբանվում են այն հանգամանքով, որ հանրային ամպը չի ապահովում end-to-end encryption, սակայն նմանատիպ ծառայություններ տրամադրող ընկերություններն արդեն իսկ սկսել են կիրառել գաղտնագրման այս տարբերակը հանրային ամպի ծառայություններ մատուցելիս: Իհարկե բոլոր ընկերությունները չեն ապահովում

անվտանգություն բարձր մակարդակ, հանրային ամպ տրամադրելիս, սակայն դրա հիման վրա այն համարել ոչ անվտանգ, այդքան էլ ճիշտ չէ [76]:

Հաշվի առնելով, որ ըստ Ֆորեստերի հետազոտության նախկին 18 ամիսների ընթացքում հետազոտված ընկերությունների 50% -ից ավելն ունեցել են անվտանգության ապահովման խնդիրներ [77], հարց է առաջանում, թե ամպային լուծումներից ո՞րն է ավելի անվտանգ: Պետք է նաև հասկանալ, որ յուրաքանչյուր ամպային ծառայություն մատուցող ընկերության անվտանգությանն ուղղված բյուջեն ավելի մեծ է, քան այն ընկերությանը, որտեղ SS-ն հանդիսանում է դրա ստորաբաժանումներից մեկը և միգրացե այս իսկ պատճառով հանրային ամպը երբեմն կարող է ավելի անվտանգ լինել, քան ընկերության կողմից ստեղծված մասնավորը:

Այս բոլոր հանգամանքները հաշվի առնելով՝ մասնավոր ամպի կազմակերպումից առաջ պետք է ունենալ հստակ պահանջներ անվտանգության նկատմամբ: Սակայն ընդունելով, որ ամպային տեխնոլոգիան կարող է շահագործվել բարձր կարևորության տվյալների պահպանման համար՝ պետք է նաև վերանայել իրավական նորմերը տեխնոլոգիայի շահագործման համար: Հաշվի առնելով ամպային տեխնոլոգիաների արագ զարգացումը և պետական գործունեության մեջ դրանց խիստ ներառված լինելը, տարբեր պետություններ կատարում են աուդիտներ բնագավառում՝ հասկանալու համար, թե ինչքանով են դրանք անվտանգ: Որպես օրինակ, Միացյալ Թագավորությունում քառ ընկերության կողմից անցկացված «Ինֆորմացիայի անվտանգության խախտումների 2015թ հարցում» ծրագրի շրջանակներում 2015 թվականին հարցմանը մասնակցած մեծ ընկերությունների 90%-ը ունեցել է անվտանգության խնդիրներ, ի տարբերություն 2014 թվականի, երբ այդ նույն ցուցանիշը կազմել է 81% [77]: Հարցված ընկերություններից 59%-ը կարծում է, որ անվտանգության ապահովման խնդիրները շարունակելու են աճել, որի հետ մեկտեղ աճելու են նաև այդ խնդիրների կարգավորման համար նախատեսված գումարները: Եթե մեկ ընկերություն, որն ունի մոտ 500 աշխատող, ունեցել է ինֆորմացիայի անվտանգության խախտում, ապա այն 2015 թվականին այդ

խնդիրների վերացման համար ծախսել է մոտ £1,46 միլիոն, ի տարբերություն 2014 թվականին ծախսած £600 000-ի [77]: Անվտանգության խնդիրները շարունակում են մեծամասամբ պատճառված լինել արտաքին հարձակումներով, որոնք կազմում են 69% մեծ կազմակերպությունների համար և 38% փոքր կազմակերպությունների համար: Երկու ցուցանիշներն էլ շարունակել են աճել նախորդ տարվա համեմատ, երբ դրանք համապատասխանաբար եղել են 55% և 33% [77]: Սակայն հարցված ընկերություններում առաջացել են նվտանգության ապահովման խնդիրներ, որոնք կապված են եղել իրենց իսկ աշխատակիցների հետ (մեծ ընկերությունների 75%-ը նշել են նմանատիպ դեպքեր, իսկ փոքր ընկերությունների համար ցուցանիշը կազմել է 31%):

Նմանատիպ խնդիրներից ելնելով՝ տարբեր պետություններ ունեն իրենց կանոնակարգերը էլեկտրոնային ինֆորմացիայի շահագործման և պահպանման համար: Օրինակ, ԱՄՆ-ում պահպանվող ինֆորմացիան պետք է համապատասխանի HIPAA-ի կանոնակարգերին [78], իսկ Եվրոպական Միության անդամ երկրներում պահպանվող էլեկտրոնային ինֆորմացիան պետք է համապատասխանի Data Protection Directive կանոնակարգին [79]: Սակայն ամպային տեխնոլոգիաների շահագործման ժամանակ կազմակերպություններին նմանատիպ կանոնակարգերին համապատասխանեցնելը դառնում է ավելի դժվար, քանի որ կազմակերպությունը, որը օրինակ գտնվում է ԱՄՆ-ում և իր ընթացիկ օգտագործման ինֆորմացիան պահպանում է իր սերվերային հանգույցում, որը կրկին գտնվում է ԱՄՆ-ում և հետևաբար ենթարկվում է HIPAA-ի կանոնակարգերին, կարող է իր պահեստային տվյալների բազաները պահպանել մեկ այլ երկրում տեղակայված սերվերների վրա: Այս պարագայում պահեստային տվյալները այլևս չեն ենթարկվի ԱՄՆ օրենքներին և հետևաբար դրանց վերստուգումն ավելի կբարդանա [79]:

Այս խնդիրներից ելնելով՝ ամպային ծառայություն ընտրելիս ցանկալի է ընտրել այնպիսի ընկերություն, որի բոլոր տվյալների բազաները գտնվում են մեկ կանոնակարգով կարգավորվող տարածքում: Խնդրի մեկ այլ լուծում է օգտվել այնպիսի

ընկերությունից, որը գաղտնագրելով բոլոր տվյալները, բանալին պահպանում է ոչ թե նույն ամպում, այլ հենց իր ֆիզիկական ռեսուրսներում: Ներկայումս մշակվում է անվտանգության ևս մեկ համակարգ, որը հիմնված է լինելու տոկենային գաղտնագրման վրա [80]:

Հիբրիդային ամպ (Hybrid cloud): Վերջին ժամանակներում առավել հաճախ կիրառվում է հանրային և մասնավոր ամպերի խառնուրդը, որը կոչվում է հիբրիդային ամպ [81, 24, 25]: Հիբրիդային ամպի գաղափարը կայանում է նրանում, որ այս տարբերակի դեպքում ռիսկային տվյալները տեղադրվում են մասնավոր ամպում, իսկ ավելի քիչ կարևորություն ունեցող տվյալները՝ հանրայինում: Օրինակ, ապահովագրական ընկերությունը, որն ունի երկու տիպի տվյալների բազա, որոնցից առաջինը պոտենցիալ հաճախորդների ցուցակն է, իսկ մյուսը արդեն իսկ պայմանագիր կնքած հաճախորդներինը, տեղադրում է առաջին ցուցակի բազան հանրային ամպում, իսկ երկրորդինը՝ մասնավոր ամպում, հաշվի առնելով, որ երկրորդ ցուցակը պարունակում է իր իսկ հաճախորդների անձնական տվյալները՝ ինչպիսիք են անձնագրի, սոցիալական քարտի և վճարման քարտի մասին տեղեկությունները: Այսպիսով ստացվում է, որ հիբրիդային ամպը պարունակում է հանրային և մասնավոր ամպերի բոլոր դրական կողմերը՝ հնարավորինս զերծ մնալով դրանց թերություններից: Ելնելով այս ամենից զարմանալի չէ, որ այս տիպի ամպային տեխնոլոգիան շատ արագ զարգանում է: Ըստ Տվյալների Միջազգային Կազմակերպության կատարած հարցումների, հարցված կազմակերպությունների կեսը նշել է, որ պլանավորում է մոտ ապագայում անցում կատարել մասնավոր ամպի, իսկ մյուս կեսը՝ հիբրիդային ամպի [82]: Հետազոտությունները նաև ցույց են տվել, որ հիբրիդային ամպի շահագործման ժամանակ կատարվում է համակարգչային ռեսուրսների 16%, սերվերների 5-10% խնայողություն [37]: Երկու ամպերի միջև կապն իրականացվում է գաղտնագրված ալիքի միջոցով, որն ապահովում է դրա անվտանգությունը: Հիբրիդային ամպն ապահովում է գրանցման/ընթերցման բարձր արագություն, ի տարբերություն հանրային ամպի [59]:

Հիբրիդային ամպի կառուցվածքում ընդգրկված ամպերն իրարից անկախ համակարգեր են, որոնք փոխկապակցվում են ծրագրային լուծումով: Փոխկապակցված աշխատանքի ապահովումն իրականացվում է առաջադրանքների բաշխման (task scheduling) միջոցով [54, 55]:

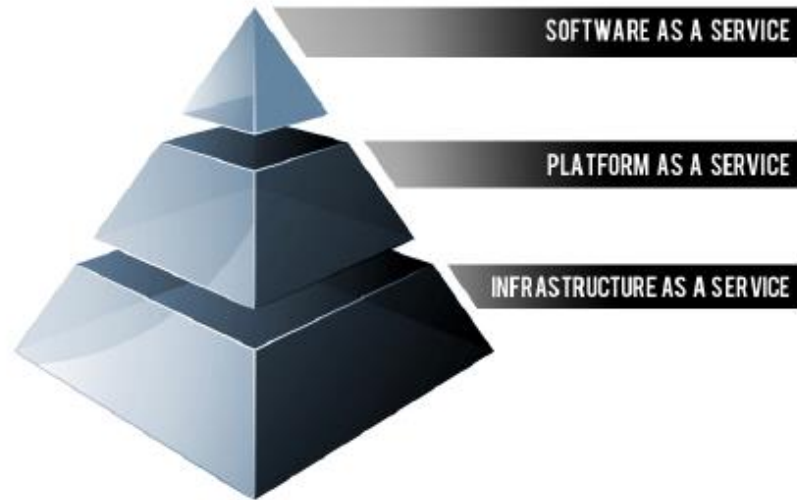
Հիբրիդային ամպի շահագործման ժամանակ մեծ դեր ունի համակարգի բեռնվածության կարգավորումը (load balancing), որպեսզի վերջինիս մեկ մասնիկի խափանման դեպքում ամբողջ համակարգն անաշխատունակ չդառնա [34, 50, 68]: Հաշվի առնելով, որ հիբրիդային ամպում հանրային ամպի բաժինը տրամադրվում է վճարովի հիմունքներով, այս տիպի շահագործման ժամանակ, բնականաբար, հարկավոր է վերլուծել նաև ֆինանսապես ավելի նպատահարմար հանրային ամպի ընտրության խնդիրը: Վերջինիս լուծման համար գոյություն ունեն տարբեր ալգորիթմներ [31, 38, 45]:

Համայնքային ամպ (Community cloud): Ամպային տեխնոլոգիայի այս տարբերակը կիրառվում է մի քանի կառույցների կողմից, որոնք կիրառում են նույն ենթակառուցվածքը՝ ելնելով այն հանգամանքից, որ ամպում գտնվող բոլոր կազմակերպություններն ունեն նույն պահանջները (անվտանգություն, կանոնադրություն և այլն): Համայնքային ամպը կարող են կառավարել ինչպես իր մեջ գտնվող կազմակերպությունները, այնպես էլ ծառայությունը մատուցող մեկ այլ ընկերություն: Այս տարբերակում բոլոր ծախսերը բաժանվում են կազմակերպությունների միջև [20]:

1.1.3. Ամպային տեխնոլոգիան ըստ տրամադրված ծառայության

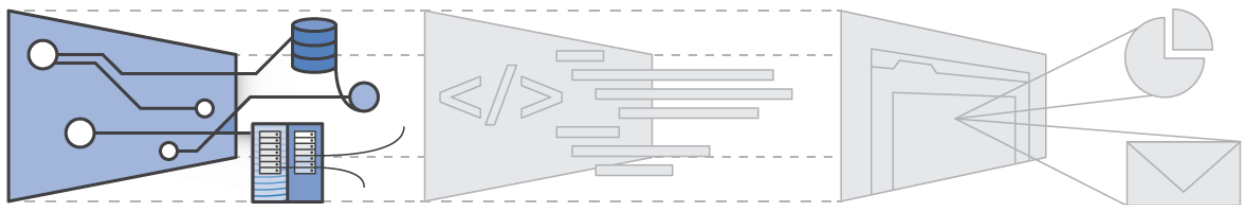
Ըստ տրամադրված ծառայության գոյություն ունեն ամպային տեխնոլոգիայի երեք հիմնական տիպեր (նկ. 1.2) [83]՝

- IaaS (Infrastructure as a service) – ենթակառուցվածքը որպես ծառայություն,
- SaaS (Software as a service) – ծրագրային ապահովումը որպես ծառայություն,
- PaaS (Platform as a service) – հարթակը որպես ծառայություն:



Նկ. 1.2. Ամպային ծառայությունները ըստ տրամադրման տիպերի [83]

IaaS տիպի դեպքում տրամադրված ծառայություններն են սերվերները, հիշողությունը, ցանցային և օպերացիոն համակարգերը [62]: Ընկերությունները սերվերներ, ծրագրեր և այլն գնելու փոխարեն ծառայությունը մատուցող ընկերությունից գնում են իրենց համակարգերի համար անհրաժեշտ փաթեթը (նկ. 1.3):



Նկ. 1.3. IaaS տիպի կառուցվածքը [94]

Ծառայությունը մատուցող ընկերությունը պատվիրատուին է տրամադրում վիրտուալ միջավայր, որտեղ հնարավոր է ստեղծել վիրտուալ մեքենաներ, որոնց վրա այնուհետև տեղադրվում են օպերացիոն համակարգերը: Այսպիսով՝ այսպիսի ծառայություն մատուցելիս վիրտուալացումն ունի առաջնային դեր: Վիրտուալացումն ապահովվում է հիպերվայզորների միջոցով, որոնցից են՝ Xen [84, 22], Oracle Virtual Box [85], KVM [86], VMware [87] կամ Microsoft ընկերության կողմից տրամադրվող Hyper-V-ն [26]: Օգտատիրոջ կողմից ստեղծվող վիրտուալ մեքենաները տեղադրվում են հիպերվայզորի վրա, որպես, այսպես կոչված, «հյուր» օպերացիոն համակարգեր: Ստացվում է, որ IaaS-ը ավելի մեծ ճկունությամբ հոստինգի տարբերակ է: Ծառայության մեջ ներառվում են նաև այնպիսի հնարավորություններ, ինչպիսիք են ցանցային հասանելիությունը (network access), երթուղավորումն ու հիշողությունը (storage): Ծառայությունը մատուցող ընկերությունն օգտատիրոջը տրամադրում է ֆիզիկական սարքեր և դրանց կառավարումը: Սովորաբար ծառայությունը մատուցող ընկերությունը օգտատիրոջ պահանջով տրամադրում է նաև ընդլայնման հնարավորություն: Ընդլայնման մեջ կարող են մտնել ինչպես հիշողության ծավալի ավելացումը, այնպես էլ սերվերների թողունակության բարձրացումը [51]:

IaaS ամպային տեխնոլոգիայի հատկանիշներն [51].

- պարզեցված համակարգչային ծառայություն և վճարման եղանակների ճկունություն,
- ադմինիստրատիվ կարգավորումների ավտոմատացում,
- դինամիկ ընդլայնման հնարավորություն,
- վիրտուալացում:

Ծառայությունից օգտվելու համար անհրաժեշտ են հետևյալ քայլերը.

1. ծառայությունը մատուցող ընկերությունն օգտատիրոջը հատկացնում է իր կողմից պահանվող ռեսուրսները,

2. օգտատերը ստեղծում է նոր վիրտուալ մեքենա այդ միջավայրում,
3. տեղադրվում է օպերացիոն համակարգը վիրտուալ մեքենայի վրա,
4. օգտատերը տեղադրում է ծրագրեր և ծառայություններ,
5. ծառայությունը մատուցող ընկերությունն ապահովում է ֆիզիկական սարքերի աշխատունակությունը:

Չնայած նրան, որ ծառայությունը մատուցողը պարտավորվում է ապահովել համակարգի առողջ և կայուն աշխատանքը, դրա վրա տեղակայված ծրագրերի ու ծառայությունների մոնիտորինգի և աշխատանքային վիճակի ապահովումը օգտատիրոջ պարտականությունն է:

Գոյություն ունեն մի շարք կազմակերպություններ, որոնք տրամադրում են IaaS տիպի ամպային տեխնոլոգիաներ, որոնցից ամենամեծ ճանաչում ունեն Google Compute Engine-ը, Rackspace Open Cloud-ը, IBM SmartCloud Enterprise-ը, Amazon Web Services-ը և Microsoft Azure-ը: Այս համակարգերից յուրաքանչյուրն ունի իր առանձնահատկությունները:

Google Compute Engine-ը Google Cloud Platform-ի IaaS ծառայություն մատուցող հատվածն է և կառուցված է Google-ի գլոբալ ենթակառուցվածքի վրա, որի վրա աշխատում են նաև Google Search-ը, Gmail և YouTube ծառայությունները: Ինչպես յուրաքանչյուր IaaS ծառայություն մատուցող համակարգ, այն տրամադրում է վիրտուալ մեքենաներ, որոնց վրա օգտատերը կարող է տեղադրել իր նախընտրած օպերացիոն համակարգերը: Համակարգն ունի ընդլայնման բարձր հնարավորություններ և բարձր թողունակություն: Այն աշխատում է վճարովի հիմունքներով, որտեղ օգտատերը վճարում է վիրտուալ մեքենայի շահագործման ժամանակի համար: Համակարգը որպես հիպերվայզոր շահագործում է KVM-ը [88]:

Amazon Web Services -ը պատկանում է Amazon ընկերությանը: Համակարգի կողմից IaaS ծառայություն մատուցվում է 13 տարբեր տարածաշրջաններից:

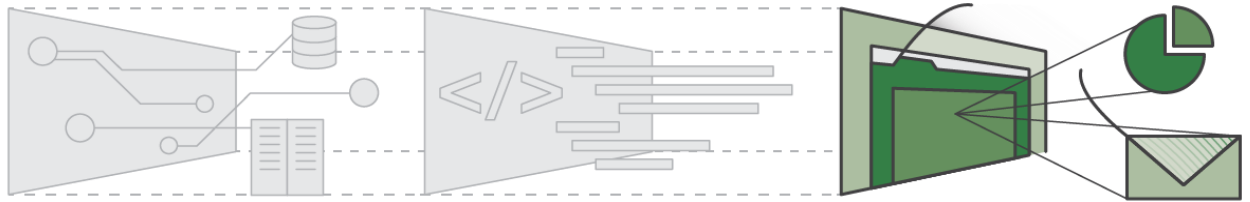
Համակարգի ամենահայտնի տրամադրվող ծառայություններից են Amazon Elastic Compute Cloud-ը, որը հայտնի է նաև «EC2» անվանմամբ և Amazon Simple Storage Service -ը, որը անվանում են նաև «S3» [94]:

Microsoft Azure ամպային տեխնոլոգիան տրամադրում է ավելի քան 50 տարբեր ծառայություններ և, ինչպես այլ համակարգեր, օգտատիրոջն է տրամադրում վիրտուալ մեքենաների ստեղծման հնարավորություն: Համակարգը հնարավորություն է տալիս կիրառել ինչպես Microsoft-ի կողմից ստեղծված ծրագրային լուծումները, այնպես էլ այլ ընկերությունների կողմից ստեղծված օպերացիոն համակարգերն ու ծրագրերը: Microsoft Azure-ը կիրառում է իր իսկ համար ստեղծված Microsoft Azure օպերացիոն համակարգը, որը նկարագրվում է որպես համակարգի «ամպային մակարդակ», աշխատում է Windows-ի մի շարք սերվերային համակարգերի և Hyper-V-ի ձևափոխված տարբերակի՝ Microsoft Azure Hypervisor-ի վրա [89]: Համակարգի ընդլայնումն ու հուսալիությունն իրականացվում են Microsoft Azure Fabric Controller-ի միջոցով:

Այսպիսով, IaaS տիպի ամպային ծառայությունն ապահովում է ֆինանսական խնայողություններ, համակարգչային բարձր թողունակություն, հիշողություն, իսկ ցանցային կարգավորումները կատարում է ծառայությունը մատուցող ընկերությունը:

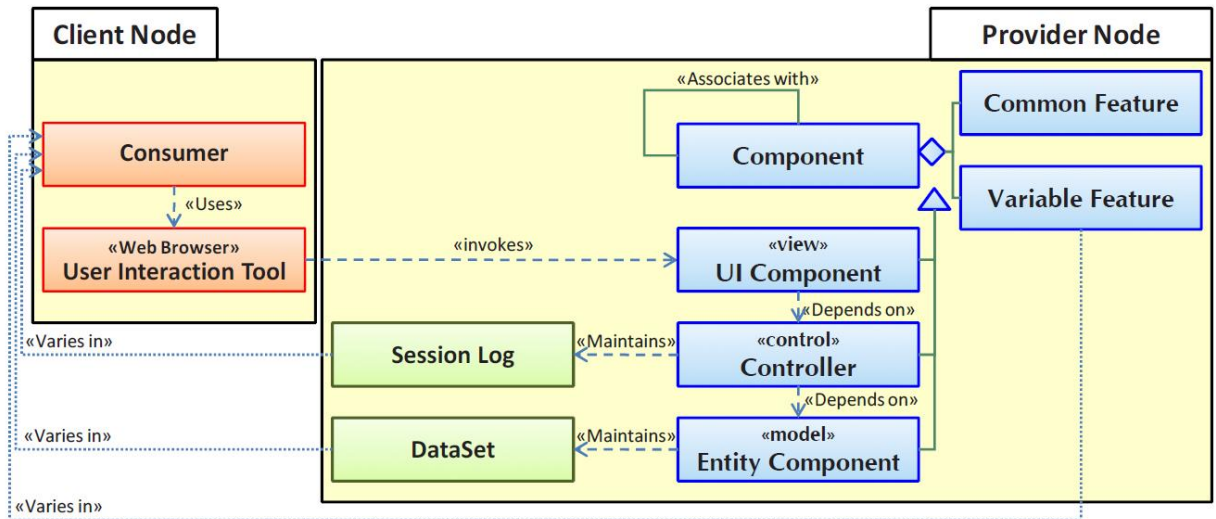
SaaS (Software as a Service): Այս տարբերակը ծրագրային լիցենզավորման և տրամադրման համակարգ է, որտեղ ծրագրային համակարգը լիցենզավորվում է բաժանորդագրության հիմունքներով և տրամադրվում կենտրոնացված սերվերից [51]: Այս տիպի ամպը սովորաբար հասանելի է լինում վեբ զննիչի (browser) միջոցով, որը օգտատիրոջ համար հանդիսանում է «thin client»: Այս տիպը մեծ տարածում ունեցող մոդել է բիզնես լուծումների, գրասենյակային ծրագրերի, նամակագրության համակարգերի, ղեկավարման համակարգերի, վիրտուալացման, հաշվապահական ծրագրերի և այլնի համար (նկ. 1.4):

SaaS ամպային տեխնոլոգիան ունի երկու հիմնական բաղադրիչ՝ Client Node և Provider Node [35]: Սովորաբար, Client Node-ն ունենում է միայն մեկ ինտերֆեյս, որից օգտատերը կարողանում է շահագործել համակարգը: Ինտերֆեյսը արտապատկերվում է վեբ բրաուզերի միջոցով: Provider Node-ը ծառայությունը մատուցող ընկերության կողմն է, որն ունի երեք մակարդակ՝ UI Component որպես տեսանելի հատված, Controller, որի միջոցով կատարվում են կարգավորումները և Entity Component, որը ներկայացնում է համակարգի մոդելը [35]:



Նկ. 1.4. SaaS տիպի կառուցվածքը [94]

Controller-ը կազմակերպում ու կառավարում է համակարգի բիզնես տրամաբանությունն ու տեղաշարժերը: Այդ իսկ պատճառով այն ունի Session Log, որտեղ գրանցվում է հաճախորդի հետ կապված փոխհարաբերությունների մասին ինֆորմացիան: SaaS համակարգի ավելի մանրամասն մետամոդելը ներկայացված է հետևյալ նկարում (նկ. 1.5)՝



Նկ. 1.5. SaaS տիպի մանրամասն կառուցվածքը [35]

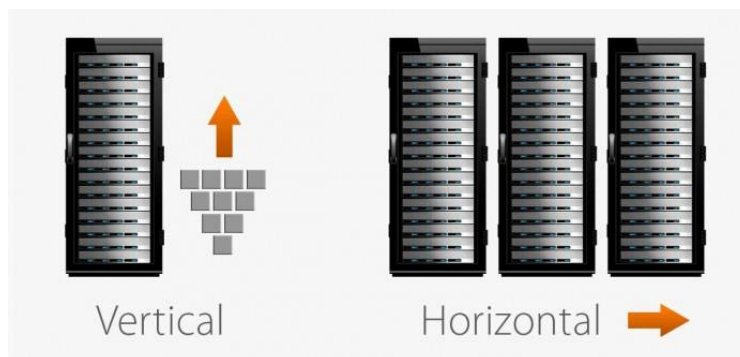
SaaS ամպային տեխնոլոգիայի հիմնական հատկանիշներն են [35]՝

- կոմերցիոն ծրագրերի վեբ հասանելիություն,
- ծրագրի կենտրոնացված կառավարում,
- փոքր տարբերություններով համակարգի մատուցում որպես մեկ մոդել բոլորի համար,
- օգտատերերի կողմից ծրագրային սպասարկման կամ թարմացումների բացառում,
- տարբեր ծրագրերի ինտեգրում Application Programming Interfaces (APIs)-ների միջոցով:

SaaS համակարգի ամենակարևոր հատկանիշը կենտրոնացված կառավարումն է, երբ օգտատերն ընթացիկ խնդիրների հետ չի առնչվում և թարմացումներ չի կատարում: Ելնելով այս հանգամանքից՝ թարմացումների փուլերը սովորաբար ավելի արագ են կատարվում, քանի որ չկա դրանք օգտատիրոջ համակարգչի տեխնիկական պարամետրերի հետ համապատասխանեցնելու անհրաժեշտություն:

Հորիզոնական ընդլայնումից բացի գոյություն ունի նաև ուղղահայաց ընդլայնում [61]: Այս պարագայում ավելացվում կամ փոխվում են ոչ թե սերվերները կամ համակարգը սպասարկող այլ սարքեր, այլ կատարվում են դրանց ռեսուրսների փոխարինում, ինչպիսիք են օրինակ պրոցեսորները, օպերատիվ հիշող սարքերը և այլն: Այսպիսի ընդլայնման տարբերակը հնարավորություն է տալիս ավելի շահավետ կիրառել վիրտուալացումը, քանի որ ռեսուրսների ավելացման պարագայում հնարավոր է բարձրացնել արդեն իսկ շահագործվող օպերացիոն համակարգերի աշխատանքի արդյունավետությունը:

Ընդլայնման երկու տարբերակներն էլ ունեն իրենց առավելություններն ու թերությունները: Հորիզոնական ընդլայնման առավելությունն այն է, որ շատ ծրագրեր ուղղակիորեն նախագծված չեն լինում այնպես, որ կարողանան շահագործել համակարգի գործունեության ընթացքում ավելացված ռեսուրսները և անհրաժեշտ է լինում դրա բաժանումը նոր սարքերի մեջ: Սակայն նոր սարքերի ավելացումը նշանակում է դրանց կառավարման և ընթացիկ սպասարկման ավելի մեծ ծավալներ, բացի այդ առաջանում են նաև փոքր ուշացումներ սարքերի միջև հաղորդակցման ժամանակ: Ուղղահայաց ընդլայնման առավելությունը կայանում է դրա ավելի հեշտ իրականացման և սպասարկման մեջ և, բացի այդ, վիրտուալացման զարգացման հետ մեկտեղ առկա ռեսուրսների հզորացումը ավելի բարձր թողունակություն է ապահովում (նկ. 1.6):



Նկ. 1.6. Ընդլայնման եղանակներ [61]

Համակարգի այն առավելությունը, որ այն գրեթե նույն տեսքով մատուցվում է մեծ քանակի օգտատերերի (ֆիզիկական սարքեր, ցանց և օպերացիոն համակարգ) հնարավորություն է տալիս շատ կարճ ժամանակահատվածում ավելացնել օգտատերերի քանակը՝ կատարելով համակարգի ընդլայնում: Այն ընդլայնումը, որի ժամանակ ավելացվում են համակարգը սպասարկող մեքենաների քանակը կոչվում է հորիզոնական ընդլայնում [66]: Եթե համակարգը սպասարկող ֆիզիկական սերվերի թողունակությունն արդեն չի բավարարում օգտատերերի քանակին և պահանջներին, ընկերությունն ավելացնում է սպասարկող սերվերների քանակը: Սակայն օգտատերերն այս խնդիրների մասին չեն տեղեկանում, քանի որ ինչպես նախապես նշվել էր ծառայությունը մատուցվում է կենտրոնացված վայրից:

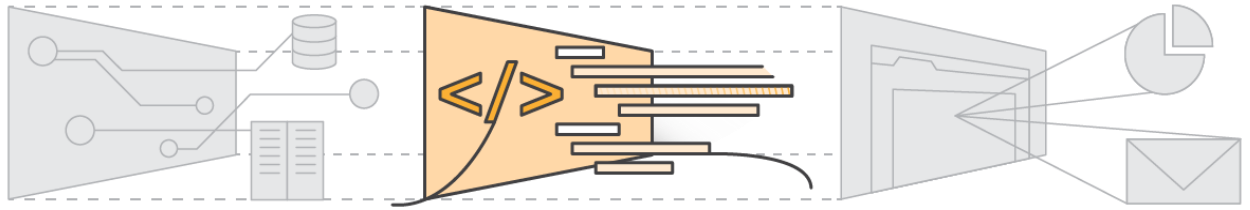
SaaS-ի կարևոր հատկանիշներից է նաև այն, որ օգտատերը հնարավորություն է ունենում կատարել կառուցվածքային փոփոխություններ, ինչպիսիք են օրինակ՝ իրեն տրամադրվող ծառայությունների պարամետրերի փոփոխությունը, ինտերֆեյսի տեսքի փոփոխությունները և այլն: Այս հատկանիշի կարևորությունը կայանում է նրանում, որ ծառայությունից օգտվող օգտատերը կարողանում է բրենդավորել մատուցվող ծառայությունը հատուկ իր ընկերության համար: Մեկ այլ կարևոր հատկանիշ է համակարգի թեստավորման կարճ ժամանակը, որն իր հերթին հնարավորություն է տալիս պահանջվող թարմացումներն ավելի արագ իրականացնել: Թեստավորման կարճ ժամանակը պայմանավորված է մատուցվող ծառայության ընդհանրական հիմքով, որտեղ յուրաքանչյուր օգտատիրոջը մատուցվող ծառայությունը հիմնված է մեկ ծրագրային լուծման վրա: Ծառայությունը մատուցող ըկերությունը հնարավորություն ունի նաև վերլուծել իր օգտատերերի վարքագիծը համակարգից օգտվելու ընթացքում, որի միջոցով կարգի համակարգի լավարկման կարիքները:

Չնայած այս բոլոր առավելություններին, SaaS-ն ունի նաև թերություններ, որոնք խոչընդոտում են դրա ավելի արագ զարգացումը: Քանի որ օգտատերերի

անձնական տվյալները պահպանվում են ծառայությունը մատուցող ընկերության սերվերներում, առաջանում են դրանց անվտանգության ապահովման խնդիրներ: Տվյալների ապահով պահպանումը երբեմն կարող է ոչ արդարացի կասկածներ ծնել, ինչի հետևանքով ընկերությունները խուսափում են նմանատիպ համակարգից: Բացի այդ, համակարգում տեղադրված ծրագրերից օգտվելու համար բնականաբար անհրաժեշտ է ինտերնետ հասանելիություն, իսկ ինֆորմացիայի գրանցման/ընթերցման ժամանակ հնարավոր են ուշացումներ, որոնք, կախված պատվիրատուի աշխատանքի բնույթից կարող են ընդունելի չլինել: Այն հանգամանքը, որ ինֆորմացիան տեղափոխվում է ինտերնետի միջոցով սահմանափակում է ծրագրի աշխատանքի թողունակությունը ինտերնետի արագության հետ, մինչդեռ ընկերության ներքին ցանցի թողունակությունը, բնականաբար, ավելի բարձր կարող է լինել: Այն առավելությունը, որ համակարգում տրամադրվող ծառայությունները բոլոր օգտատերերի համար ունենում են նույն ծրագրային հիմքը, ունի նաև իր թերությունը՝ արդեն իսկ առկա հնարավորությունների մեջ մեկ օգտատիրոջ համար հնարավոր չէ կատարել փոփոխություններ, քանի որ հնարավոր է, որ այլ օգտատերերի համար այդ փոփոխությունները նախընտրելի չլինեն:

SaaS ծառայությունը անվճար տարբերակներ գրեթե չունի: Եթե համակարգը չի հատկացվում ամսական վճարի դիմաց, ապա ծառայությունը ուղեկցվում է գովազդներով: Բացառություն են կազմում այնպիսի համակարգերը, ինչպիսիք են օրինակ Wordpress-ը, Drupal-ը և այլն:

PaaS - ը ամպային տեխնոլոգիայի տեսակ է, որը տրամադրում է հարթակ, որից օգտատերերը օգտվելով կարող են ստեղծել, շահագործել և թեստավորել ծրագրեր, առանց դրանց համար ֆիզիկական հանգույց ստեղծելու անհրաժեշտության (նկ. 1.7) [51]:



Նկ. 1.7. PaaS տիպի կառուցվածքը [94]

PaaS-ը նախապես ստեղծվել էր ծրագրավորողների աշխատանքը հեշտացնելու նպատակով, որպեսզի նրանք ժամանակ չկորցնեին ծրագրերի աշխատանքային միջավայրի կարգավորման վրա: Նախնական շրջանում բոլոր PaaS ծառայությունները մատուցվում էին հանրային ամպի տեսքով, սակայն ժամանակի հետ ստեղծվեցին նաև համակարգի մասնավոր և հիբրիդային տարբերակները, քանի որ շատ ընկերություններ չէին ցանկանում, որ իրենց տվյալները լինեն հանրային հասանելիության: Օգտատերը համակարգից օգտվելիս պատասխանատվություն է կրում միայն իր կողմից շահագործվող ծրագրի համար, իսկ ծառայությունը մատուցող ընկերությունն ապահովում է սերվերների, օպերացիոն համակարգերի, հիշողության, ցանցի և այլ կառավարման համակարգերի աշխատանքային վիճակը: Այս հանգամանքը ծառայության գլխավոր առավելությունն է, քանի որ օգտատերը միջավայրի բարդ ճարտարապետության ստեղծման համար ժամանակ չի կորցնում [51]:

PaaS-ի կիրառումը հարմարավետ է այն դեպքերում, երբ մի քանի ծրագրավորողներ միաժամանակ աշխատում են որևէ նախագծի վրա կամ այդ նախագծի պատվիրատուն ցանկանում է հետևել ծրագրի ստեղծման ընթացքին:

1.1.4. Տվյալների կառավարումն ամպային տեխնոլոգիաներում

Տվյալների կառավարումն առաջնային դեր ունի ամպի շահագործման ժամանակ: Հիշողությունը պետք է ոչ միայն լինի հուսալի և անվտանգ, այլ նաև ունենա ընդլայնման հնարավորություն առանց նախնական կառուցվածքի խաթարման: Մեծ չափսի տվյալների պահեստավորման, գրանցման/ընթերցման համակարգը պետք է կարողանա իրականացնել գործառույթներն արագ և առանց ընդհատումների:

Հասկանալի է դառնում, որ տվյալների պահպանման համար ավանդական լուծումները չեն կարող ապահովել անհրաժեշտ հուսալիությունն ու անվտանգությունը: Օբյեկտային կողմնորոշմամբ տվյալների պահպանման համակարգերը ֆայլային համակարգերից [67] տարբերվում են նրանով, որ դրանք ինֆորմացիան պահպանում են որպես օբյեկտներ [49], ի տարբերություն ֆայլային համակարգերի, որտեղ այն գրանցվում է բլոկերի մեջ [56]: Յուրաքանչյուր օբյեկտ իր մեջ պարունակում է տվյալ և հատուկ իդենտիֆիկացնող համար: Օբյեկտային կողմնորոշմամբ տվյալների պահպանման համակարգը կարող է կիրառվել մի շարք մակարդակներում, ինչպիսիք են համակարգային, ինտերֆեյսային մակարդակները: Յուրաքանչյուր դեպքում համակարգի նպատակն է այնպիսի միջոցների տրամադրումը, որոնք ֆայլային համակարգերը չեն կարող տրամադրել, օրինակ ծրագրերի կողմից ուղիղ ծրագրավորվող ինտերֆեյսները և այլն: Օբյեկտային կողմնորոշմամբ համակարգը մեծ տարածում ունի չկարգավորված տվյալների պահպանման համար: Այն լայնորեն կիրառվում է նաև մեծածավալ տվյալներ պահպանող ընկերությունների կողմից, ինչպիսիք են օրինակ Facebook-ը, Dropbox-ը և այլն [22]: Օբյեկտային կողմնորոշմամբ տվյալների պահպանման համակարգի ճարտարապետության հիմքում ընկած է ծրագրերի կողմից ցածր մակարդակների փակումը՝ այսպիսով տվյալները ցույց տալով որպես օբյեկտներ, ոչ թե բլոկներ: Համակարգի մեկ այլ կարևոր հատկանիշ է ինդեքսավորումը, երբ յուրաքանչյուր

օբյեկտ պարունակում է իրեն յուրահատուկ կցված համարը, որը մեծապես ազդում է ինֆորմացիայի գրանցման/ընթերցման արագության վրա: Համակարգը նաև առանձնացնում է տվյալները և դրանց մետատվյալները (անունը, ստեղծման ամսաթիվը, տիպը և այլն)՝ ապահովելով հետևյալ հնարավորությունները.

- ծրագրին կամ օգտատիրոջն անհրաժեշտ ինֆորմացիայի կորզում բարելավված ինդեքսավորման համար,
- տվյալների կառավարման կանոնների ապահովում,
- նոդերի և տվյալների պահպանման տարբեր սերվերների կենտրոնացված կառավարում,
- մետատվյալների պահպանման օպտիմալացում:

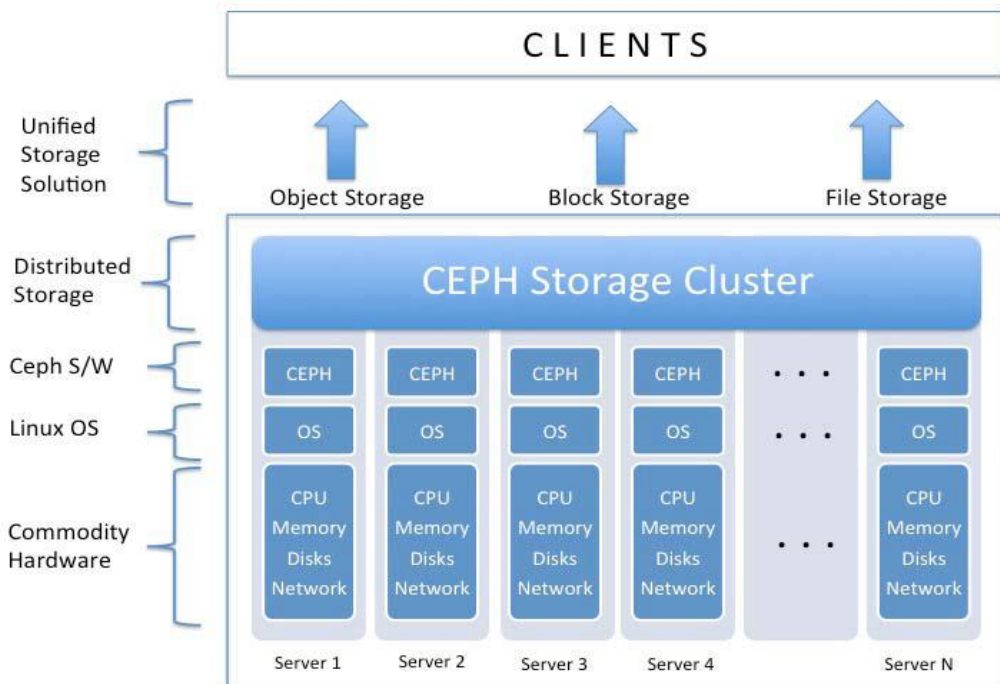
Ինչպես նախապես նշվել է, օբյեկտային կողմնորոշմամբ տվյալների պահպանման համակարգն ընձեռում է ծրագրավորման ինտերֆեյսներ, որոնք ավելի ճկուն են դարձնում տվյալների կառավարումն ու պահեստավորումը:

Օբյեկտային կողմնորոշմամբ տվյալների պահպանման համակարգերը մեծ դեր ունեն ամպային տեխնոլոգիաների շահագործման մեջ: Հաշվի առնելով, որ ամպի հիմնական բաղադրիչը տվյալների ապահով պահպանումն է, ապա պարզ է դառնում, որ ավանդական լուծումները չեն կարող ապահովել պահանջվող արդյունքը: Հայաստանի ազգային պոլիտեխնիկական համալսարանում ամպի կառուցումն ու շահագործումն իրականացնելուց առաջ կատարվել է համեմատություն օբյեկտային կողմնորոշմամբ տվյալների երկու համակարգերի միջև՝ առկա խնդրի համար լավագույնն ընտրելու նպատակով: Դրանք են բաց ծրագրային կոդով Ceph [90] և GlusterFS [91] համակարգերը:

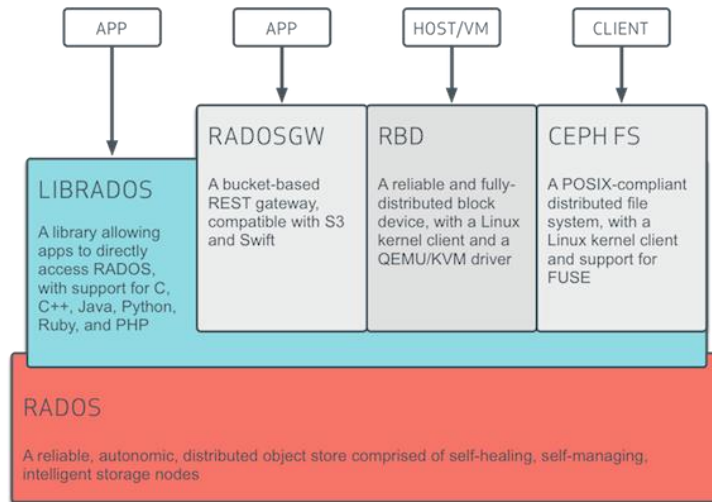
Ceph օբյեկտային կողմնորոշմամբ տվյալների պահպանման համակարգի հիմքում օբյեկտային պահպանման սկզբունքն է, որը կոչվում է Rados: Որպես օբյեկտային պահպանմամբ համակարգ Ceph-ը տվյալները պահպանում է բինարային օբյեկտների տեսքով՝ առաջիկայում, անհրաժեշտության դեպքում, դրանց ծավալն ավելի փոքրացնելու հնարավորությամբ: Սովորական տվյալների պահպանման

սարքերի խնդիրը կայանում է նրանում, որ երբ տվյալները գրանցվում են բլոկների մեջ, այդ իսկ տվյալներն առաջիկայում ընթերցելու համար անհրաժեշտ է լինում նախ ընթերցել ամբողջ սարքը և նոր միայն այդ տվյալը գտնելուց հետո ընթերցել այն: Այս գործընթացը բավականին երկար ժամանակ է պահանջում, ի տարբերություն օբյեկտային պահպանման եղանակի: Ceph-ը խուսափում է այս խնդրից ավելացնելով մի մակարդակ, որը կառավարում է բլոկները: Այս մակարդակը կոչվում է RADOS: Ինքն իրեն մակարդակը բաղկացած է երկու մասից, որոնցից առաջինը Object Store Device (OSD)-ն է: Վերջինիս գաղափարն այն է, որ յուրաքանչյուր սկավառակ պատկանում է տվյալների համակարգի կլաստերին (նկ. 1.8):

Երբ օգտատերը գրում կամ կարդում է որևէ տվյալ, ամբողջ գործառույթն իրականացվում է այս մասի կողմից: Երկրորդ մասը մոնիտորինգի համակարգն է, որը կառավարում է OSD-ները: Գաղափարը պարզ է՝ նշանակություն չունի, թե քանի սերվեր կամ կոշտ սկավառակ է կիրառվում տվյալների պահպանման համար: Օգտատերը տեսնում է միայն դրա ինտերֆեյսը, որը միշտ կլինի նույնը (նկ.1.9):



Նկ. 1.8. Ceph-ի կլաստերային համակարգ [43]

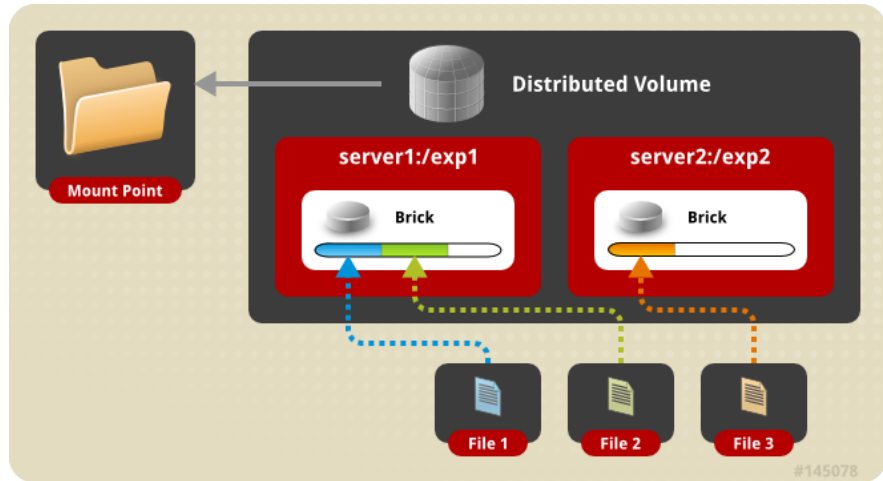


Նկ. 1.9. Ceph-ի ճարտարապետությունը [92]

Բացի կենտրոնացված սերվեր ունենալու անհրաժեշտությունից, Ceph-ին հարկավոր է, որ իր կլիենտները տեղադրված լինեն հիպերվայզորների վրա, որպեսզի այն կարողանա գրանցել/ընթերցել ցանցային տվյալների պահպանման համակարգում: Հիպերվայզորի վրա տվյալներ գրանցելու/ընթերցելու ժամանակ Ceph-ի կենտրոնական սերվերը դիմում է վերոհիշյալի վրա տեղադրված իր կլիենտին, հաստատում կապ, այնուհետև կատարում անհրաժեշտ փոփոխությունները: Համակարգում բոլոր փոփոխությունները կատարվում են front-end-ում, առանց ֆիզիկական մասերի հետ աշխատելու կարիքի:

GlusterFS օբյեկտային կողմնորոշմամբ տվյալների պահպանման համակարգը: Ինչպես Ceph-ը, GlusterFS-ը նույնպես բաց ծրագրային կոդով համակարգ է: Համակարգը նման է NAS-ին նրանով, որ այն կապում է ցանցում գտնվող տվյալների պահպանման սերվերները մեկը մյուսին՝ ստեղծելով մեկ ընդհանուր տվյալների պահպանման համակարգ, որտեղ յուրաքանչյուր սերվեր իր volume-ներից մեկն է: Համակարգը ևս ունի կլիենտներ, որոնք տեղադրվում են առանձին տվյալների պահպանման սերվերների վրա, իսկ կենտրոնացված կառավարման սերվերն

ապահովում կլիենտների կարգավորումները և դրանց ընդհանրացված աշխատանքը: GlusterFS-ի աշխատանքի սկզբունքը ընկած է ֆայլային mirroring-ի, stripping-ի և replication-ի վրա: Համակարգը ունի նաև հզոր load balancing-ի բաղադրիչ, որը կարգավորում է վերջինիս ծանրաբեռնվածությունները (1.20):



Նկ. 1.20. GlusterFS-ի ճարտարապետություն [93]

Համակարգի հիմնական գործառույթն իրականացնում են այսպես կոչված «թարգմանիչները», որոնք տեղակայված են կլիենտների վրա, որոնք կառավարում են սերվերի վրա տեղադրված տվյալները: Համակարգը, ի տարբերություն Ceph-ի, որը կիրառում է մետատվյալներ տվյալների պահպանման ժամանակ, աշխատում է հեշինգի ալգորիթմի հիման վրա:

1.3. Եզրակացություններ 1-ին գլխի վերաբերյալ և ատենախոսության նպատակները

Ուսումնասիրվել են ամպային տեխնոլոգիաները, դրանց դերը ժամանակակից տեղեկատվական ցանցերում, կազմակերպություններում և այլն: Ուսումնասիրվել են

ամպային տեխնոլոգիաների հիմնական տիպերը և դրանց կողմից մատուցվող ծառայությունների տարատեսակները, ամպային տեխնոլոգիայի տրամադրման համար առկա գործիքամիջոցները և դրանց միջև համատեղ աշխատանքի կազմակերպման հնարավոր լուծումները:

Կատարված ուսումնասիրությունների հիման վրա կարելի է եզրակացնել, որ միջինից մեծ կազմակերպության համար, որտեղ ամպը շահագործվելու է տարբեր տիպի լուծումներ տրամադրելու համար՝ ամենահարմարը հիբրիդային ամպի շահագործումն է, վերջինիս ճկունության շնորհիվ: Բացի այդ, շատ կարևոր է ապահովել ամպի անվտանգությունը, անխափան աշխատանքը և տվյալների ապահով պահուստավորումը: Տվյալների ապահով պահպանման համար նպատակահարմար է կիրառել մեծածավալ տվյալների գրանցման/ընթերցման համար նախատեսված համակարգերը, որոնց մեծաքանակ և ճկուն գործիքամիջոցների շնորհիվ հնարավոր է դառնում եղած ինֆորմացիայի ոչ միայն ապահով պահուստավորումը, այլ նաև վերջինիս գրանցման/ընթերցման բարձր արագության ապահովումը:

Այսպիսով, աշխատանքի նպատակը հիբրիդային ամպի ստեղծման ճանապարհով տեղեկատվատեխնոլոգիական ենթակառուցվածքների ու ծառայությունների տրամադրման ավտոմատացված համակարգի մշակումն ու ներդրումն է: Համակարգում ռեսուրսների բաշխումը պետք է իրականացվի ընթացիկ վիճակը բնութագրող պարամետրերի և օգտատերերի պատվերների վերլուծության հիման վրա ինտեգրալ գնահատականների կիրառմամբ: Օգտատերերի համար պետք է մշակվի հարմարավետ ինտերֆեյս նախագիծ ստեղծելու, ղեկավարելու, փոփոխելու և պահպանելու համար:

ԳԼՈՒԽ 2. ՏԵՂԵԿԱՏՎԱԿԱՆ ԾԱՌԱՅՈՒԹՅՈՒՆՆԵՐԻ ՏՐԱՄԱԴՐՄԱՆ ԱՎՏՈՄԱՏԱՑՎԱԾ ՀԱՄԱԿԱՐԳԻ ՄԱԹԵՄԱՏԻԿԱԿԱՆ ՈՒ ՏԵՂԵԿԱՏՎԱԿԱՆ ԱՊԱՀՈՎՈՒՄՆԵՐԸ

2.1. Համակարգի մաթեմատիկական ապահովումը

Առաջին գլխում կատարված ուսումնասիրությունները ցույց են տալիս, որ յուրաքանչյուր մեծ ընկերության տեղեկատվական կարիքները հոգալու համար ամպային ծառայությունների տրամադրումը կարևոր դեր ունի: Սույն աշխատանքի նպատակն է Հայաստանի ազգային պոլիտեխնիկական համալսարանում IaaS տիպի ծառայություններ մատուցող ավտոմատացված համակարգի մշակումը հիբրիդային ամպի կառուցման ճանապարհով: Հիբրիդային ամպի բնույթից ելնելով վերջինիս շահագործման համար անհրաժեշտ է ընտրել հանրային ամպի ծառայությունները մատուցող համակարգ: Որպես հանրային ամպ շահագործվել են Amazon ընկերության տրամադրած վեբ սերվիսները՝ Amazon Web Services [94]: Ընկերությունը տրամադրում է գործիքամիջոցների մեծ խումբ, որը կարող է շահագործվել տարբեր տիպի ծառայությունների մատուցման համար: Բացի այդ, գրանցումից սկսած մեկ տարվա ընթացքում ծառայությունները մատուցվում են անվճար հիմունքներով [95]: Համալսարանում ստեղծված մասնավոր ամպի հետ Amazon Web Services-ը կապվում է իր իսկ կողմից տրամադրվող API-ների միջոցով [96]: Տեղեկատվական ռեսուրսների ու ծառայությունների տրամադրման համար մշակված ավտոմատացված համակարգը պատվիրվող նախագծի և առկա ռեսուրսների ընթացիկ վերլուծության արդյունքում կայացնում է վճիռ նախագիծը մասնավոր կամ հանրային ամպում բաշխելու մասին:

Հիբրիդային ամպում ռեսուրսների ավտոմատացված բաշխումը կազմակերպելու նպատակով ընտրվել են հետևյալ գործոնները՝

- Պրոցեսորների քանակ (P),

- RAM-ի ծավալ (R),
- Հիշողության ծավալ (V),
- Ցանցային ադապտորների քանակ (A),
- Անվտանգության ցուցանիշ (S),
- Նախագծի տեսակը (N),
- Կիրառման հաճախականություն (W):

Պրոցեսորների քանակ: Պրոցեսորների քանակի ընտրությունն ունի կարևոր դեր համակարգում բաշխման համար: Եթե համակարգը ստուգում է առկա իրադրությունը և վերլուծում, որ ռեսուրսները բավարար չեն ծառայությունը մասնավոր ամպում մատուցելու համար ապա ծառայությունը տրամադրվում է հանրային ամպի միջոցով:

RAM-ի ծավալ: Այս ցուցանիշի համար գործում է նույն տրամաբանությունը, ինչ կիրառվում է պրոցեսորների համար:

Ցանցային ադապտորների քանակ: Աշխատում է նույն տրամաբանությամբ ինչ պրոցեսորների քանակը և RAM-ի չափսը:

Հիշողության տարածք: Ֆայլի ծավալը մեծ դեր ունի հիբրիդային ամպի տրամադրման մեջ: Եթե համակարգում տեղադրվող ֆայլի ծավալը բավականին մեծ է, ապա հասկանալի է դառնում, որ ցանկալի է այն տեղադրել լոկալ ռեսուրսների վրա, որպեսզի ցանցի վրա մեծ ծանրաբեռնվածություն չառաջանա: Սակայն այս դրույթը կարող է փոխվել այն ժամանակ, երբ տեղադրված ֆայլը հաճախ չի օգտագործվելու օգտատիրոջ կողմից: Այս պարագայում այն կարող է տեղադրվել նաև հանրային ամպի վրա:

Անվտանգության ցուցանիշ: Այս ցուցանիշն ընտրվում է օգտատիրոջ կողմից, այնուհետև համակարգն իրկանացնում է վերլուծություն ու որոշում այս ցուցանիշի կշիռը: Համակարգն առաջարկում է ընտրել անվտանգության հետևյալ ցուցանիշները՝

- ցածր (1),

- ցածր-միջին (2),
- միջին (3),
- միջին-բարձր (4),
- բարձր (5):

Նախագծի տեսակը: Կախված իրականացվող աշխատանքի բնույթից համակարգը կարող է խնայել ներքին ռեսուրսները և ծառայությունը տրամադրել հանրային ամպի միջոցով: Իհարկե վերոհիշյալ պայմանը, ինչպես մնացած բոլոր պայմանները, համադրվում են, որից հետո միայն կայացվում է վճիռ բաշխման վերաբերյալ: Նախագծի տեսակներն օգտատիրոջն առաջարկվում է ընտրել հետևյալ ցանկից՝

- ազատ բնույթի, կարճաժամկետ աշխատանքներ (1),
- ուսանողական նախագծեր, ավարտական աշխատանքներ (2),
- ատենախոսական հետազոտություններ (3),
- փորձարարական աշխատանքներ (4),
- ներքին օգտագործման փաստաթղթերի պահպանում և վերլուծություն (5),
- գիտահետազոտական բնույթի աշխատանքներ (6):

Կիրառման հաճախականություն: Եթե ամպում տեղադրված տվյալներն ունեն կիրառման բարձր հաճախականություն, ապա այս պարագայում ևս ցանկալի է դրանց տեղադրումը մասնավոր ամպում՝ ցանցային ծանրաբեռնվածությունից խուսափելու համար: Այս ցուցանիշն օգտատիրոջ կողմից չի սահմանվում, քանի որ մինչև նախագծի շահագործումը օգտատերը դժվար թե հստակ պատկերացում ունենա ցուցանիշի մասին: Համակարգը ցուցանիշի նշանակությունը ստանում է նախկին նմանատիպ նախագծերից հավաքագրված տվյալների վերլուծության և պատվիրված նախագծի շահագործման ընթացքում կատարված վերլուծության հիման վրա:

Ավտոմատացված համակարգում յուրաքանչյուր նոր նախագիծ ստեղծելիս բաշխումն իրականացնելու նպատակով նախ որոշվում են մասնավոր ամպում առկա ռեսուրսների սահմանային արժեքները՝

- Պրոցեսորների առավելագույն քանակը (P_{max}),
- RAM-ի առավելագույն ծավալը (R_{max}),
- Հիշողության առավելագույն ծավալը (V_{max}),
- Ցանցային ադապտորների առավելագույն քանակը (A_{max}),

Այնուհետև, յուրաքանչյուր գործոնի համար որոշվում են գնահատականները հետևյալ կերպ (աղյ. 2.1).

Գնահատականները հաշվելուց հետո, յուրաքանչյուր գործոնի համար կշռային գործակիցների միջոցով [3, 5, 7, 10-16] կարող ենք ստանալ ինտեգրալ գնահատական, որն էլ թույլ կտա վճիռ կայացնել հիբրիդ ամպի որևէ բաղադրիչում նախագիծը ստեղծելու ու ծառայություններ մատուցելու մասին: Մի քանի սցենարներով փորձարկումների կատարումից հետո առաջարկվում են գործոնների աղյ. 2.2.-ում ներկայացված նախընտրելի կշռային գործակիցները, բավարարելով $\alpha_P^0 + \alpha_R^0 + \alpha_V^0 + \alpha_A^0 + \alpha_S^0 + \alpha_N^0 = 1$ պայմանը:

Աղյուսակ 2.1.

Գործոնների գնահատականները

Գործոնը	Գնահատականը	
	Մասնավոր ամպ	Հանրային ամպ
Պրոցեսորների քանակ (P)	$1 - \frac{P}{P_{max}}$	$\frac{P}{P_{max}}$
RAM-ի չափս ծավալ (R)	$1 - \frac{R}{R_{max}}$	$\frac{R}{R_{max}}$
Հիշողության տարածք (V)	$\frac{V}{V_{max}}$	$1 - \frac{V}{V_{max}}$
Ցանցային ադապտորների քանակ (A)	$1 - \frac{A}{A_{max}}$	$\frac{A}{A_{max}}$
Անվտանգության ցուցանիշ (S)	$\frac{S}{5}$	$1 - \frac{S}{5}$
Նախագծի տեսակը (N)	$\frac{N}{6}$	$1 - \frac{N}{6}$

Գործոնների կշռային գործակիցները

Գործոնը	Կշռային գործակիցը
Պրոցեսորների քանակ (P)	$\alpha_P^0=0.15$
RAM-ի չափս ծավալ (R)	$\alpha_R^0=0,15$
Հիշողության տարածք (V)	$\alpha_V^0=0,15$
Ցանցային ադապտորների քանակ (A)	$\alpha_A^0=0,1$
Անվտանգության ցուցանիշ (S)	$\alpha_S^0=0,3$
Նախագծի տեսակը (N)	$\alpha_N^0=0,15$

Համակարգի աշխատանքի ծանրաբեռնվածությունից ելնելով գործոնների նախընտրելի արժեքները կարող են փոփոխվել: Դրանց օպտիմալ արժեքները որոշելու համար առաջարկվում է ոչ գծային ծրագրավորման հետևյալ խնդիրը.

$$(\alpha_P - \alpha_P^0)^2 + (\alpha_R - \alpha_R^0)^2 + (\alpha_V - \alpha_V^0)^2 + (\alpha_A - \alpha_A^0)^2 + (\alpha_S - \alpha_S^0)^2 + (\alpha_N - \alpha_N^0)^2 \rightarrow \min_{\alpha_P, \alpha_R, \alpha_V, \alpha_A, \alpha_S, \alpha_N}$$

$$\left\{ \begin{array}{l} \alpha_P + \alpha_R + \alpha_V + \alpha_A + \alpha_S + \alpha_N = 1, \\ 0.2 \frac{P^* - P_{max}}{P^*} \leq \alpha_P \leq 1, \\ 0.25 \frac{R^* - R_{max}}{R^*} \leq \alpha_R \leq 1, \\ 0.25 \frac{V^* - V_{max}}{V^*} \leq \alpha_V \leq 1, \\ 0.2 \frac{A^* - A_{max}}{A^*} \leq \alpha_A \leq 1, \\ 0.07 \leq \alpha_S \leq 1, \\ 0.03 \leq \alpha_N \leq 1, \end{array} \right. \quad (2.1)$$

որտեղ P^* -ը, R^* -ը, V^* -ը, A^* -ը համակարգի մասնավոր ամպի համապատասխանաբար պրոցեսորների առավելագույն քանակն է, RAM-ի առավելագույն ծավալը, հիշողության առավելագույն ծավալը, ցանցային ադապտորների առավելագույն քանակը, P_{max} -ը, R_{max} -ը, V_{max} -ը, A_{max} -ը մասնավոր ամպի տվյալ պահին հասանելի ռեսուրսների առավելագույն արժեքներն են, α_P^0 , α_R^0 , α_V^0 , α_A^0 , α_S^0 , α_N^0 - երը կշռային գործակիցների նախընտրելի, իսկ α_P , α_R , α_V , α_A , α_S , α_N - երը՝ փնտրվող օպտիմալ արժեքներն են:

Լագրանժի ընդհանրացված եղանակով խնդիրը լուծելու համար ներկայացնենք (2.1) խնդիրը հետևյալ կերպ՝

$$f(\alpha_P, \alpha_R, \alpha_V, \alpha_A, \alpha_S, \alpha_N) = f(\cdot) = (\alpha_P - \alpha_P^0)^2 + (\alpha_R - \alpha_R^0)^2 + (\alpha_V - \alpha_V^0)^2 + (\alpha_A - \alpha_A^0)^2 + (\alpha_S - \alpha_S^0)^2 + (\alpha_N - \alpha_N^0)^2 \rightarrow \min_{\alpha_P, \alpha_R, \alpha_V, \alpha_A, \alpha_S, \alpha_N}$$

$$\left\{ \begin{array}{l} g_1(\cdot) = -\alpha_P + 0.2 \frac{P^* - P_{max}}{P^*} \leq 0, \\ g_2(\cdot) = -\alpha_R + 0.25 \frac{R^* - R_{max}}{R^*} \leq 0, \\ g_3(\cdot) = -\alpha_V + 0.25 \frac{V^* - V_{max}}{V^*} \leq 0, \\ g_4(\cdot) = -\alpha_A + 0.2 \frac{A^* - A_{max}}{A^*} \leq 0, \\ g_5(\cdot) = -\alpha_S + 0.07 \leq 0, \\ g_6(\cdot) = -\alpha_N + 0.03 \leq 0, \\ \\ g_7(\cdot) = \alpha_P - 1 \leq 0, \\ g_8(\cdot) = \alpha_R - 1 \leq 0, \\ g_9(\cdot) = \alpha_V - 1 \leq 0, \\ g_{10}(\cdot) = \alpha_A - 1 \leq 0, \\ g_{11}(\cdot) = \alpha_S - 1 \leq 0, \\ g_{12}(\cdot) = \alpha_N - 1 \leq 0, \\ \\ g_{13}(\cdot) = \alpha_P + \alpha_R + \alpha_V + \alpha_A + \alpha_S + \alpha_N - 1 = 0, \end{array} \right. \quad (2.2)$$

(2.2) – ում $g_i(\cdot) \leq 0$, $i = 1,12$ սահմանափակումները հավասարության տեսքի բերելու համար ներմուծենք $U^2 = (u_1^2, u_2^2, \dots, u_{12}^2)^T$ լրացուցիչ ոչ բացասական փոփոխականները, իսկ Լագրանժի ֆունկցիան կազմելու համար՝ $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_{13})^T$ անորոշ գործակիցները [16]:

Լագրանժի ֆունկցիան կլինի՝

$$\begin{aligned}
 L(\alpha_P, \alpha_R, \alpha_V, \alpha_A, \alpha_S, \alpha_N, U, \lambda) &= f(\cdot) - \sum_{i=1}^{12} \lambda_i (g_i(\cdot) + u_i^2) - \lambda_{13} g_{13}(\cdot) = \\
 &= (\alpha_P - \alpha_P^0)^2 + (\alpha_R - \alpha_R^0)^2 + (\alpha_V - \alpha_V^0)^2 + (\alpha_A - \alpha_A^0)^2 + (\alpha_S - \alpha_S^0)^2 + (\alpha_N - \alpha_N^0)^2 - \\
 &- \lambda_1 \left(-\alpha_P + 0.2 \frac{P^* - P_{max}}{P^*} + u_1^2 \right) - \lambda_2 \left(-\alpha_R + 0.25 \frac{R^* - R_{max}}{R^*} + u_2^2 \right) - \\
 &- \lambda_3 \left(-\alpha_V + 0.25 \frac{V^* - V_{max}}{V^*} + u_3^2 \right) - \lambda_4 \left(-\alpha_A + 0.2 \frac{A^* - A_{max}}{A^*} + u_4^2 \right) - \\
 &- \lambda_5 (-\alpha_S + 0.07 + u_5^2) - \lambda_6 (-\alpha_N + 0.03 + u_6^2) - \\
 &- \lambda_7 (\alpha_P - 1 + u_7^2) - \lambda_8 (\alpha_R - 1 + u_8^2) - \\
 &- \lambda_9 (\alpha_V - 1 + u_9^2) - \lambda_{10} (\alpha_A - 1 + u_{10}^2) - \\
 &- \lambda_{11} (\alpha_S - 1 + u_{11}^2) - \lambda_{12} (\alpha_N - 1 + u_{12}^2) - \\
 &- \lambda_{13} (\alpha_P + \alpha_R + \alpha_V + \alpha_A + \alpha_S + \alpha_N - 1):
 \end{aligned} \tag{2.3}$$

Կուն-Թակկերի պայմանները [5, 16] կներկայացվեն հետևյալ կերպ՝

$$\left. \begin{aligned}
& \lambda_i \leq 0, i = 1, \dots, 12, \\
& \left((2(\alpha_P - \alpha_P^0), 2(\alpha_R - \alpha_R^0), 2(\alpha_V - \alpha_V^0), 2(\alpha_A - \alpha_A^0), 2(\alpha_S - \alpha_S^0), 2(\alpha_N - \alpha_N^0)) - \right. \\
& \left. -(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, \lambda_8, \lambda_9, \lambda_{10}, \lambda_{11}, \lambda_{12}, \lambda_{13}) \right) \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} = 0, \\
& \lambda_1 \left(-\alpha_P + 0.2 \frac{P^* - P_{max}}{P^*} + u_1^2 \right) = 0, \\
& \lambda_2 \left(-\alpha_R + 0.25 \frac{R^* - R_{max}}{R^*} + u_2^2 \right) = 0, \\
& \lambda_3 \left(-\alpha_V + 0.25 \frac{V^* - V_{max}}{V^*} + u_3^2 \right) = 0, \\
& \lambda_4 \left(-\alpha_A + 0.2 \frac{A^* - A_{max}}{A^*} + u_4^2 \right) = 0, \\
& \lambda_5 (-\alpha_S + 0.07 + u_5^2) = 0, \\
& \lambda_6 (-\alpha_N + 0.03 + u_6^2) = 0, \\
& \lambda_7 (\alpha_P - 1 + u_7^2) = 0, \\
& \lambda_8 (\alpha_R - 1 + u_8^2) = 0, \\
& \lambda_9 (\alpha_V - 1 + u_9^2) = 0, \\
& \lambda_{10} (\alpha_A - 1 + u_{10}^2) = 0, \\
& \lambda_{11} (\alpha_S - 1 + u_{11}^2) = 0, \\
& \lambda_{12} (\alpha_N - 1 + u_{12}^2) = 0, \\
& -\alpha_P + 0.2 \frac{P^* - P_{max}}{P^*} \leq 0, \\
& -\alpha_R + 0.25 \frac{R^* - R_{max}}{R^*} \leq 0, \\
& -\alpha_V + 0.25 \frac{V^* - V_{max}}{V^*} \leq 0, \\
& -\alpha_A + 0.2 \frac{A^* - A_{max}}{A^*} \leq 0, \\
& -\alpha_S + 0.07 \leq 0, \\
& -\alpha_N + 0.03 \leq 0, \\
& \alpha_P - 1 \leq 0, \alpha_R - 1 \leq 0, \alpha_V - 1 \leq 0, \alpha_A - 1 \leq 0, \alpha_S - 1 \leq 0, \alpha_N - 1 \leq 0, \\
& \alpha_P + \alpha_R + \alpha_V + \alpha_A + \alpha_S + \alpha_N - 1 = 0:
\end{aligned} \right\}$$

Կուն-Թակկերի պայմաններով ստացված ոչ գծային հավասարումների համակարգը լուծելուց և օպտիմումի նշանակությունը որոշելուց հետո կորոշվեն կշռային գործակիցների օպտիմալ արժեքները:

Կշռային գործակիցների օպտիմալ արժեքները որոշելուց հետո ինտեգրալ գնահատականները կորոշենք հետևյալ կերպ՝

- մասնավոր ամպում տեղաբաշխելու համար՝

$$S_{if} = \alpha_P \left(1 - \frac{P}{P_{max}}\right) + \alpha_R \left(1 - \frac{R}{R_{max}}\right) + \alpha_V \frac{V}{V_{max}} + \alpha_A \left(1 - \frac{A}{A_{max}}\right) + \alpha_S \frac{S}{5} + \alpha_N \frac{N}{6}, \quad (2.4)$$

- հանրային ամպում տեղաբաշխելու համար՝

$$S_h = \alpha_P \frac{P}{P_{max}} + \alpha_R \frac{R}{R_{max}} + \alpha_V \left(1 - \frac{V}{V_{max}}\right) + \alpha_A \frac{A}{A_{max}} + \alpha_S \left(1 - \frac{S}{5}\right) + \alpha_N \left(1 - \frac{N}{6}\right) : \quad (2.5)$$

Եթե $S_{if} > S_h$, ապա նախագիծը ստեղծվում է մասնավոր ամպում, հակառակ դեպքում՝ հանրայինում:

Այսպիսով, հիբրիդ ամպում նախագծի տեղաբաշխումը որոշելու համար կատարվում է քայլերի հետևյալ հաջորդականությունը.

1. Նախագծի ստեղծման նախնական տվյալների վերլուծություն, անհամաձայնությունների բացառում;
2. Մասնավոր ամպի տվյալ պահին հասանելի ռեսուրսների առավելագույն արժեքների ստացում՝ P_{max} , R_{max} , V_{max} , A_{max} ;
3. Նախագծի ստեղծման տվյալների վերլուծություն և վճռի կայացում հետևյալ իրավիճակների համար՝
 - P_{max} , R_{max} , V_{max} , A_{max} առավելագույն արժեքներից ցանկացածի գերազանցման դեպքում նախագծի ստեղծում հանրային ամպում;
 - անվտանգության ցուցանիշի 5, ինչպես նաև նախագծի տեսակի 5 և 6 արժեքների դեպքում նախագծի ստեղծում մասնավոր ամպում;

- նախորդ 2 իրավիճակների բացառման դեպքում մասնավոր ու հանրային ամպում գործոնների գնահատականների հաշվարկ համաձայն աղյուսակ 2.1-ում ներկայացված բանաձևերի, կշռային գործակիցների օպտիմալ արժեքների որոշում (2.1) խնդրի լուծմամբ և ինտեգրալ գնահատականների հաշվարկ համաձայն (2.4) և (2.5) առնչությունների: Եթե $S_{it} > S_h$, ապա նախագիծը ստեղծվում է մասնավոր ամպում, հակառակ դեպքում՝ հանրայինում:

Ներկայացնենք կշռային գործակիցների օպտիմալ արժեքների որոշման մի քանի տարբերակներ համակարգում առկա ռեսուրսների աղյ. 2.3-ում ներկայացված սցենարների դեպքում:

Աղյուսակ 2.3

Համակարգում առկա ռեսուրսների տարբեր սցենարներ

	Սցենարներ						
	1	2	3	4	5	6	7
Պրոցեսորների քանակ (P_{max})	16	90	90	16	64	0	96
RAM-ի չափս ծավալ (R_{max})	16	16	16	2	128	0	128
Հիշողության տարածք (V_{max})	126	16	16000	16	10000	0	16000
Ցանցային ադապտորների քանակ (A_{max})	2	8	4	4	2	0	16

1-ին սցենարի դեպքում (2.1) խնդիրը $P^*=96$, $R^*=128GB$, $V^*=16TB$, $A^*=16$ արժեքների դեպքում ստանում է հետևյալ տեսքը՝

- նպատակային ֆունկցիան՝

$$\begin{aligned}
& (\alpha_P - 0.15)^2 + (\alpha_R - 0.15)^2 + (\alpha_V - 0.15)^2 + (\alpha_A - 0.1)^2 + \\
& + (\alpha_S - 0.3)^2 + (\alpha_N - 0.15)^2 \rightarrow \min_{\alpha_P, \alpha_R, \alpha_V, \alpha_A, \alpha_S, \alpha_N}
\end{aligned}$$

- սահմանափակումները՝

$$\left\{ \begin{array}{l}
\alpha_P + \alpha_R + \alpha_V + \alpha_A + \alpha_S + \alpha_N = 1, \\
\frac{1}{6} \leq \alpha_P \leq 1, \\
\frac{7}{32} \leq \alpha_R \leq 1, \\
\frac{8129}{32768} \leq \alpha_V \leq 1, \\
\frac{7}{40} \leq \alpha_A \leq 1, \\
0.07 \leq \alpha_S \leq 1, \\
0.03 \leq \alpha_N \leq 1:
\end{array} \right.$$

Խնդիրը Լագրանժի անորոշ գործակիցների ընդհանրացված եղանակով լուծելու համար կազմենք Լագրանժի ֆունկցիան՝

$$\begin{aligned}
L(\alpha_P, \alpha_R, \alpha_V, \alpha_A, \alpha_S, \alpha_N, U, \lambda) &= f(\cdot) - \sum_{i=1}^{12} \lambda_i (g_i(\cdot) + u_i^2) - \lambda_{13} g_{13}(\cdot) = \\
&= (\alpha_P - 0.15)^2 + (\alpha_R - 0.15)^2 + (\alpha_V - 0.15)^2 + (\alpha_A - 0.1)^2 + \\
&\quad + (\alpha_S - 0.3)^2 + (\alpha_N - 0.15)^2 - \\
&\quad - \lambda_1 \left(-\alpha_P + \frac{1}{6} + u_1^2 \right) - \lambda_2 \left(-\alpha_R + \frac{7}{32} + u_2^2 \right) - \\
&\quad - \lambda_3 \left(-\alpha_V + \frac{8129}{32768} + u_3^2 \right) - \lambda_4 \left(-\alpha_A + \frac{7}{40} + u_4^2 \right) - \\
&\quad - \lambda_5 (-\alpha_S + 0.07 + u_5^2) - \lambda_6 (-\alpha_N + 0.03 + u_6^2) - \lambda_7 (\alpha_P - 1 + u_7^2) - \\
&\quad - \lambda_8 (\alpha_R - 1 + u_8^2) - \lambda_9 (\alpha_V - 1 + u_9^2) - \lambda_{10} (\alpha_A - 1 + u_{10}^2) - \\
&\quad - \lambda_{11} (\alpha_S - 1 + u_{11}^2) - \lambda_{12} (\alpha_N - 1 + u_{12}^2) - \\
&\quad - \lambda_{13} (\alpha_P + \alpha_R + \alpha_V + \alpha_A + \alpha_S + \alpha_N - 1):
\end{aligned}$$

Կուն-Թակկերի պայմանները կլինեն՝

$$\left\{ \begin{array}{l}
\lambda_i \leq 0, i = 1, \dots, 12, \\
(2(\alpha_P - 0.15), 2(\alpha_R - 0.15), 2(\alpha_V - 0.15), 2(\alpha_A - 0.1), 2(\alpha_S - 0.3), 2(\alpha_N - 0.15)) - \\
-(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, \lambda_8, \lambda_9, \lambda_{10}, \lambda_{11}, \lambda_{12}, \lambda_{13}) \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} = 0, \\
\lambda_1 \left(-\alpha_P + \frac{1}{6} + u_1^2 \right) = 0, \\
\lambda_2 \left(-\alpha_R + \frac{7}{32} + u_2^2 \right) = 0, \\
\lambda_3 \left(-\alpha_V + \frac{8129}{32768} + u_3^2 \right) = 0, \\
\lambda_4 \left(-\alpha_A + \frac{7}{40} + u_4^2 \right) = 0, \\
\lambda_5 \left(-\alpha_S + 0.07 + u_5^2 \right) = 0, \\
\lambda_6 \left(-\alpha_N + 0.03 + u_6^2 \right) = 0, \\
\lambda_7 \left(\alpha_P - 1 + u_7^2 \right) = 0, \\
\lambda_8 \left(\alpha_R - 1 + u_8^2 \right) = 0, \\
\lambda_9 \left(\alpha_V - 1 + u_9^2 \right) = 0, \\
\lambda_{10} \left(\alpha_A - 1 + u_{10}^2 \right) = 0, \\
\lambda_{11} \left(\alpha_S - 1 + u_{11}^2 \right) = 0, \\
\lambda_{12} \left(\alpha_N - 1 + u_{12}^2 \right) = 0, \\
-\alpha_P + \frac{1}{6} \leq 0, \\
-\alpha_R + \frac{7}{32} \leq 0, \\
-\alpha_V + \frac{8129}{32768} \leq 0, \\
-\alpha_A + \frac{7}{40} \leq 0, \\
-\alpha_S + 0.07 \leq 0, \\
-\alpha_N + 0.03 \leq 0, \\
\alpha_P - 1 \leq 0, \alpha_R - 1 \leq 0, \alpha_V - 1 \leq 0, \alpha_A - 1 \leq 0, \alpha_S - 1 \leq 0, \alpha_N - 1 \leq 0, \\
\alpha_P + \alpha_R + \alpha_V + \alpha_A + \alpha_S + \alpha_N - 1 = 0,
\end{array} \right.$$

Ստանում ենք 31 փոփոխականներով ոչ գծային հավասարումների հետևյալ համակարգը՝

$$\left\{ \begin{array}{l}
 2(\alpha_P - 0.15) + \lambda_1 - \lambda_7 - \lambda_{13} = 0, \\
 2(\alpha_R - 0.15) + \lambda_2 - \lambda_8 - \lambda_{13} = 0, \\
 2(\alpha_V - 0.15) + \lambda_3 - \lambda_9 - \lambda_{13} = 0, \\
 2(\alpha_A - 0.1) + \lambda_4 - \lambda_{10} - \lambda_{13} = 0, \\
 2(\alpha_S - 0.3) + \lambda_5 - \lambda_{11} - \lambda_{13} = 0, \\
 2(\alpha_N - 0.15) + \lambda_6 - \lambda_{12} - \lambda_{13} = 0, \\
 \\
 \left(-\alpha_P + \frac{1}{6} + u_1^2 \right) = 0, \\
 \left(-\alpha_R + \frac{7}{32} + u_2^2 \right) = 0, \\
 \left(-\alpha_V + \frac{8129}{32768} + u_3^2 \right) = 0, \\
 \left(-\alpha_A + \frac{7}{40} + u_4^2 \right) = 0, \\
 (-\alpha_S + 0.07 + u_5^2) = 0, \\
 (-\alpha_N + 0.03 + u_6^2) = 0, \\
 (\alpha_P - 1 + u_7^2) = 0, \\
 (\alpha_R - 1 + u_8^2) = 0, \\
 (\alpha_V - 1 + u_9^2) = 0, \\
 (\alpha_A - 1 + u_{10}^2) = 0, \\
 (\alpha_S - 1 + u_{11}^2) = 0, \\
 (\alpha_N - 1 + u_{12}^2) = 0, \\
 \alpha_P + \alpha_R + \alpha_V + \alpha_A + \alpha_S + \alpha_N - 1 = 0, \\
 \lambda_1 u_1 = 0, \\
 \lambda_2 u_2 = 0, \\
 \lambda_3 u_3 = 0, \\
 \lambda_4 u_4 = 0, \\
 \lambda_5 u_5 = 0, \\
 \lambda_6 u_6 = 0, \\
 \lambda_7 u_7 = 0, \\
 \lambda_8 u_8 = 0, \\
 \lambda_9 u_9 = 0, \\
 \lambda_{10} u_{10} = 0, \\
 \lambda_{11} u_{11} = 0, \\
 \lambda_{12} u_{12} = 0:
 \end{array} \right.$$

Ոչ գծային հավասարումների համակարգը լուծվել է Նյուտոնի եղանակով:
 Լուծման արդյունքում ստանում ենք հետևյալ օպտիմալ լուծումը՝

- նպատակային ֆունկցիայի օպտիմալ արժեքը՝

$$f^*(\cdot) = 0.0553688824346382,$$

- փոփոխականների՝ կշռային գործակիցների օպտիմալ արժեքները՝

$$\alpha_P^* = 0.16666667,$$

$$\alpha_R^* = 0.21875,$$

$$\alpha_V^* = 0.24807739,$$

$$\alpha_A^* = 0.175,$$

$$\alpha_S^* = 0.14150593947426,$$

$$\alpha_N^* = 0.05:$$

- լրացուցիչ փոփոխականների, անորոշ գործակիցների օպտիմալ արժեքները՝

$$u_i^* = 0, \quad i = \overline{1,6},$$

$$u_7^* = 0.9128709,$$

$$u_8^* = 0.88388347,$$

$$u_9^* = 0.867134360,$$

$$u_{10}^* = 0.908295106,$$

$$u_{11}^* = 0.92654956,$$

$$u_{12}^* = 0.974679434:$$

$$\lambda_1^* = -0.43333333,$$

$$\lambda_2^* = -0.5375,$$

$$\lambda_3^* = -0.59615478,$$

$$\lambda_4^* = -0.55$$

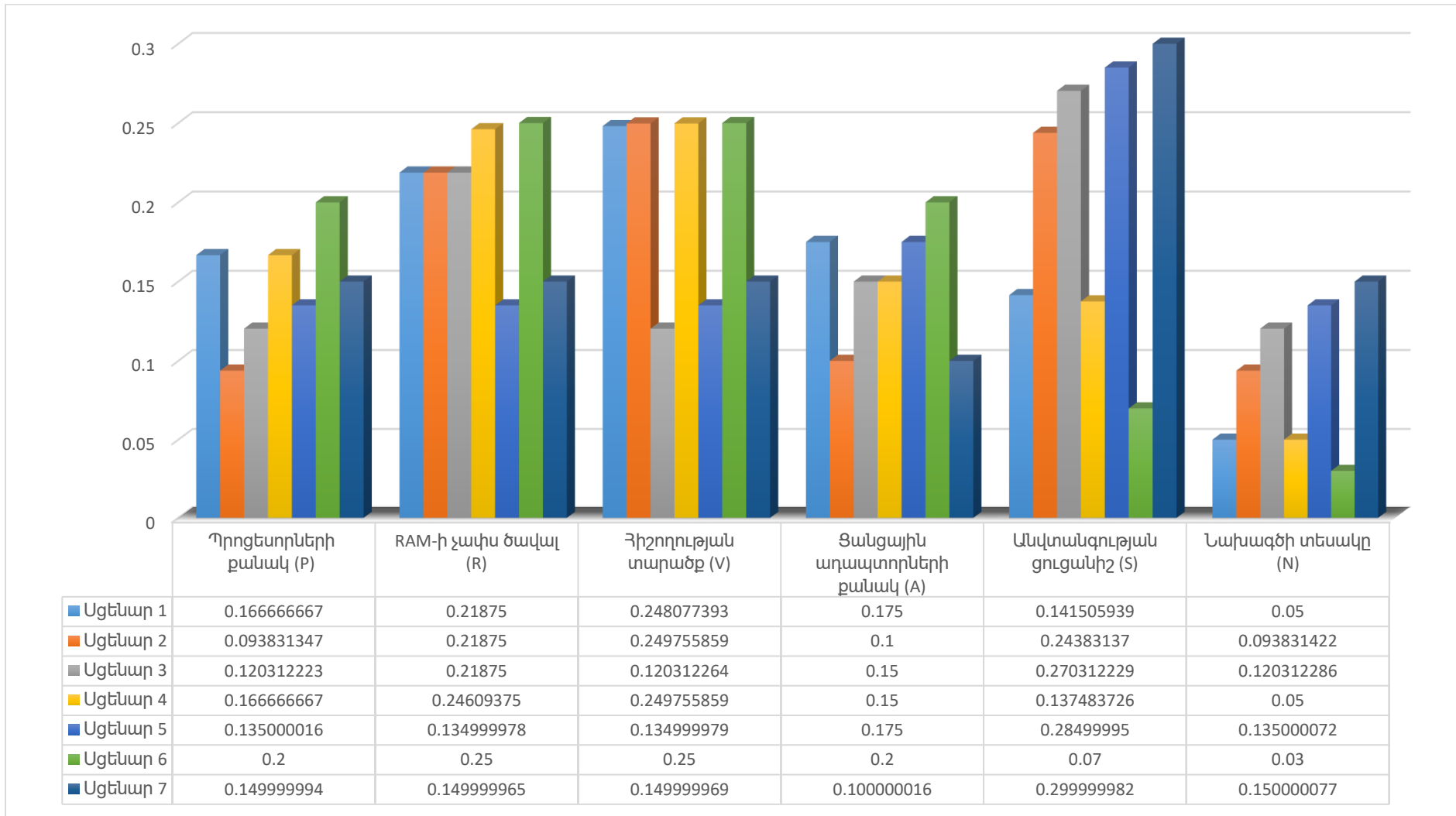
$$\lambda_5^* = -0.08301187,$$

$$\lambda_6^* = -0.2,$$

$$\lambda_i^* = 0, \quad i = \overline{7,12},$$

$$\lambda_{13}^* = -0.4:$$

Աղյ. 2.3 – ում բերված բոլոր սցենարների համար ստացված կշռային գործակիցների օպտիմալ արժեքները ներկայացված են նկ. 2.1-ում:



Նկ. 2.1 Կշռային գործակիցները տարբեր սցենարների դեպքում

Ներկայացնենք նախագծի բաշխման մի քանի տարբերակներ դիտարկված սցենարների դեպքերում (աղյուսակ 2.4):

Աղյուսակ 2.4

Նախագծերի բաշխման սցենարներ

Գործոն	Նախագծի տվյալներ	Առկա ռեսուրսներ,	Կշռային գործակիցները	Գնահատականը		Վճիռը
				մասնավոր ամպ	հանրային ամպ	
Սցենար 1, 1-ին տարբերակ						
P	4	16	0.2	0.15	0.05	$S_{\text{մ}} > S_{\text{հ}}$ մասնավոր ամպ
R	10	16	0.25	0.09375	0.15625	
V	100	126	0.25	0.1984127	0.0515873	
A	2	2	0.2	0	0.2	
S	4	{1,2,3,4,5}	0.07	0.056	0.014	
N	2	{1,2,3,4,5,6}	0.03	0.01	0.02	
				$S_{\text{մ}}=0,5081627$	$S_{\text{հ}}=0,4918373$	
Սցենար 1, 2-րդ տարբերակ						
P	12	16	0.2	0.05	0.15	$S_{\text{մ}} < S_{\text{հ}}$ հանրային ամպ
R	10	16	0.25	0.09375	0.15625	
V	64	126	0.25	0.12698413	0.12301587	
A	2	2	0.2	0	0.2	
S	4	{1,2,3,4,5}	0.07	0.056	0.014	
N	3	{1,2,3,4,5,6}	0.03	0.015	0.015	
				$S_{\text{մ}}=0,34173413$	$S_{\text{հ}}=0,6526587$	
Սցենար 2, 1-ին տարբերակ						
P	90	90	0.093831347	0	0.09383135	$S_{\text{մ}} < S_{\text{հ}}$ հանրային ամպ
R	16	16	0.21875	0	0.21875	
V	4	16	0.249755859	0.06243896	0.18731689	
A	8	8	0.1	0	0.1	
S	4	{1,2,3,4,5}	0.24383137	0.1950651	0.04876627	
N	4	{1,2,3,4,5,6}	0.093831422	0.06255428	0.03127714	
				$S_{\text{մ}}=0,32005834$	$S_{\text{հ}}=0,67994166$	

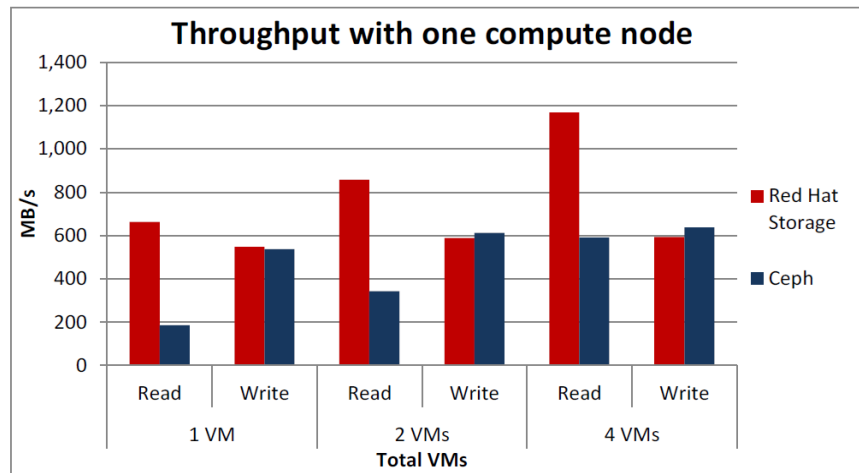
Ացենար 2, 2-րդ տարբերակ						
P	10	90	0.093831347	0.08340564	0.01042571	$S_{\bar{a}} > S_h$ մասնավոր ամս
R	4	16	0.21875	0.1640625	0.0546875	
V	4	16	0.249755859	0.06243896	0.18731689	
A	8	8	0.1	0	0.1	
S	4	{1,2,3,4,5}	0.24383137	0.1950651	0.04876627	
N	4	{1,2,3,4,5,6}	0.093831422	0.06255428	0.03127714	
				$S_{\bar{a}}=0,56752648$	$S_h=0,43247351$	
Ացենար 4, 1-ին տարբերակ						
P	10	16	0.166666667	0.0625	0.10416667	$S_{\bar{a}} < S_h$ հանրային ամս
R	2	2	0.24609375	0	0.24609375	
V	4	16	0.249755859	0.06243896	0.18731689	
A	2	4	0.15	0.075	0.075	
S	4	{1,2,3,4,5}	0.137483726	0.10998698	0.02749675	
N	4	{1,2,3,4,5,6}	0.05	0.03333333	0.01666667	
				$S_{\bar{a}}=0,34325928$	$S_h=0,65674072$	
Ացենար 5, 1-ին տարբերակ						
P	10	64	0.166666667	0.140625	0.02604167	$S_{\bar{a}} > S_h$ մասնավոր ամս
R	2	128	0.24609375	0.24224854	0.00384521	
V	4	10000	0.249755859	9.9902E-05	0.24965596	
A	2	2	0.15	0	0.15	
S	4	{1,2,3,4,5}	0.137483726	0.10998698	0.02749675	
N	4	{1,2,3,4,5,6}	0.05	0.03333333	0.01666667	
				$S_{\bar{a}}=0,52629375$	$S_h=0,47370625$	
Ացենար 7, 1-ին տարբերակ						
P	64	96	0.149999994	0.05	0.1	$S_{\bar{a}} < S_h$ հանրային ամս
R	128	128	0.149999965	0	0.14999996	
V	10000	16000	0.149999969	0.09374998	0.05624999	
A	8	16	0.100000016	0.05000001	0.05000001	
S	2	{1,2,3,4,5}	0.299999982	0.11999999	0.17999999	
N	4	{1,2,3,4,5,6}	0.150000077	0.10000005	0.05000003	
				$S_{\bar{a}}=0,41375003$	$S_h=0,58624997$	
Ացենար 7, 2-րդ տարբերակ						
P	96	96	0.149999994	0	0.14999999	$S_{\bar{a}} > S_h$ մասնավոր ամս
R	16	128	0.149999965	0.13124997	0.01875	
V	8000	16000	0.149999969	0.07499998	0.07499998	
A	16	16	0.100000016	0	0.10000002	
S	4	{1,2,3,4,5}	0.299999982	0.23999999	0.06	
N	4	{1,2,3,4,5,6}	0.150000077	0.10000005	0.05000003	
				$S_{\bar{a}}=0,54624999$	$S_h=0,45375001$	

2.2. Համակարգի տեղեկատվական ապահովումը

Ինչպես նշվել էր գրականության վերլուծության բաժնում, տվյալների հոսքերի կառավարումն առաջնային դեր ունի ամպի շահագործման ժամանակ: Հիշողությունը պետք է ոչ միայն լինի հուսալի և անվտանգ, այլ նաև ունենա առաջիկայում ընդլայնման հնարավորություն՝ առանց նախնական կառուցվածքի խաթարման: Մեծածավալ տվյալների պահեստավորման, գրանցման/ընթերցման համակարգը պետք է կարողանա իրականացնել տվյալների հետ գործողություններն արագ և առանց ընդհատումների: Աշխատանքի առաջին գլխում ուսումնասիրվել են օբյեկտային կողմնորոշմամբ տվյալների պահպանված երկու տարբեր համակարգեր և առաջադրված խնդրի տեսանկյունից անհրաժեշտ է հիմնավորել, թե այդ համակարգերից որն է ավելի նպատակահարմար կիրառել հիբրիդային ամպի շահագործման համար:

2.2.1. Ceph և GlusterFS համակարգերը

Ceph-ի և GlusterFS-ի համակարգերն ուսումնասիրելիս անհրաժեշտ է պարզել, թե երկու համակարգերից որն ունի ավելի բարձր թողունակություն: Թեստավորումների ժամանակ պարզ է դառնում, որ երկու համակարգերն էլ ունեն իրենց առավելություններն ու թերությունները և կախված առկա խնդրից պետք է որոշել թե որն է ավելի հարմար: Մեկ սերվերի հետ աշխատելիս GlusterFS-ը ցուցաբերում է ավելի բարձր թողունակություն գրել/կարդալու ժամանակ, սակայն Ceph-ի թողունակության ցուցանիշներն ավելի բարձր են, երբ համակարգերը թեստավորվում են 4-ից ավել վիրտուալ մեքենաներ շահագործելիս (նկ 2.2):



Նկ. 2.2. Համակարգերի գրանցման/ընթերցման համեմատությունը [44]

Օբյեկտային կողմնորոշմամբ տվյալների պահպանման համակարգերը հսկայական դեր ունեն ամպային տեխնոլոգիաների զարգացման բնագավառում, քանի որ ապահովում են տվյալների պահպանման ճկունություն, առանց ֆիզիկական սարքերի հետ կախվածության մեջ լինելու [2, 27]: Ուսումնասիրված երկու օբյեկտային կողմնորոշմամբ տվյալների պահպանման համակարգերն էլ ունեն իրենց առավելությունները, սակայն վերը նշված թեստավորման արդյունքներից ելնելով և հաշվի առնելով, որ Հայաստանի ազգային պոլիտեխնիկական համալսարանում կառուցվող ամպը շահագործվելու է 4-ից ավել վիրտուալ մեքենաների կիրառմամբ՝ որպես տվյալների պահպանման համակարգ ընտրվել է Ceph ծրագրային բաց կոդով օբյեկտային կողմնորոշմամբ տվյալների պահպանման համակարգը:

Այսպիսով ստեղծվող համակարգի տեղեկատվական ապահովման հիմքում կլինի համակարգը: Ի տարբերություն այլ համակարգերի, Ceph-ը կախված չէ ֆիզիկական սարքավորումներից և կարող է տեղադրվել յուրաքանչյուր տիպի կոշտ սկավառակ ունեցող համակարգչի վրա: Համակարգը տրամադրում է կենտրոնացված կառավարման սերվեր, որը կրկին բավականին հեշտացնում է դրա շահագործումը:

Ինչպես արդեն իսկ նշվել է տվյալների տեղափոխությունը՝ միգրացիան, իրականացվում է վերջինիս համար հատկացված առանձին VLAN-ով՝ համակարգի անվտանգությունն ու արագորժությունը ապահովելու համար:

Համակարգի անվտանգ և հուսալի աշխատանքի ապահովման համար մեկ այլ կարևոր հանգամանք է firewall-ի առկայությունը: Այն պաշտպանում է ցանցն արտաքին հարձակումներից, վերահսկում ցանցի աշխատանքը, ինչպես նաև ապահովում է անցանկալի ցանցային հոսքերի մուտք/ելքը համակարգի մեջ [97]: Գոյություն ունեն ինչպես ֆիզիկական, այնպես էլ ծրագրային firewall-ներ: Հաշվի առնելով, որ համակարգում շահագործվող firewall-ը պետք է լինի ճկուն և ֆինանսապես շատ ներդրումներ չպահանջի, որոշվել է օգտագործել pfSense ծրագրային firewall-ը [98]: Համակարգի ընտրության մեջ մեծ դեր են խաղացել ամպային տեխնոլոգիաների հետ մշակվող համակարգի ինտեգրման գործիքամիջոցները [99, 100]: Համակարգը բաց ծրագրային կոդով է և հիմնված է FreeBSD լինուքս դիստրիբուտիվի վրա: Բացի firewall լինելուց համակարգը տրամադրում է նաև երթուղավորման ծառայություններ, որոնք ևս կիրառվել են համակարգում տարբեր տիպի ցանցերը մեկը մյուսից տարանջատելու համար:

Ամպային համակարգի շահագործման համար անհրաժեշտ է ունենալ կենտրոնացված կառավարման համակարգ: Ամպային տեխնոլոգիաների տրամադրման համար մեծ տարածում ունեն երկու տիպի համակարգեր: Դրանք են CloudStack-ը և OpenStack-ը: Երկուսն էլ բաց ծրագրային կոդով համակարգեր են, որոնք նախատեսված են IaaS ծառայություն մատուցելու համար: Ի տարբերություն CloudStack-ի OpenStack-ին անհրաժեշտ են արտոնյալ համակարգեր տարբեր տիպի ամպային տեխնոլոգիաները կառավարելու համար: CloudStack-ը կառավարումն իրականացվում է ավելի արագ և պարզ եղանակով [101, 42, 36, 41]:

2.2.2. CloudStack համակարգը

Նախապես ստեղծվելով Cloud.com-ի կողմից [102], CloudStack-ը ձեռք է բերվել Citrix ընկերության կողմից և այնուհետև թողարկվել Apache Incubator ծրագրի շրջանակներում: Այն այժմ կառավարվում է Apache Software Foundation-ի կողմից և աջակցվում է Citrix-ի կողմից: Apache-ի կողմից ձեռք բերումից հետո համակարգը սկսեց լայն տարածում ունենալ տարբեր ընկերություններում, այսպիսով մեծացնելով իր գրադարաններն ու հնարավորությունները: Համակարգի առաջին հաստատուն տարբերակը թողարկվել է 2013թ-ին: Cloudstack-ը բաց ծրագրային կոդով համակարգ է, որը հնարավորություն է տալիս տրամադրել մասնավոր, բաց և հիբրիդային IaaS տիպի ամպային ծառայություններ (նկ. 2.3): Համակարգը կառավարում է համակարգչային, ցանցային ծառայությունները: Այն տրամադրում է API-ներ, որոնց միջոցով հնարավոր է ծրագրավորել այլ համակարգերի հետ դրա ինտեգրված աշխատանքը: Բացի իր իսկ API-ները տրամադրելուց, այն նաև ապահովում է Amazon Web Services (AWS)-ի API-ների ինտեգրում [63]: CloudStack-ի հիմնական առավելություններն են [21]՝

- **Յուրահատուկ ֆունկցիաներ** – CloudStack-ի վերջին տարբերակն ունի հզոր ֆունկցիաներ, որոնցից են տվյալների պահպանման տարբերակից անկախ աշխատանք և անվտանգության նոր մասնիկներ, որոնք թույլ են տալիս ադմինիստրատորներին ստեղծել տարբեր անվտանգության զոնաներ:
- **Հեշտ ներդրում** – CloudStack-ի տեղադրումը բավականին պարզեցված է: Սովորական տեղադրման ժամանակ, համակարգին անհրաժեշտ է միայն մեկ վիրտուալ մեքենա, իսկ այլ վիրտուալ մեքենաները համարվում են ընդհանուր ամպային համակարգի մասնիկներ:

- **Ընդլայնման հնարավորություն** – համակարգը ստեղծվել է կենտրոնացված կառավարման և անսահմանափակ ընդլայնման համար՝ այսպիսով հնարավորություն տալով կառավարել աշխարհագրական տարբեր դիրքերում գտնվող մասերից բաղկացած ամպային համակարգը:

- **Տարբեր հիպերվայզորների աջակցում (support):** CloudStack-ն աջակցում է հետևյալ հիպերվայզորներին՝

- Windows server 2012 R2 (Hyper-V Role-ը միացված)
- Hyper-V 2012 R2
- CentOS 6.2 KVM-ի հետ
- Red Hat enterprise Linux 6.2 KVM-ի հետ
- XenServer 6.0.2 (Hotfix-ով)
- XenServer 5.0, 5.1 և 5.5

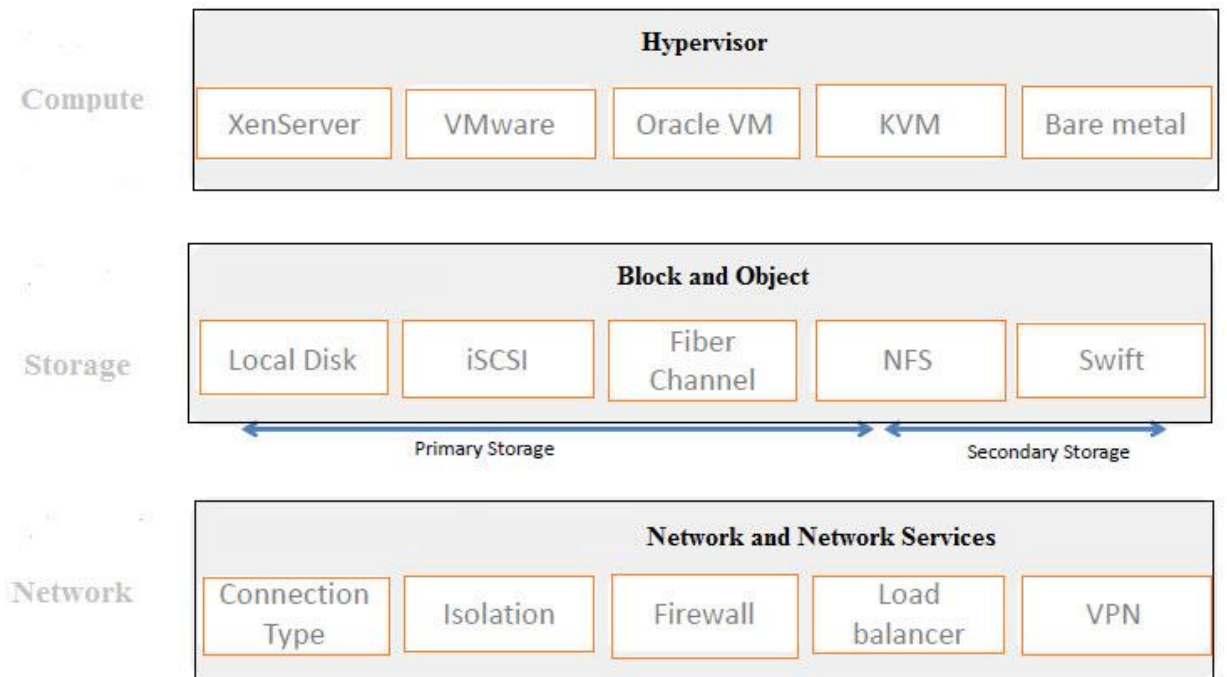
- **Մանրամասն գրականություն**

- **Ինտերակտիվ վեբ ինտերֆեյս**

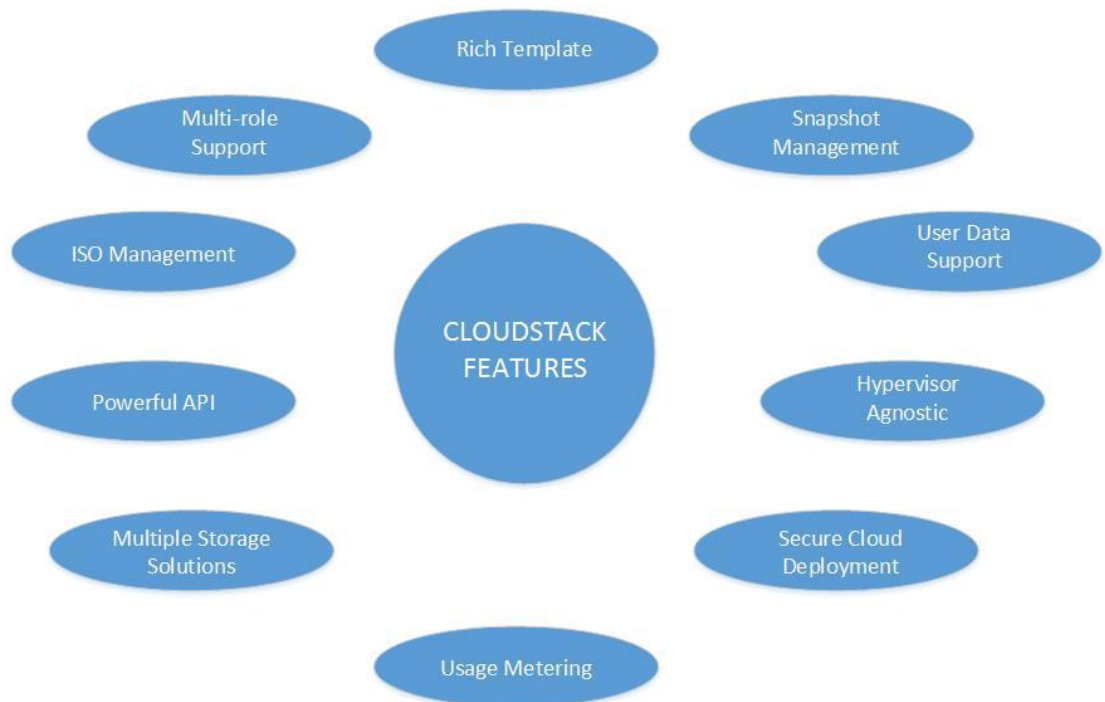
Cloudstack-ը կարող է շահագործվել մի շարք խնդիրների լուծման համար, ինչպիսիք են՝

- նորաստեղծ համակարգերի թեստավորումը,
- վեբ ինֆրաստրուկտուրայի տրամադրումը,
- վիրտուալացման ծառայությունների տրամադրումը (նկ. 2.4):

Վերոհիշյալ համակարգերի շահագործման համար բնականաբար անհրաժեշտ է ֆիզիկական սերվեր, որի վիրտուալացման շնորհիվ հնարավոր է դառնում վերոհիշյալ համակարգերի տրամադրումը մեկ ֆիզիկական սերվերի միջոցով:



Նկ. 2.3. Cloudstack-ի տրամադրած ծառայությունները



Նկ. 2.4. Cloudstack-ի ֆունկցիաները

Ինչպես ներկայացրել ենք 1-ին գլխում, վիրտուալացումը հիմնովին հեղափոխեց այս խնդիրների լուծման ուղիները: Ապահովելով ֆիզիկական սերվերների օպտիմալացում՝ այն հնարավորություն է տալիս մեկից ավել օպերացիոն համակարգերի շահագործում մեկ ֆիզիկական սարքի վրա: Վիրտուալացման միջոցով սերվերի վրա ստեղծվում են վիրտուալ մեքենաներ, որոնք աշխատում են իրարից առանձնացված միջավայրերում:

Ավանդական լուծումների ժամանակ սերվերի վրա տեղադրվում էր մեկ օպերացիոն համակարգ, որն էլ շահագործվում էր: Եթե օգտագործվող սերվերի հիշողությունն ամբողջությամբ օգտագործված էր լինում, ապա այն պետք է փոխարինվեր նոր սերվերով կամ դրան կցվեր նորը [65]:

Ավանդական լուծումով սերվերներն ունեն հետևյալ առավելությունները՝

- համակարգերի հեշտ տեղադրում,
- վերականգնման պարզ տարբերակ:

Թերություններն են՝

- ֆիզիկական մակարդակի ղեկավարման ֆինանսական ոչ շահավետություն,
- դուալիկացիայի խնդիրներ,
- ֆիզիկական մակարդակի թարմացման դժվարություն:

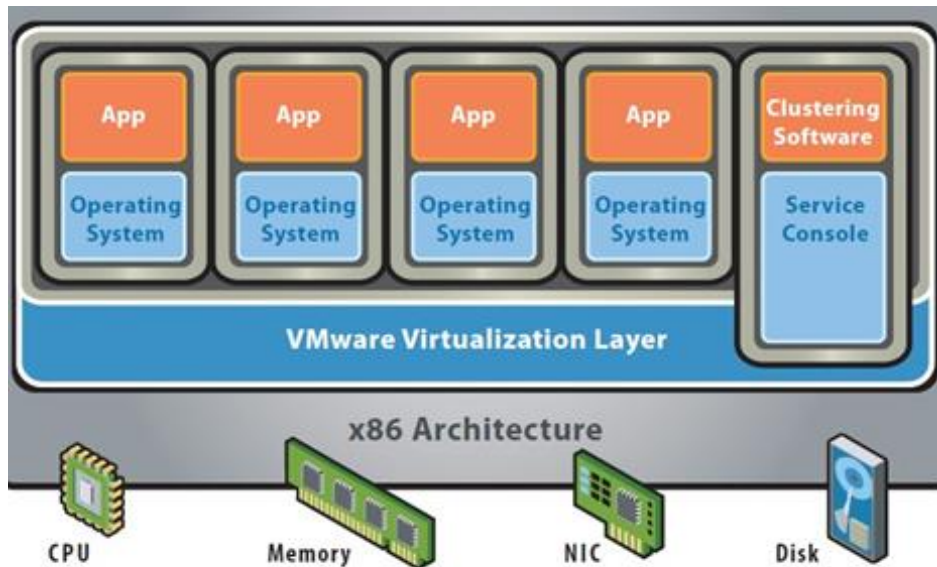
Այս խնդիրները մեծամասամբ լուծվում են վիրտուալացման կիրառման ժամանակ: Վիրտուալացման համակարգերն իրականացնում են ծրագրային մակարդակի ինկապսուլյացիա՝ բաժանելով այն ֆիզիկական մակարդակից: Այս լուծումը հնարավորություն է տալիս առաջիկայում, անհրաժեշտության պարագայում, տեղափոխել տեղադրված օպերացիոն համակարգը (տրամադրվող ծառայության հետ) մեկ այլ ֆիզիկական սերվերի վրա:

Տեխնոլոգիայի մեկ այլ առավելությունն այն է [39, 104], որ եթե շահագործման ընթացքում տրամադրված ռեսուրսները բավարար չլինեն, ապա առանց որևէ ապարատային կառուցվածքի փոփոխության, հնարավոր կլինի կատարել լրացուցիչ ռեսուրսների հավելում արդեն առկա վիճակին: Բացի այդ, եթե հաշվի առնենք, որ տարածքը հնարավոր է տրամադրել նաև այնպես, որ օգտատերն ինքնուրույն կարգավորումներ անի, ապա այս պարագայում ևս պարզ է դառնում, որ լուծումը միայն վիրտուալացման կիրառումն է: Այս տեխնոլոգիան տրամադրում են մի շարք ընկերություններ՝ բաց և փակ կոդով տարբերակներով [105]: Բաց ծրագրային կոդով լուծումները հնարավորություն են տալիս փոքրացնել կատարվող ծախսերը: Ընդհանուր առմամբ վիրտուալացման կիրառումը տալիս է հետևյալ առավելությունները՝

- կիրառվող ֆիզիկական ռեսուրսների ՕԳԳ-ի բարձրացում,
- գնի նվազում՝ ելնելով ֆիզիկական սարքերի քանակի փոքրացումից,
- սպասարկման ծախսերի փոքրացում,
- վերականգնման ծախսերի փոքրացում,
- ադմինիստրավորման փոքրացում,
- ծրագրերի տեղադրման ժամանակի կրճատում,
- համատեղելիության խնդիրների կրճատում:

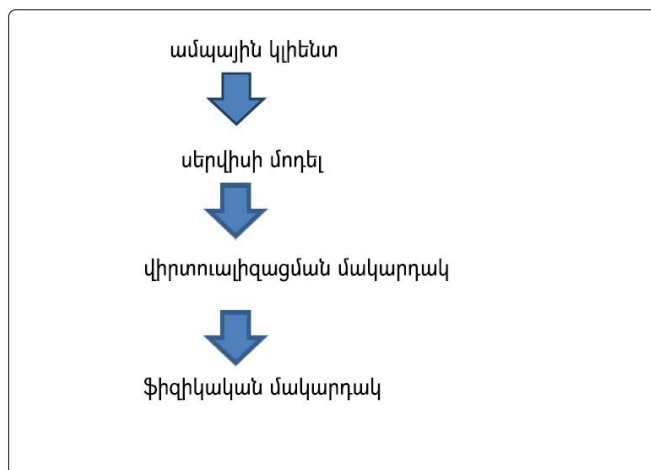
Որպես վիրտուալացման տեխնոլոգիա ուսումնասիրենք VMware տեխնոլոգիան [106, 53, 46]՝ հաշվի առնելով դրա լայն ֆունկցիոնալությունն ու համատեղելիության բարձր ցուցանիշները: Վիրտուալացման համակարգը տեղադրվում է սերվերի վրա, որպես դրա հիմնական օպերացիոն համակարգ: Առաջիկայում տեղադրվող օպերացիոն համակարգերը (ՕՀ) արդեն իսկ տեղադրված հիմնական օպերացիոն համակարգի «հյուր» ՕՀ-եր են (նկ. 2.5): Համակարգը հնարավորություն է տալիս «հյուր» օպերացիոն համակարգերի տեղափոխություն (migration), կլոնավորում

(cloning), առաջիկայում ռեսուրսների ավելացում, հիշողության տարածքի մեծացում և այլն [107]: Համակարգի առավելություններից է նաև CloudStack ամպային համակարգի հետ ինտեգրված լինելը [108]:



Նկ. 2.5. Վիրտուալացման տեխնոլոգիայի կառուցվածքը [106]

Ամպային տեխնոլոգիաներում տարածք/հիշողությունը վիրտուալ կերպով տրամադրվում է օգտատերերին, որն էլ պահանջում է կենտրոնական սերվեր (պլատֆորմ), որտեղ հիպերվայզորը (ծրագիր, որն ուղիղ աշխատում է ֆիզիկական սարքերի հետ) աշխատում է (նկ.2.6):



Նկ. 2.6. Հիմնական կառուցվածքը

Վիրտուալացման մոդելը կազմված է ամպային ծառայությունից օգտվող օգտատերերից, ծառայության մոդելից, վիրտուալացման մոդելից և ամբողջ ծառայությունը մատուցող ծրագրից, որն անմիջապես աշխատում է ֆիզիկական մակարդակի հետ: Վիրտուալացման տեխնոլոգիան հնարավորություն է տալիս մեկ սերվերի վրա շահագործել մեկից ավելի օպերացիոն համակարգեր՝ այդպիսով բարձրացնելով շահագործվող սերվերի արդյունավետությունը: Բացի այն, որ մեկ ֆիզիկական սերվերի վրա շահագործվում են մի քանի օպերացիոն համակարգեր, դրանք նաև օգտագործում են սերվերի հզորությունը միայն այն ժամանակ, երբ դրա կարիքն ունենում են: Այսինքն՝ եթե մեկ օպերացիոն համակարգը նշված պահին ավելի շատ պրոցեսորային թողունակության կարիք ունի, ապա վիրտուալացման համակարգն այդ պահին O/S-ին տրամադրում է իր կողմից պահանջված թողունակության ծավալը: Այս առավելություններից բացի, ամպային տեխնոլոգիաներում վիրտուալացման առավելություններին ավելանում են նաև հետևյալ առավելությունները՝

- չկա ֆիզիկական սերվերներ ձեռք բերելու անհրաժեշտություն,
- ռեսուրսները վարձակալվում են ըստ կարիքների,
- ինքնուրույն վիրտուալ մեքենաների ստեղծման հնարավորություն:

Ելնելով վիրտուալացման առավելություններից և դրա շահագործման մեծ ծավալներից, այժմ ապարատային արտադրողներն ապահովում են վիրտուալացման շահագործումն իրենց կողմից արտադրված ֆիզիկական սարքավորումների վրա: Այսպես, «Ինթել» ընկերությունը առաջինն էր, որ ներկայացրեց Intel® VT տեխնոլոգիան: Intel® VT -ի աշխատանքային սկզբունքը պրոցեսորի վիրտուալացման ապահովումն է [104]: VMware ընկերությունը նույնպես ունի հատուկ լուծումներ տարբեր ընկերությունների կողմից արտադրված սերվերների համար՝ յուրաքանչյուրի համար իր իսկ դրայվերների ինտեգրումով:

Վիրտուալացումն ապահովում է մի շարք գործողությունների ավտոմատացում [104], որոնցից են՝

- վիրտուալ մեքենաների տրամադրում,
- վիրտուալ մեքենաների ռեսուրսների վերաբաշխում,
- համակարգերի աշխատանքի մասին տեղեկատվության տրամադրում:

2.3. Եզրակացություններ 2-րդ գլխի վերաբերյալ

1. ՀԱՊՀ-ում հիբրիդային ամպի մատուցման համար ընտրվել է Amazon Web Services հանրային ամպը՝ դրա առաջին տարին անվճար լինելու, մեծ ֆունկցիոնալ ունենալու և Cloudstack-ի հետ ինտեգրման համար նախատեսված API-ների առկայության համար: Ընտրությունը հիմնավորվել է ինչպես ֆինանսական շահավետությամբ, այնպես էլ բազմաֆունկցիոնալ հնարավորությունների առկայությամբ:
2. Համեմատվել են երկու օբյեկտային տվյալների պահպանման համակարգերը՝ Ceph և GlusterFS: Կատարված ուսումնասիրությունների հիման վրա պարզվել է, որ ստեղծվող ավտոմատացման համակարգում շահագործելու համար նպատակարար է կիրառել Ceph-ը:
3. Ամպային ծառայությունների մատուցման համար նախատեսված երկու խոշոր լուծումներից՝ CloudStack և OpenStack, ընտրվել է առաջինը: Ընտրության հիմքում ընկել են CloudStack-ի ինչպես բարձր ֆունկցիոնալությունն ու համեմատաբար պարզ շահագործման հնարավորությունները, այնպես էլ AWS-ի հետ ինտեգրման համար արդեն իսկ առկա գործիքամիջոցները:

4. Վերոհիշյալ համակարգերի շահագործումն ապահովվելու է վիրտուալացման հիման վրա՝ ռեսուրսների օպտիմալ շահագործման համար: Վիրտուալացումն ապահովվում է VMware ընկերության կողմից տրամադրվող ESXi համակարգի միջոցով:
5. Համակարգում պատվիրված նախագծերի օպտիմալ տեղաբաշխումն ապահովելու նպատակով կիրառվել է վճիռների կայացման կշռային գործակիցների եղանակը, ընդ որում կշռային գործակիցների օպտիմալ արժեքների որոշման համար ձևակերպվել են ոչ գծային ծրագրավորման խնդիրներ: Մշակվել է ալգորիթմ նախագծի պատվիրված տվյալների և համակարգում առկա ռեսուրսների միջոցով կշռային գործակիցների ստացման, ինտեգրալ գնահատականների հաշվման և վճռի կայացման համար:

ԳԼՈՒԽ 3. ՏԵՂԵԿԱՏՎԱԿԱՆ ԾԱՌԱՅՈՒԹՅՈՒՆՆԵՐԻ ՏՐԱՄԱԴՐՄԱՆ ԱՎՏՈՄԱՏԱՑՎԱԾ ՀԱՄԱԿԱՐԳԻ ՏԵԽՆԻԿԱԿԱՆ ԱՊԱՀՈՎՈՒՄԸ

3.1. Ավտոմատացված համակարգի ամպային ենթակառուցվածքն ու դրա տեղադրումը

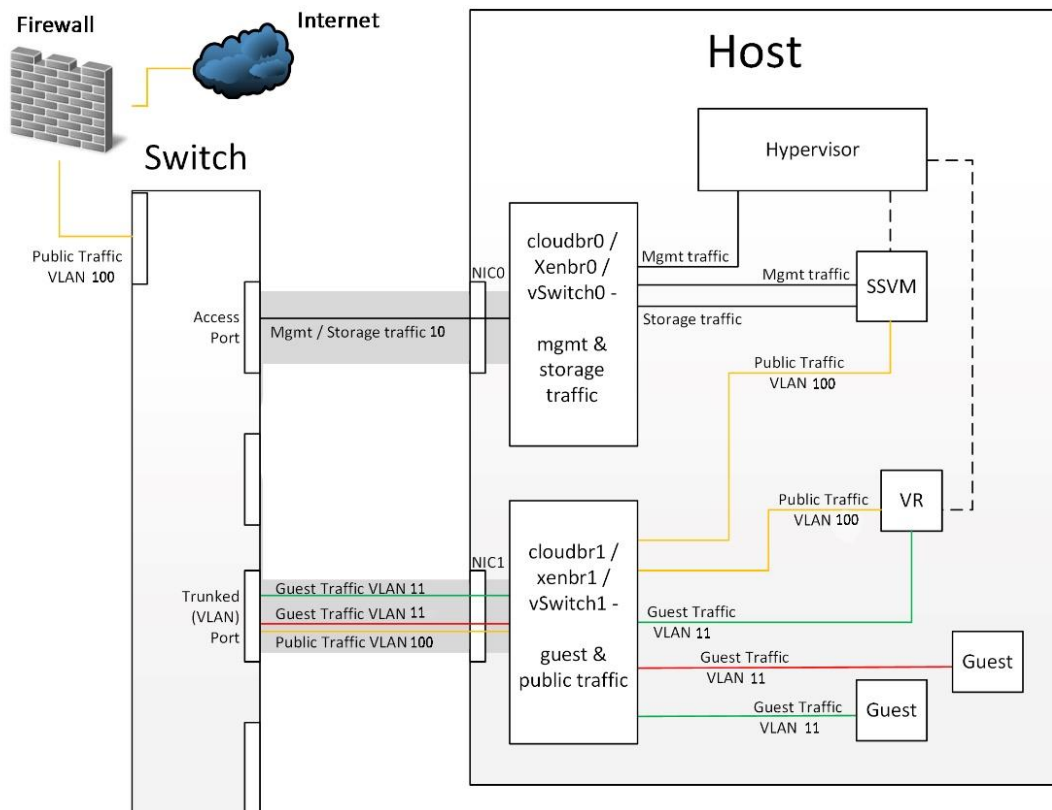
Յուրաքանչյուր ամպային համակարգ շահագործման հանձնելուց առաջ, մի քանի գործոններ անհրաժեշտ է հաշվի առնել, մասնավորապես՝

- անվտանգությունը,
- ընդլայնման հնարավորությունը,
- աշխատունակությունը,
- հուսալիությունը,
- կառավարումը:

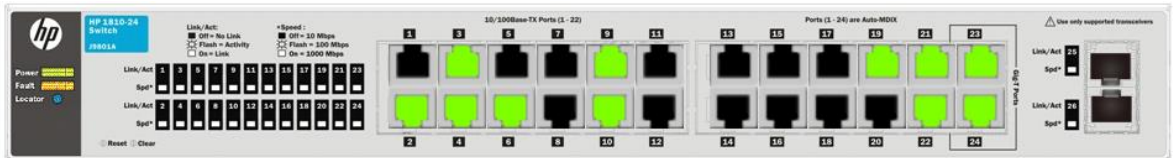
Նշված գործոնները հաշվի առնելով՝ մի շարք սերվերներ են շահագործվել ավտոմատացված համակարգը ստեղծելու և անհրաժեշտ ծառայությունները մատուցելու համար: Առաջնահերթ, համալսարանի ընդհանուր լոկալ ցանցից ամպի ցանցը բաժանելու համար մի քանի VLAN-ներ են ստեղծվել, որոնք բաժանում են ցանցի կառավարման, հիշողության և համացանցի տրամադրման հոսքերը ինչպես ընդհանուր ցանցից, այնպես էլ մեկը մյուսից: Ընդհանուր առմամբ ստեղծվել են չորս առանձին VLAN-ներ՝ կառավարման, հիշողության, լոկալ ցանցի և համացանցի համար: VLAN-ներով բաժանված ցանցի օրինակը ներկայացված է նորքոհիշյալ (նկ. 3.1):

Յուրաքանչյուր տիպի ցանցը պետք է բաժանված լինի մյուսից ինչպես անվտանգության ապահովման համար, այնպես էլ այն համգամանքով պայմանավորված, որ որոշակի սերվերներ և ցանցեր ուղղակի կարիք չունեն հասանելի լինել ամբողջ

համակարգին: Դրանք պետք է կապ հաստատեն միայն միմյանց միջև: Վերոհիշյալը վերաբերում է տվյալների օբյեկտային պահպանման համակարգի նոդերի և դրանց կառավարման սերվերին: Նոդերը կարիք չունեն հասանելի լինել որևէ համակարգի, բացի իրենց կառավարման սերվերից: Առաջիկայում կտեսնենք, որ տվյալների պահպանման համակարգի նոդերը կապնվում են իրենց կառավարման սերվերին և առաջիկա կապն ամբողջ համակարգի հետ ապահովում է միայն կառավարման սերվերը: Հետևաբար տվյալների օբյեկտային պահպանման համակարգի կառավարման սերվերը պետք է ունենա երկու տիպի ցանցեր, որոնք կգտնվեն տարբեր VLAN-ների մեջ՝ մեկը նոդերի հետ կապ հաստատելու համար, իսկ մյուսը ամբողջ ամպային համակարգի հետ հաղորդակցվելու համար: Վերոհիշյալ խնդիրը լուծելու համար կիրառվել է խելացի բաժանարար (նկ 3.2):



Նկ. 3.1. Ցանցի ճարտարապետությունը



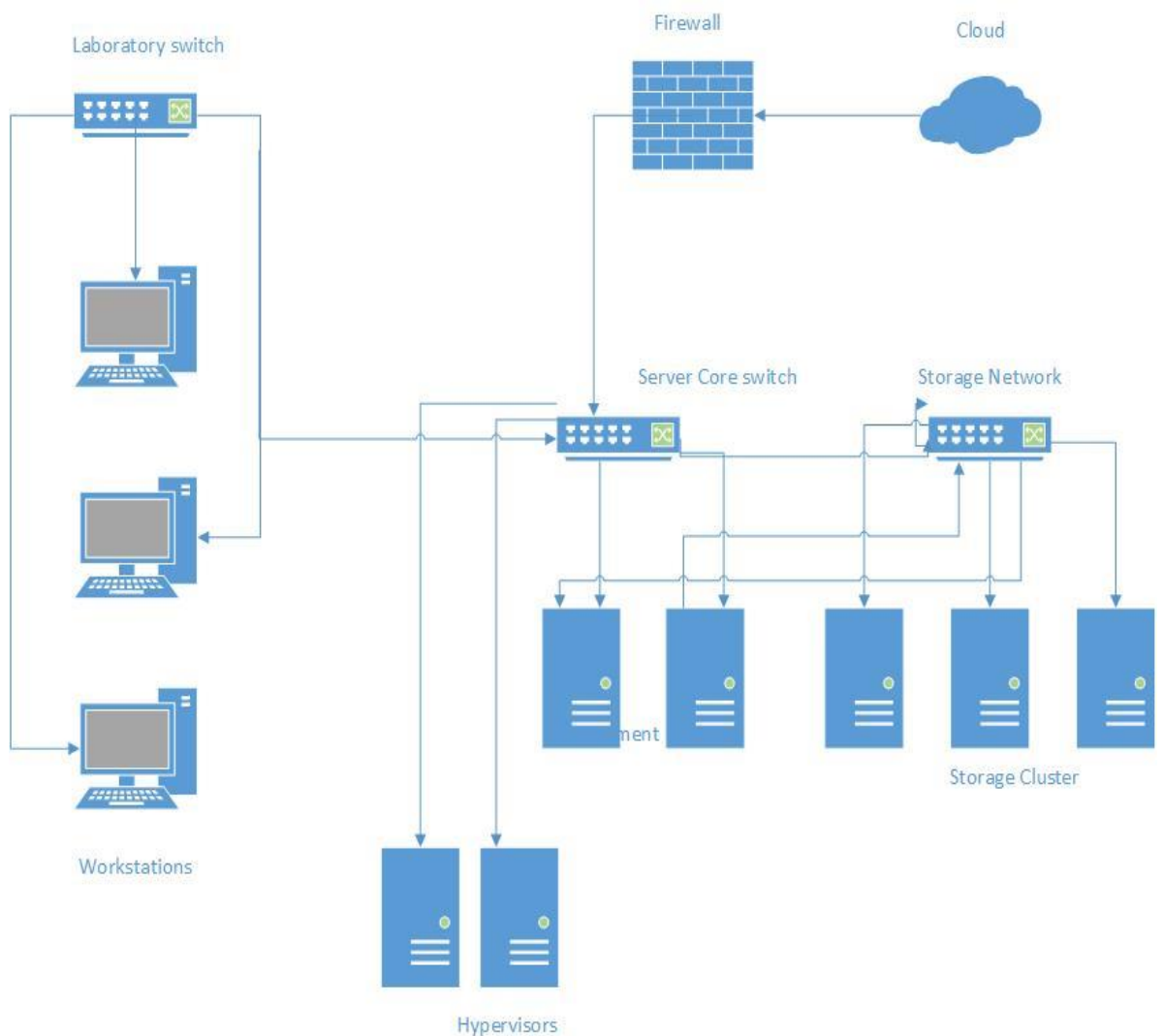
Նկ. 3.2. Խելացի բաժանարարը

Հիմնական սերվերը, որը գտնվում է սերվերային հանգույցում, կառավարում է ամպի շահագործման համար ստեղծված բոլոր համակարգերը, որոնցից են ցանցային հիշողու-թյուններն իրենց օպերացիոն համակարգերով, հիպերվայզորներն իրենց նոդերի հետ: Ամպային ենթակառուցվածքը գտնվում է տվյալների կենտրոնում, իսկ ամպի կլիենտները գտնվում են 5-րդ մասնաշենքում, որոնց հետ կապն իրականացվում է պաշտպանված ցանցով (նկ 3.3): Այս ճարտարապետությունը հնարավորություն է տալիս ունենալ արագագործ, ճկուն համակարգ, որից կարող են օգտվել ինչպես ուսանողները, այնպես էլ համալսարանի աշխատակիցները, գիտաշխատողները:

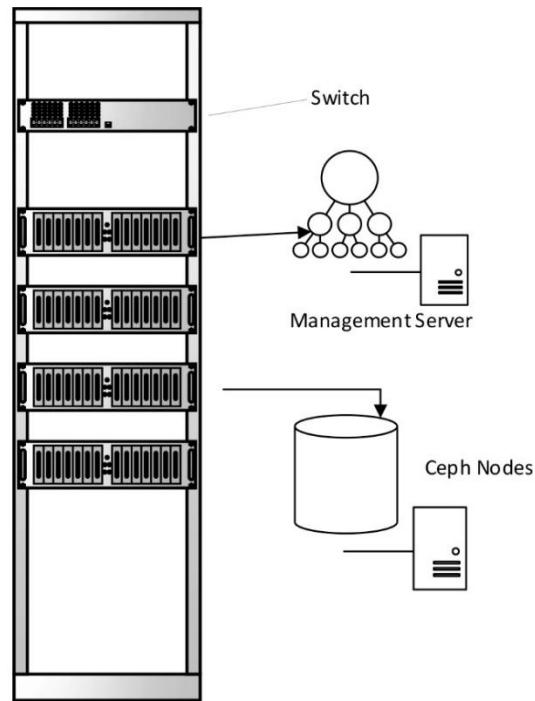
Շահագործվող սերվերային պահարանն ունի նկ. 3.4ա-ում ներկայացված տեսքը, իսկ սերվերների պարամետրերը ներկայացված են նկ. 3. 4բ-ում:

Հայաստանի ազգային պոլիտեխնիկական համալսարանում (ՀԱՊՀ) ստեղծվող ամպային լուծումների համար կարևորագույն դեր ունի վիրտուալացումը, որի վրա կառուցվում են բոլոր մյուս համակարգերը: Կիրառելով VMware ընկերության կողմից առաջարկվող լուծումները՝ հնարավոր է դառնում մեկ ֆիզիկական սերվերի վիրտուալացման շնորհիվ աշխատացնել 5 վիրտուալ կառավարման համակարգեր, որոնց դերը ՀԱՊՀ ամպային լուծումների անխափան և անվտանգ աշխատանքի ապահովումն է: Բացի այդ, ներդրված ամպային ծառայությունները մատուցող համակարգը՝ Apache Cloudstack-ը նույնպես աշխատում է վիրտուալացման հիմունքների վրա: Միևնույն

Ժամանակ, առկա խնդիրներին համապատասխան ծառայությունը մատուցող ծրագրային լուծումն ընտրելիս ու ներդնելիս, պետք է հաշվի առնել համակարգի հետագա ընդլայնման ու անվտանգության խնդիրները, վերականգնման գործիքամիջոցների առկայությունը:



Նկ. 3.3. Անպային ենթակառուցվածքը



ա.

General	
Manufacturer:	Dell Inc.
Model:	PowerEdge R520
CPU Cores:	12 CPUs x 2.199 GHz
Processor Type:	Intel(R) Xeon(R) CPU E5-2420 v2 @ 2.20GHz
License:	VMware vSphere 5 Enterprise Plus - Licensed for 2 physic...
Processor Sockets:	2
Cores per Socket:	6
Logical Processors:	24
Hyperthreading:	Active
Number of NICs:	4
State:	Connected
Virtual Machines and Templates:	5
vMotion Enabled:	N/A
VMware EVC Mode:	Disabled
vSphere HA State	ⓘ N/A
Host Configured for FT:	N/A
Active Tasks:	
Host Profile:	N/A
Image Profile:	Dell-ESXi-5.5U2-2068190-A00
Profile Compliance:	ⓘ N/A
DirectPath I/O:	Supported

բ.

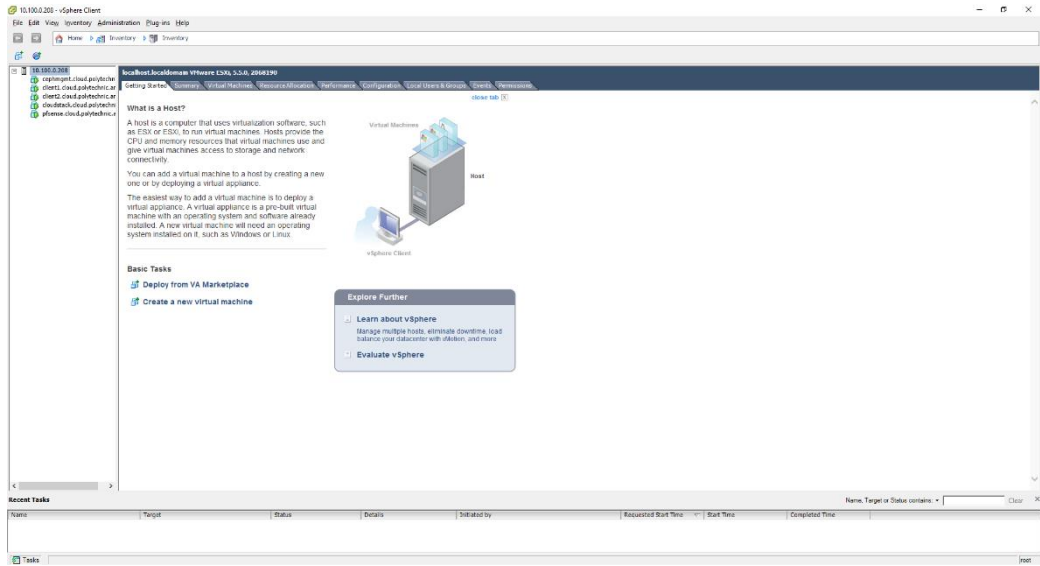
Նկ. 3.4. Սերվերային պահարանի կառուցվածքը (ա) և սերվերների պարամետրերը (բ):

3.1.1. VMWare ESXi 6 համակարգի ներդրումը ֆիզիկական սերվերի վրա

ESXi 6 համակարգը տեղադրելու համար նախ անհրաժեշտ է կարգավորել ֆիզիկական սերվերը՝ կազմակերպել RAID-ի տիպը, BIOS-ից ցանցային մուտքի արգելափակում կամ տրամադրում և այլն: Վերոհիշյալ առաջնային խնդիրները լուծելուց հետո ESXi 5.5-ը կարելի է տեղադրել և կատարել անհրաժեշտ կարգավորումները: Համակարգից օգտվելու և վիրտուալ մեքենաներ ստեղծելու համար հատկացված է կլիենտ, որը տեղադրվում է օգտատիրոջ համակարգչի վրա (նկ. 3.6): Կլիենտի միջոցով համակարգ մուտք գործելուց հետո, այն ունի նկ.3.7-ում ցուցադրված տեսքը:

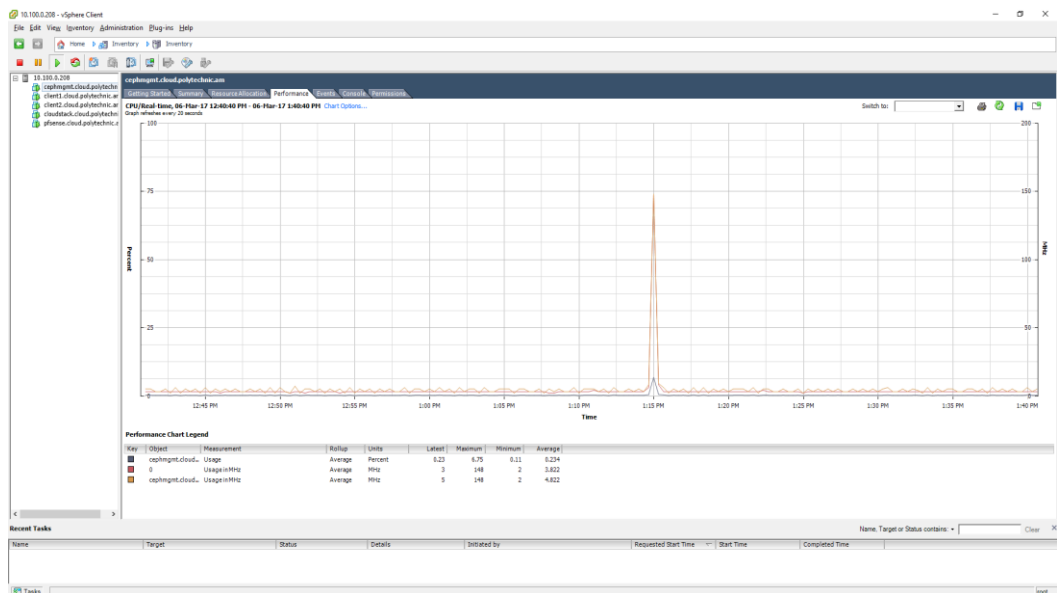


Նկ. 3.6 VMware vSphere Client-ի տեսքը



Նկ. 3.7. WMwarre համակարգի գլխավոր էջը

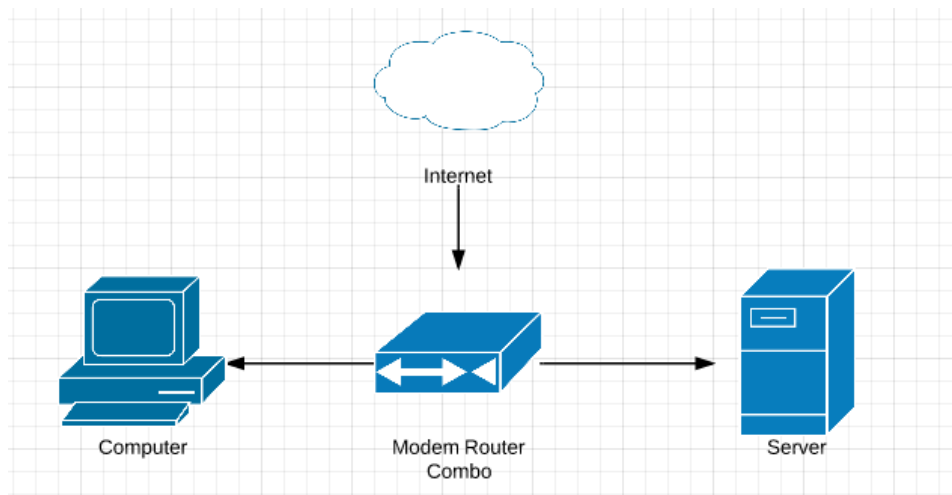
Ինչպես երևում է նկ.3.7-ում, համակարգում արդեն իսկ ստեղծված են բոլոր անհրաժեշտ վիրտուալ մեքենաները, որոնց միջոցով ապահովվելու է ամպային համակարգի աշխատանքը: Համակարգը տրամադրում է մանրամասն տեղեկատվություն յուրաքանչյուր վիրտուալ մեքենայի աշխատանքի և ծանրաբեռնվածության մասին: Նկ. 3.8-ում բերված է Ceph համակարգի կառավարման սերվերի աշխատանքային ծանրաբեռնվածության մասին տեղեկատվություն, որը տրամադրում է ESXi համակարգը:



Նկ. 3.8. Ceph վիրտուալ մեքենայի ռեսուրսների ծանրաբեռնվածության էջը

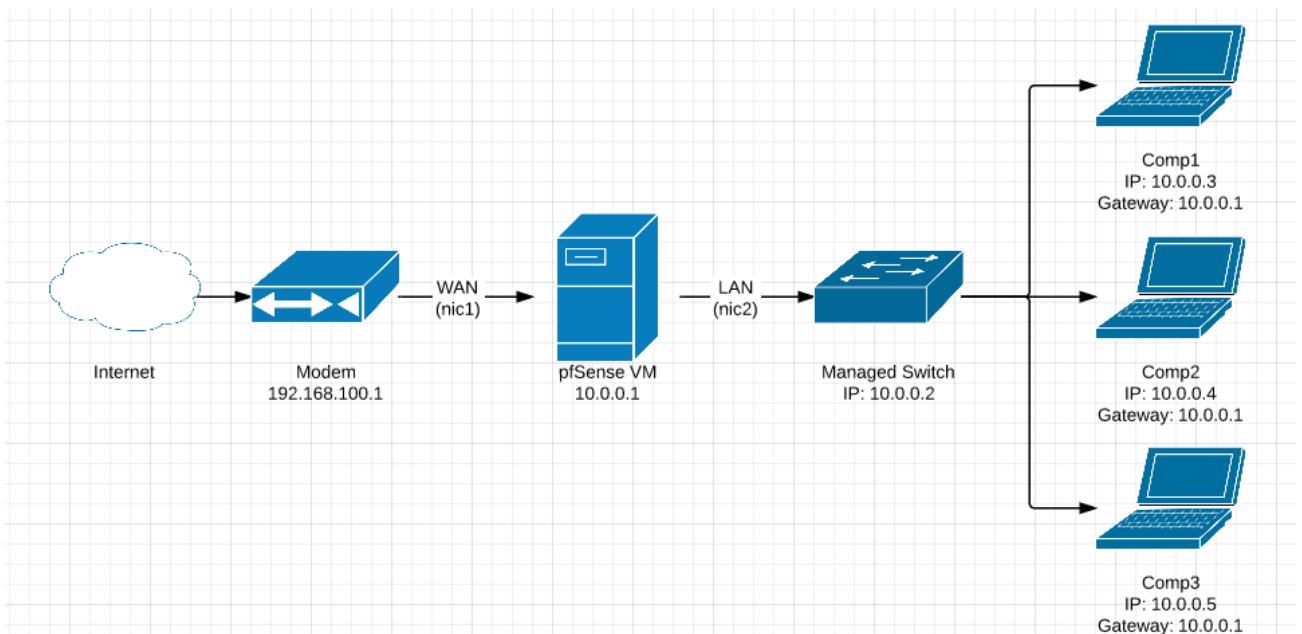
3.1.2. pfSense համակարգի տեղադրումը vmWare-ի վրա

Ինչպես արդեն իսկ նշվել է ամպային համակարգի անվտանգությունն ու անհրաժեշտ երթուղավորումները իրականացնելու համար շահագործվել է pfSense բաց ծրագրային կողով firewall-ը: Մինչ firewall-ի շահագործումը ցանցն ունի հետևյալ տեսքը (նկ. 3.9)՝



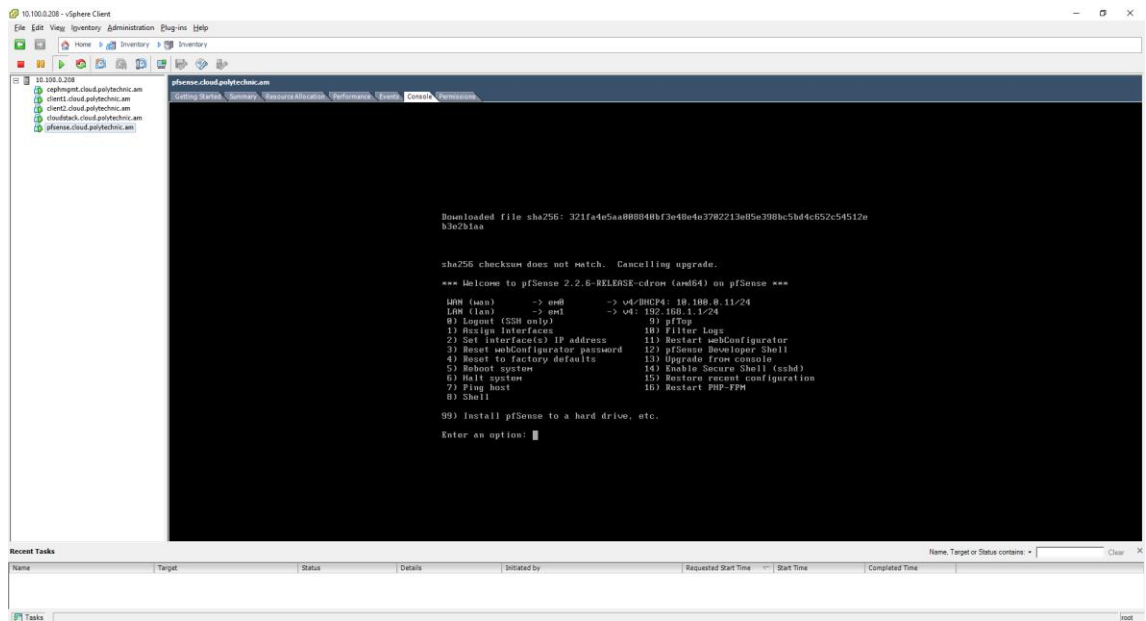
Նկ.3.9. Ցանցի կառուցվածքն առանց

Firewall-ի շահագործումից հետո ցանցը կունենա նկ. 3.10-ում բերված տեսքը:



Նկ. 3.10 Ցանցի կառուցվածքը firewall-ով

pfSense-ը տեղադրելուց և անհրաժեշտ ցանցային կարգավորումներից հետո այն կունենա հետևյալ տեսքը (նկ.3.11).



Նկ.3.11 pfSense համակարգ

3.1.3. Ceph համակարգի տեղադրումը

Տվյալների պահպանման համար կիրառվող Ceph բաց կոդով ծրագրային համակարգի տեղադրման համար իրականացվում են հետևյալ քայլերը՝

- նոր վիրտուալ մեքենայի ստեղծում, որը կառավարելու է ceph նոդերը,
- վիրտուալ մեքենայի վրա օպերացիոն համակարգի տեղադրում,
- օպերացիոն համակարգի վրա ceph cluster-ի ստեղծում,
- Ceph նոդերի վրա օպերացիոն համակարգի տեղադրում,
- կառավարման սերվերի և նոդերի միջև կապի ստեղծում:

Ceph կառավարման սերվերի համար կիրառվում է Ubuntu 14.04 TLS օպերացիոն համակարգը, որը տեղադրելուց հետո պետք է կատարել ցանցային կարգավորումներ: Կարգավորումներն իրականացնելուց հետո, դրանք կարելի ստուգել կիրառելով ifconfig հրամանը, որի արդյունքում կունենանք (նկ. 3.12)՝

```
levon@cephmgmt:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:1d:69:84
          inet addr:10.100.100.100  Bcast:10.100.100.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe1d:6984/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1918705 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1133 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:139650156 (139.6 MB)  TX bytes:68604 (68.6 KB)

eth1      Link encap:Ethernet  HWaddr 00:0c:29:1d:69:8e
          inet addr:10.100.0.52  Bcast:10.100.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe1d:698e/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7060721 errors:0 dropped:720 overruns:0 frame:0
          TX packets:8455 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:522093903 (522.0 MB)  TX bytes:724624 (724.6 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1936 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1936 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:195064 (195.0 KB)  TX bytes:195064 (195.0 KB)

levon@cephmgmt:~$
```

Նկ. 3.12 Ceph սերվերի ցանցային ադապտորների ցանկ

Ինչպես երևում է նկ.3.12-ում օպերացիոն համակարգն ունի երկու տիպի ցանցեր, որոնցից առաջինը նախատեսված է նոդերի հետ կապ հաստատելու համար, իսկ երկրորդը նախատեսված է այլ համակարգերի հետ կապ հաստատելու համար: Անհրաժեշտ է ստուգել մի քանի պայման մինչ Ceph համակարգի տեղադրումը: Անհրաժեշտ է տեղադրել SSH սերվեր, որի համար պետք է կիրառել հետևյալ հրամանը՝

```
sudo apt-get install openssh-server
```

Այնուհետև պետք է ստեղծել նոդերի օգտանունները՝

```
ssh user@ceph-server, sudo useradd -d /home/ceph -m ceph, sudo  
passwd ceph
```

Անհրաժեշտ թույլտվությունները տրամադրելու համար հարկավոր է կիրառել հետևյալ հրամանը՝

```
echo "ceph ALL = (root) NOPASSWD:ALL" | sudo tee  
/etc/sudoers.d/ceph  
sudo chmod 0440 /etc/sudoers.d/ceph
```

Հրամանն ապահովում է կառավարման սերվերի վրա ստեղծված օգտատիրոջն առանց անվտանգության գաղտնաբառի մուտքագրման փոփոխություններ իրականացնել համակարգի նոդերի վրա:

```
ssh-keygen  
Generating public/private key pair.  
Enter file in which to save the key (/ceph-  
client/.ssh/id_rsa):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /ceph-  
client/.ssh/id_rsa.  
Your public key has been saved in /ceph-  
client/.ssh/id_rsa.pub.
```

Այնուհետև կարող ենք տեղադրել Ceph համակարգը հիմնական կառավարման սերվերի վրա՝ կիրառելով `ceph-deploy install` հրամանը: Հրամանում նախընտրելի տարբերակը չնշելու դեպքում կտեղադրվի ամենաթարմը: Սակայն, եթե կա

ցանկություն որևէ թողարկում տեղադրելու հրամանն աշխատացնելուց առաջ պետք է նշել ցանկալի տարբերակի անունը: Մեր պարագայում տեղադրվում է Firefly թողարկումը՝ քանի որ, համաձայն վերլուծության, այն ամենաթարմ և կայուն աշխատող թողարկումն է՝

```
levon@cephmngmnt: ~  
levon@cephmngmnt:~$ ceph-deploy install --release firefly {ceph-node1 ceph-node2  
ceph-node3}
```

Հրամանն աշխատացնելուց հետո տեղադրվում է ceph համակարգը երեք նոդերի վրա, որոնց անուններն են՝ ceph-node1, ceph-node2, ceph-node3:

Բոլոր նոդերի վրա ceph-ը տեղադրելուց հետո անհրաժեշտ է ստեղծել կլաստեր՝

```
levon@cephmngmnt: ~  
levon@cephmngmnt:~$ ceph-deploy --cluster cscluster mon create ceph-node{1,2,3}
```

Ինչպես երևում է, հրամանը ստեղծում է նոր cscluster անվանմամբ կլաստեր: Կլաստերը ստեղծելուց հետո կարելի է արդեն ստեղծել մոնիտորներ նոդերի վրա, որն իրականացվում է հետևյալ կերպ՝

```
levon@cephmngmnt: ~  
levon@cephmngmnt:~$ ceph-deploy mon create ceph-node{1,2,3}
```

Մինչ նոդերը կարող են աշխատացնել ceph OSD-ները անվտանգությունից ելնելով պետք է իրականացնել բանալիների փոխանակում՝

```
levon@cephmngmnt: ~  
levon@cephmngmnt:~$ ceph-deploy --cluster cscluster gatherkeys ceph-node{1,2,3}
```

Ceph-ի OSD-ները շահագործելու համար անհրաժեշտ է թվարկել նոդերի վրա առկա կոշտ սկավառակները՝

```
levon@cephmngmnt: ~  
levon@cephmngmnt:~$ ceph-deploy disk list ceph-node{1,2,3}
```

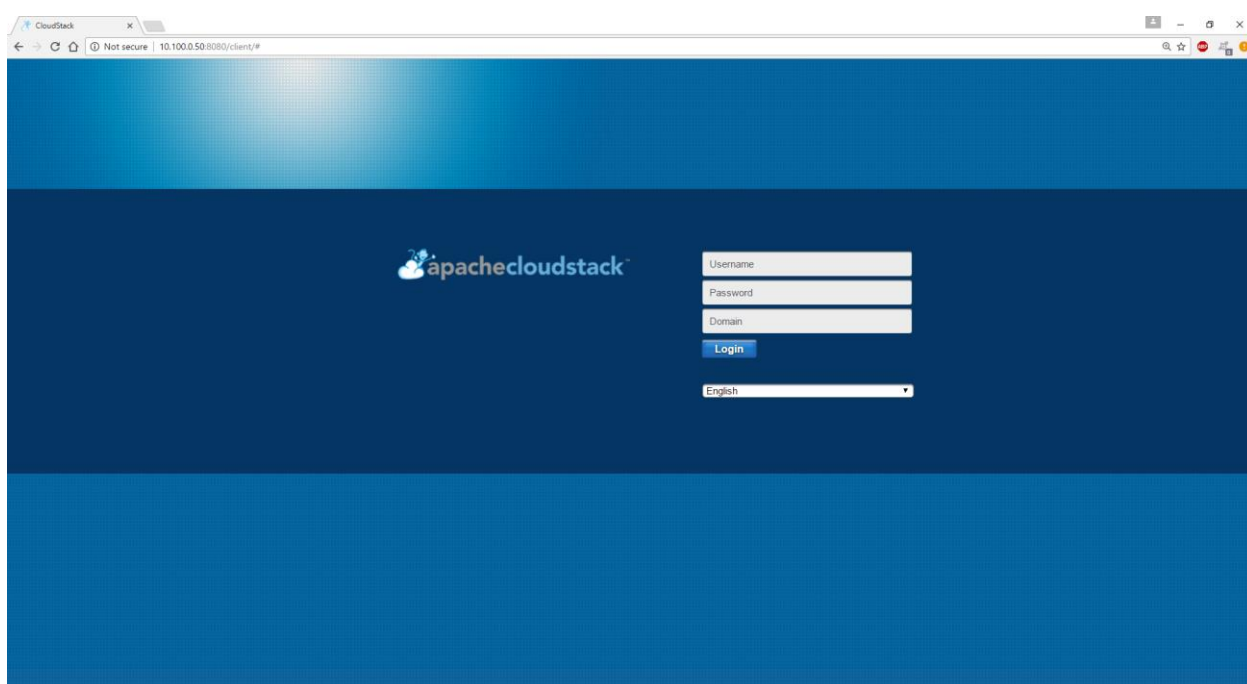
Վերոհիշյալ կարգավորումները կատարելուց հետո օբյեկտային կառավարմամբ տվյալների պահպանման համակարգը պատրաստ է շահագործման համար:

Եթե մինչ այս տեղեկատվական հոսքը ցանցում պետք է կազմակերպվեր բաց, ապա SSL համակարգը տեղադրելուց և կարգավորումներն իրականացնելուց հետո, ինֆորմացիան կընթանա փակ վիճակում և միայն սերվերն ու օգտատերը կկարողանան ընթերցել այն: Նախապես տեղադրված Apache վեբ սերվերի մեջ արդեն իսկ առկա է OpenSSL մոդուլը, որն օգտագործելու համար նաև պետք է ստեղծել գաղտնի բանալին և հավաստագիրը սերվերի համար: Այս երկուսի շնորհիվ սերվերի և օգտատիրոջ միջև փոխանակվելու է գաղտնագրման ալգորիթմը, որի շնորհիվ ինֆորմացիան փակվել և բացվելու է երկու կողմերի միջև:

3.1.4. Cloudstack համակարգի շահագործումը

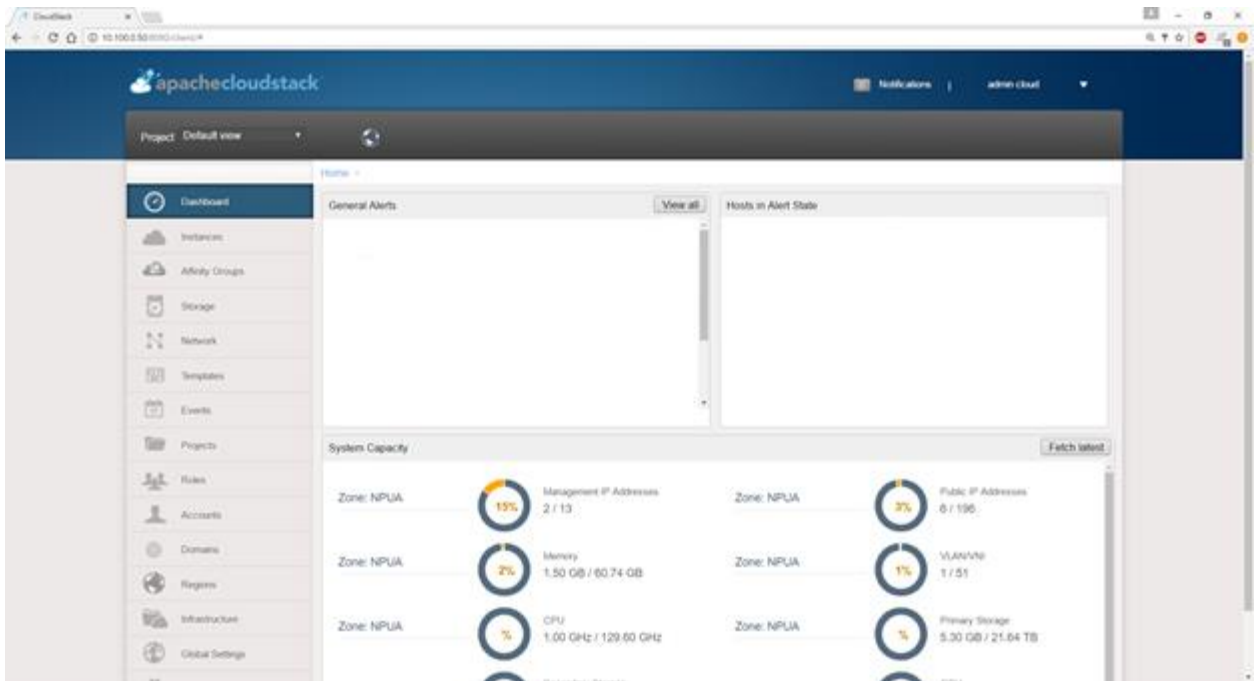
Cloudstack համակարգի տեղադրումից հետո անհրաժեշտ է կատարել բոլոր կարգավորումները, որոնց մեջ ներառվում են ֆիզիկական սերվերների կցումը, նոր զոնայի ստեղծումը, օգտատերերի կարգավորումը և այլն:

Համակարգի մուտքային էջն ունի հետևյալ տեսքը (նկ. 3.14)՝



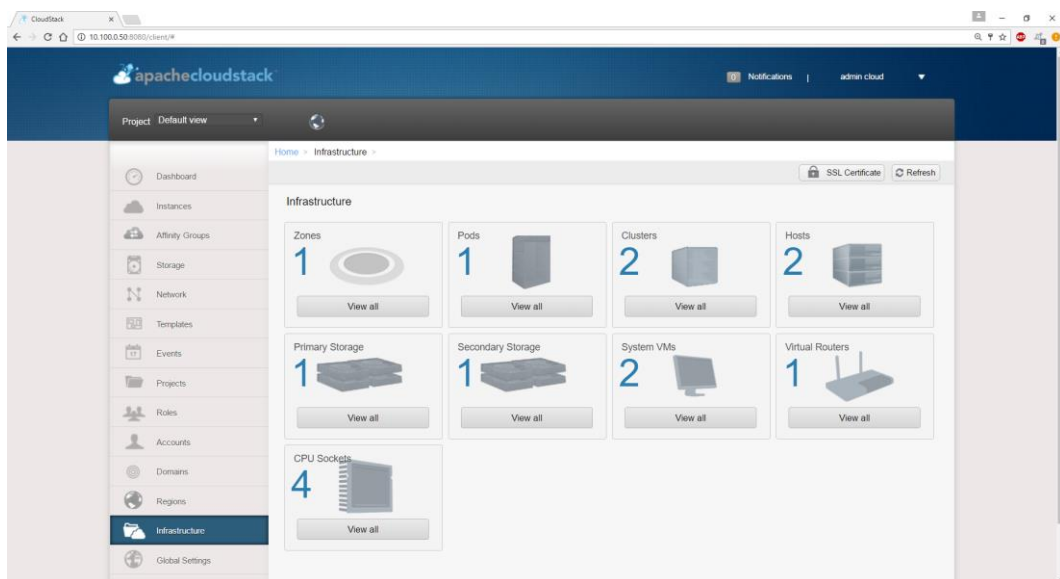
Նկ. 3.14. Համակարգի մուտքային էջը

Օգտատիրոջ անունն ու գաղտնաբառը մուտքագրելուց հետո մուտք ենք գործում համակարգի գլխավոր էջ, որտեղ արտապատկերված է համակարգի ընթացիկ կարգավիճակները (նկ. 3.15)՝



Նկ. 3.15 Cloudstack համակարգի գլխավոր էջ

Համակարգի կառուցվածքը փոփոխելու համար անհրաժեշտ է մուտք գործել “Infrastructure” բաժինը, որտեղից հնարավոր է ավելացնել նոր ֆիզիկական սերվերներ, որոնք հանդես են գալիս որպես հոստեր: Վերոհիշյալ բաժինն ունի հետևյալ տեսքը (նկ.3.16)՝

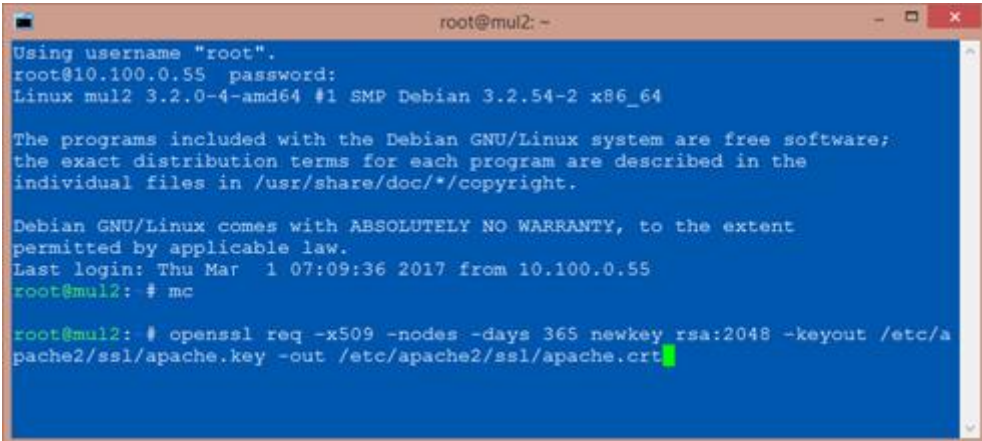


Նկ. 3.16 Cloudstack համակարգի “Infrastructure” բաժինը

3.1.5 Գաղտնի բանալու և հավաստագրի ստեղծումը վեբ սերվերի համար

Գաղտնագրման և վերծանման համար կիրառվում է OpenSSL բաց կոդով ծրագիրը, որը գաղտնագրումը կատարում է SSL և TLS արձանագրությունների հիման վրա: Ծրագրի հիմնական հատվածը գրված է C ծրագրավորման լեզվով, որի հիման վրա կիրառվում են կրիպտոգրաֆիայի ֆունկցիաները: Առաջին քայլով պետք է ստեղծել սերվերի հավաստագիրը, որը կհանդիսանա հիմք, որի համար կստեղծվի բանալին օգտատերի կիրառելու համար:

Հավաստագիրը ստեղծելուց հետո անհրաժեշտ է ստեղծել բանալին, որպեսզի երբ օգտատերը հարցում կատարի կայքին սերվերը կարողանա տրամադրել ծածկագրման բանալին ինֆորմացիայի գաղտնագրման համար (նկ.3.17)՝



```
root@mul2: ~
Using username "root".
root@10.100.0.55 password:
Linux mul2 3.2.0-4-amd64 #1 SMP Debian 3.2.54-2 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

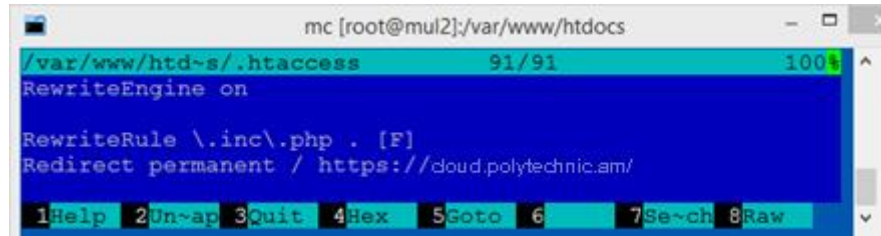
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Mar  1 07:09:36 2017 from 10.100.0.55
root@mul2: # mc

root@mul2: # openssl req -x509 -nodes -days 365 newkey rsa:2048 -keyout /etc/a
pache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt
```

Նկ. 3.17 Սերտիֆիկատի ստեղծումը կայքի համար

Բանալին ստեղծելու ժամանակ սերվերը պահանջելու է տրամադրել տվյալներ կազմակերպության անվանման, գտնվելու մասին, որոնք տրամադրելուց հետո բանալին և հավաստագիրը կստեղծվեն: Սակայն դեռ կայքին հարցում կատարելու ժամանակ սերվերից պատասխան է ստացվելու կայքի բաց տեսքը առանց https

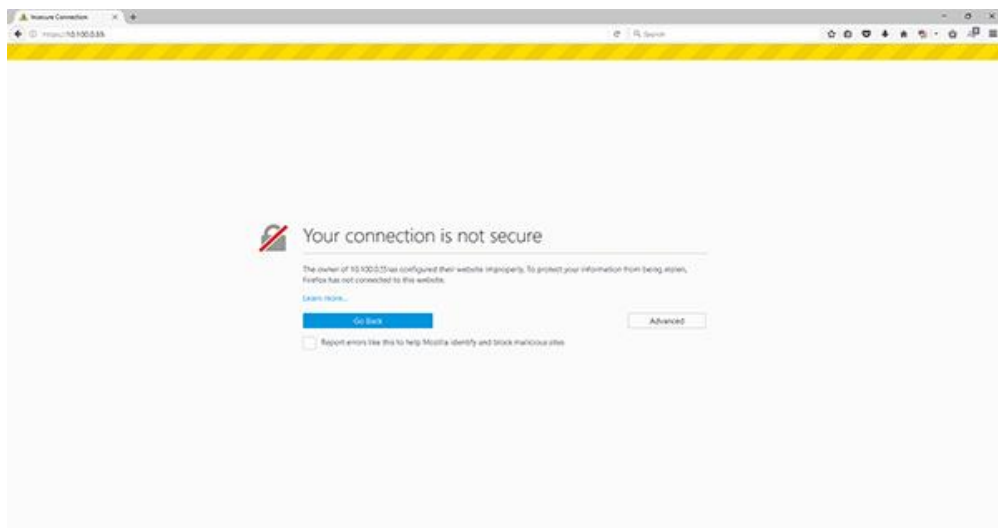
արձանագրության: Սրանից խուսափելու համար և բոլոր հարցումները https արձանագրությունով անցկացնելու համար, կայքի htaccess կարգավորող ֆայլի մեջ պետք է կատարել հետևյալ փոփոխությունները՝ բոլոր հարցումներն անվտանգ տեղափոխելու համար (նկ.3.18)՝



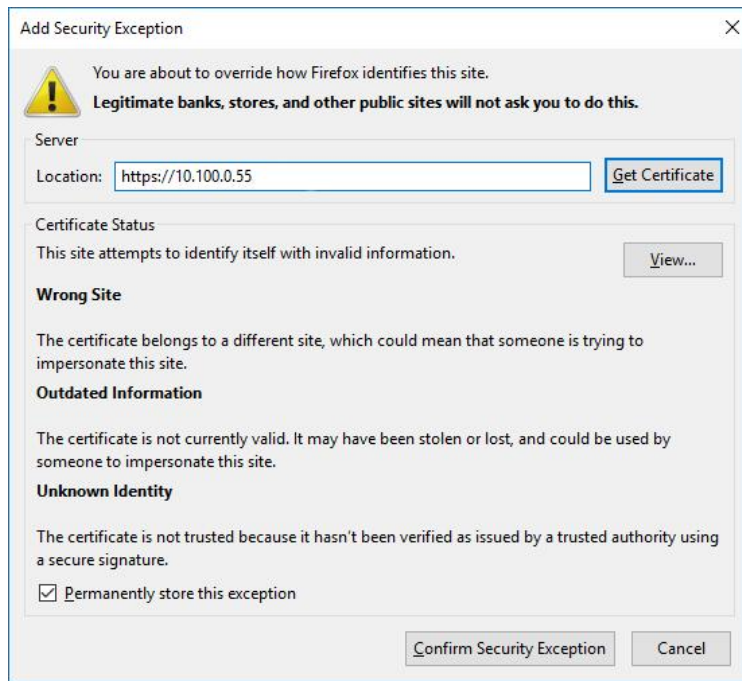
```
mc [root@mul2]:/var/www/htdocs
/var/www/htd~s/.htaccess 91/91 100%
RewriteEngine on
RewriteRule \.inc\.php . [F]
Redirect permanent / https://cloud.polytechnic.am/
1Help 2Un~ap 3Quit 4Hex 5Goto 6 7Se~ch 8Raw
```

Նկ. 3.18 Կայքի դիմման վերահասցեավորում

Կատարված փոփոխությունները պահպանելուց և սերվերը ևս մեկ անգամ վերբեռնելուց հետո կայքին դիմելու ժամանակ բոլոր հարցումները կանցնեն https պաշտպանված արձանագրության միջով: Այսուհետ կայքին դիմելու դեպքում սերվերից օգտատերը դրա կողմից ստանալու է գաղտնագրման բանալին կապի հաստատման համար: Սակայն քանի որ մեր ստեղծած բանալին գրանցած չէ բրաուզերի մեջ, ապա այն կտա անվտանգության զգուշացում անձանոթ բանալու մասին, որը պետք է անտեսել, քանի որ մենք գիտենք թե ում կողմից է այդ բանալին տրամադրված (նկ. 3.19, 3.20):

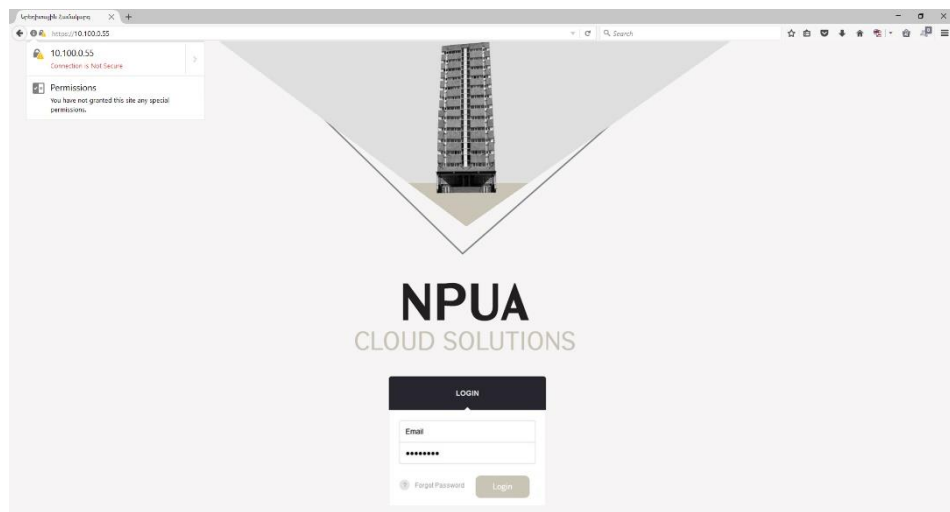


Նկ. 3.19 Սերտիֆիկատի հաստատում բրաուզերի համար



Նկ. 3.20 Հաստատումը կայքի համար

Կայքի համար բացառությունը պահպանելուց հետո, կարող ենք մուտք գործել համակարգ և տեսնել հետևյալ էջը (նկ. 3.21)՝



Նկ. 3.21. Կայքի ավարտված տեսքը

3.2. Եզրակացություններ 3-րդ գլխի վերաբերյալ

1. Հայաստանի ազգային պոլիտեխնիկական համալսարանում ստեղծված հիբրիդային ամպի տարբեր տիպերի ցանցերը տարանջատվել են VLAN-ների միջոցով՝ ցանցի անվտանգությունն ու արագագործությունն ապահովելու նպատակով:
2. Ֆիզիկական սերվերի վրա տեղադրվել է ESXi 6.0 վիրտուալացում ապահովող օպերացիոն համակարգը, որի վրա արդեն տեղադրվել են ներքոհիշյալ համակարգերը որպես վիրտուալ մեքենաներ:
3. Համակարգի անվտանգությունն ապահովելու համար շահագործվել է նաև pfSense ծրագրային firewall-ը, որն ապահովում է նաև երթուղավորում ցանցի մեջ:
4. Տեղադրվել է Ceph օբյեկտային կառավարմամբ տվյալների պահպանման համակարգը՝ իր կլիենտների հետ ներառյալ (կլիենտները տեղադրվում են առանձին ֆիզիկական սերվերներ վրա):
5. Տեղակայվել է Cloudstack համակարգը, որն ապահովում է հիբրիդային ամպի ծառայության մատուցումը:

ԳԼՈՒԽ 4. ՏԵՂԵԿԱՏՎԱԿԱՆ ԾԱՌԱՅՈՒԹՅՈՒՆՆԵՐԻ ՏՐԱՄԱԴՐՄԱՆ ԱՎՏՈՄԱՏԱՑՎԱԾ ՀԱՄԱԿԱՐԳԻ ԾՐԱԳՐԱՅԻՆ ԱՊԱՀՈՎՈՒՄԸ

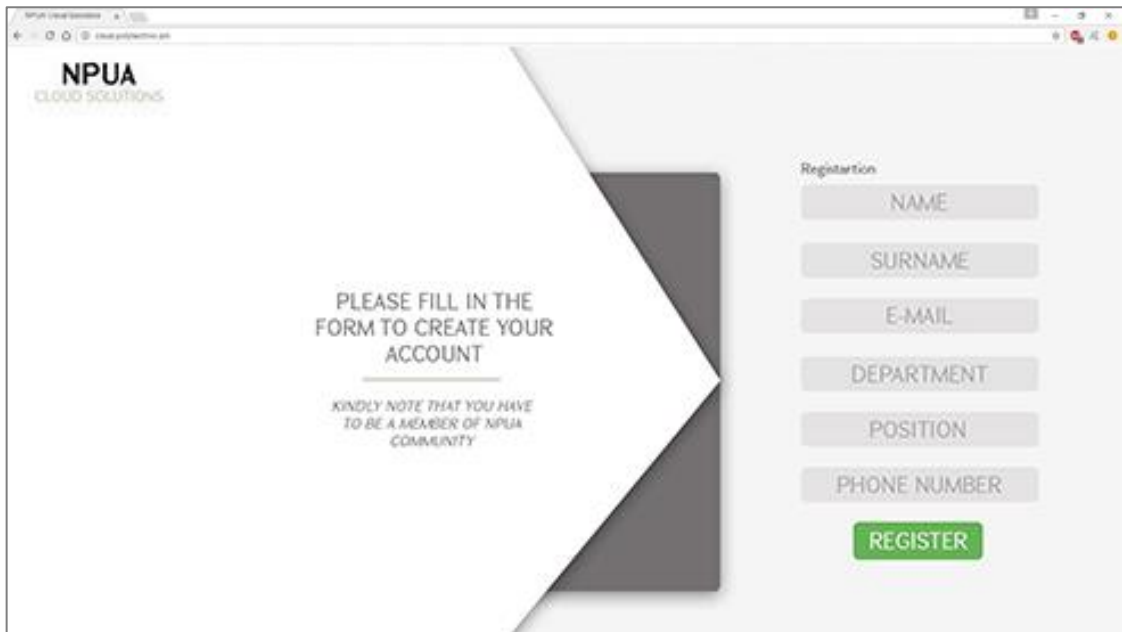
4.1 «ՀԱՊՀ ամպային լուծումներ» ավտոմատացված համակարգի աշխատանքային միջավայրը

«ՀԱՊՀ ամպային լուծումներ» ավտոմատացված համակարգից օգտվելու համար մշակվել է վեբ ինտերֆեյս, որին մուտք գործելու համար կիրառվում է բրաուզեր: Համակարգը մշակվել է AngularJS [26] ծրագրային լեզվով, իսկ որպես տվյալների բազայի կառավարման համակարգ կիրառվել է MySQL [23] բաց ծրագրային կոդով փաթեթը: Յուրաքանչյուր օգտատեր նախագիծ սկսելու համար պետք է նախ գրանցվի համակարգում, որից հետո այն կարող է հաստատվել ադմինիստրատորի կողմից՝ կախված նախապես որոշված պայմանների: Յուրաքանչյուր դեպքում օգտատերը ստանում է նամակ իր էլեկտրոնային փոստարկղի վրա (որը տրամադրվում է գրանցման ժամանակ) հաստատող/կամ մերժող նամակ: Հաստատման պարագայում օգտատերը ստանում է գրանցումը հաստատելու հղում, որի վրա սեղմելով նա վերադարձնում է համակարգ և կարող է պրոյեկտ սկսել: Հակառակ պարագայում օգտատերը ստանում է էլեկտրոնային նամակ, որտեղ նշված են լինում գրանցման մերժման պատճառները: Համակարգի գլխավոր էջն ունի նկ. 3.22- ում բերված տեսքը՝



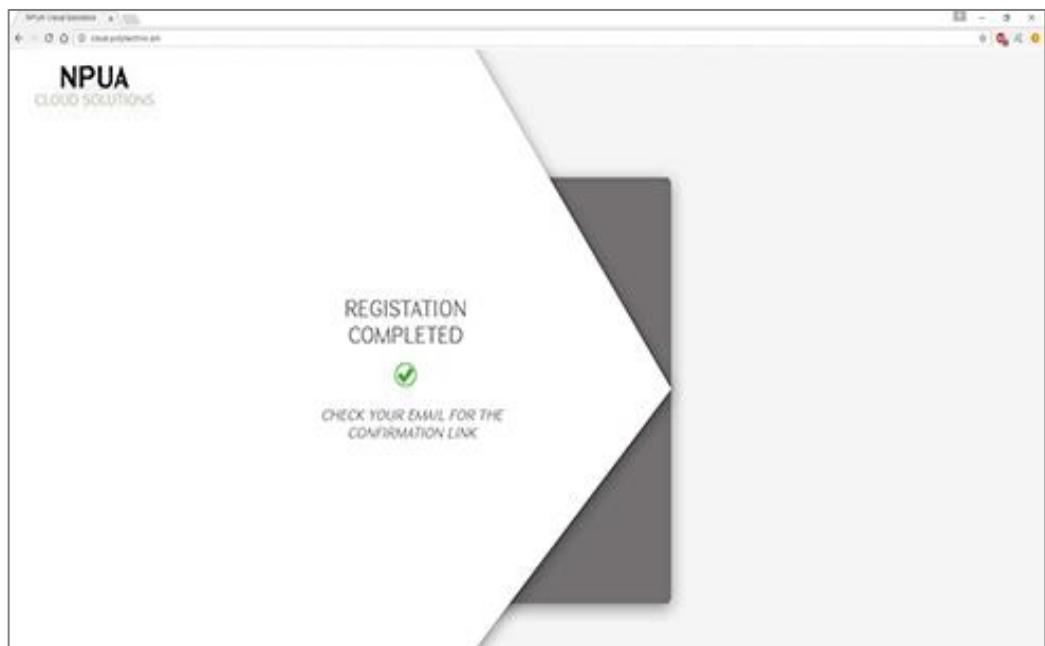
Նկ. 3.22. Գլխավոր էջը

Գլխավոր էջում առկա են համակարգ մուտք գործելու դաշտերը, որտեղ օգտատերը պետք է լրացնի իր էլեկտրոնային փոստարկղի հասցեն և գաղտնաբառը: Եթե օգտատերը դեռ գրանցված չէ, ապա նա պետք է սեղմի մուտքային դաշտերի տակ գտնվող գրանցման կոճակի վրա, որը կբացի հետևյալ էջը (նկ. 3.23).



Նկ. 3.23. Գրանցման էջը

Գրանցման էջում նոր օգտատերը պետք է տրամադրի անուն, ազգանուն, էլեկտրոնային փոստարկղի հասցե, համալսրանում իր աշխատելու կամ սովորելու ստորաբաժանման անունը, զբաղեցրած պաշտոնը և հեռախոսահամար: Բոլոր բաժինները պարտադիր են լրացման համար: Դրանք լրացնելուց հետո օգտատերը ստանում է հաստատման մասին հաղորդագրություն նկ. 3.24 –ում բերված էջի տեսքով:



Նկ. 3.24. Գրանցումը հաստատող էջ

Գրանցման հաստատումը ստանալուց հետո օգտատերը պետք է ակտիվացնի իր էջը, հետևելով էլեկտրոնային փոստին ուղարկված հղմանը: Էջը ակտիվացնելուց հետո օգտատերը կարող է մուտք գործել իր անձնական էջը, որտեղ նա կարող է իրականացնել մի շարք գործողություններ (նկ. 3.25):



Նկ. 3.25. Օգտատիրոջ անձնական էջը

«My Projects»-ի վրա սեղմելով օգտատերը տեսնում է իր ստեղծած ընթացիկ նախագծերը, որտեղից կարող է կառավարել ստեղծված համակարգերը կամ ջնջել դրանք:

«Request Upgrade» բաժնից օգտատերը կարող է դիմել ադմինիստրատորին ռեսուրսների ավելացման համար՝ տրամադրելով բացատրություն: Ադմինիստրատորը ստանալով հարցումը և ուսումնասիրելով տվյալ նախագծի համար նախատեսված ռեսուրսներն ու դրանց ծանրաբեռնվածությունը կարող է հաստատել կամ մերժել հարցումը:

«Maintenance» բաժնի միջոցով օգտատերը կարող է հասկանալ, թե իր կողմից ստեղծված նախագծերն ինչպիսի ծանրաբեռնվածությամբ են աշխատում և ըստ բաժնի տրամադրած վերլուծության կատարել փոփոխություններ:

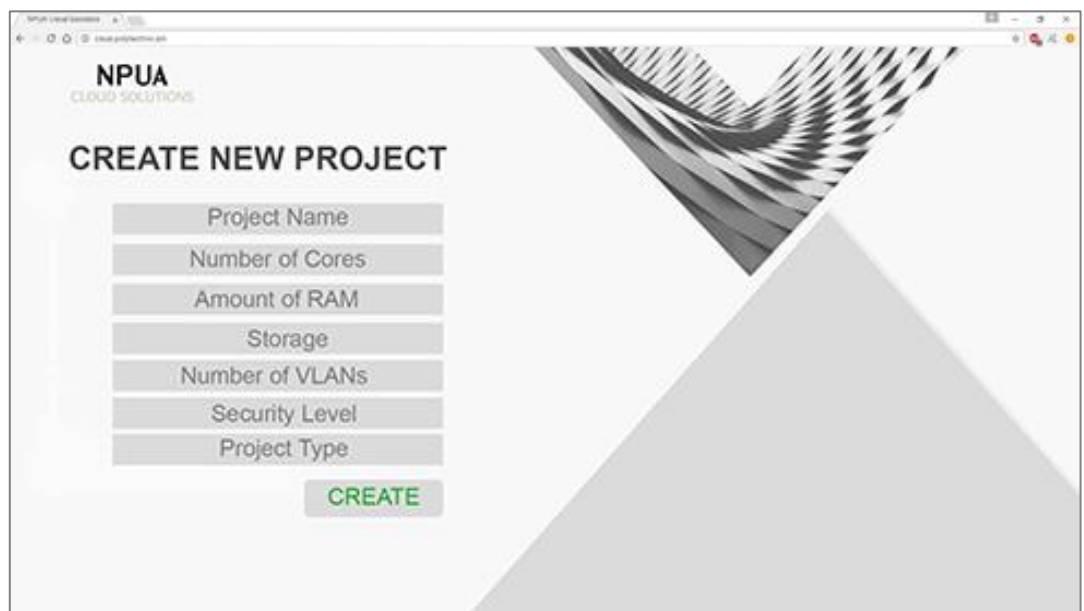
«Report a problem» բաժնի միջոցով օգտատերը դիմում է համակարգի ադմինիստրատորին, եթե նկատել է որևէ խնդիր համակարգի աշխատանքի ընթացքում:

«My Projects» բաժինն այցելելով օգտատերը տեսնում է հետևյալ էջը (նկ. 3.26).



Նկ. 3.26. My Projects էջը

Բաժնում երևում են աշխատող համակարգերի ցուցակը, յուրաքանչյուր համակարգի կարճ նկարագրությունը, աշխատանքային վիճակը, համակարգի խմբագրման կոճակը: Էջի ներքևի հատվածում առանձնացված է «Նոր նախագիծ» ստեղծելու հղումը, իսկ աջ անկյունում տրամադրվում է տեղեկատվություն համակարգերի ընդհանուր աշխատանքային վիճակի մասին: Եթե աշխատող համակարգերից որևէ մեկը խնդիր ունենա, ապա վերոհիշյալ հաղորդագրությունը կփոխարինվի զգուշացմամբ: «Create new project» հղումը սեղմելիս բացվում է հետևյալ էջը (նկ.3. 27).

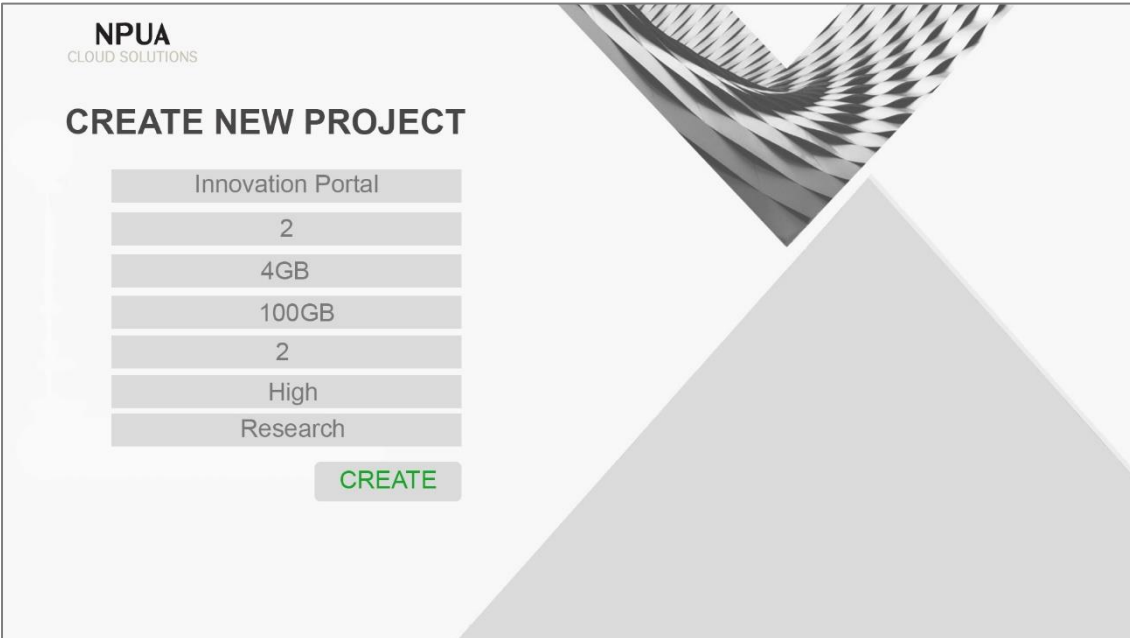


Նկ. 3.27. Նոր նախագիծ ստեղծելու էջը

Համակարգն օգտատիրոջից պահանջում է լրացնել հետևյալ դաշտերը՝

- նախագծի անվանում,
- պրոցեսորների քանակ,
- RAM-ի ծավալ,
- հիշողության ծավալ,
- ցանցային ադապտորների քանակ,
- անվտանգության մակարդակ,
- նախագծի տեսակ:

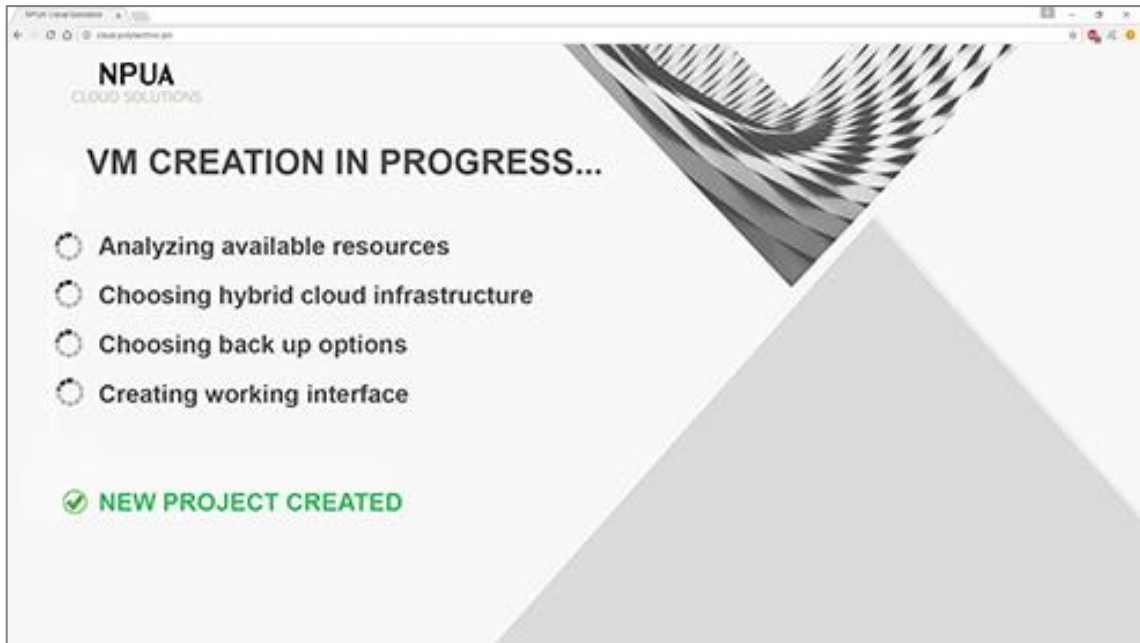
Դաշտերը լրացնելուց հետո, այն ունենում է հետևյալ տեսքը (նկ. 3.28)՝



The screenshot shows a web interface for creating a new project. At the top left, it says 'NPUA CLOUD SOLUTIONS'. Below that is the heading 'CREATE NEW PROJECT'. The form consists of several input fields, each with a light gray background and rounded corners. The fields contain the following text from top to bottom: 'Innovation Portal', '2', '4GB', '100GB', '2', 'High', and 'Research'. Below the last field is a green button with the word 'CREATE' in white capital letters. The background of the interface is light gray with a decorative abstract pattern in the top right corner.

Նկ. 3.28. Նոր նախագծի էջը լրացված դաշտերով

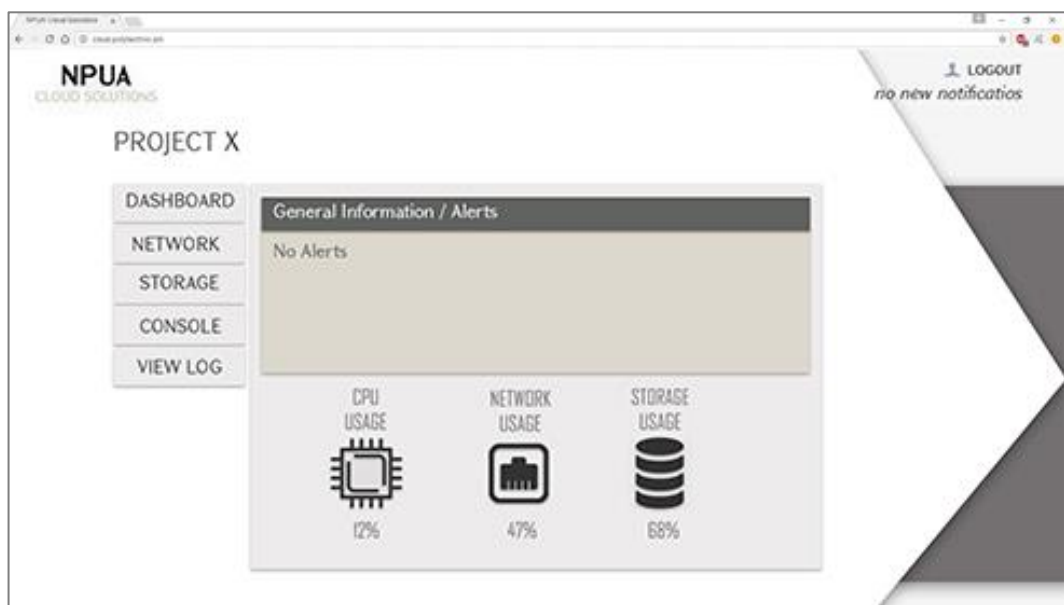
Բոլոր դաշտերում անհրաժեշտ պարամետրերն ընտրելուց և «Create» կոճակը սեղմելուց հետո օգտատիրոջը ցուցադրվում է նոր նախագծի ստեղծման ընթացքը (նկ. 3.29):



Նկ. 3.29. Նոր նախագծի ստեղծման ընթացքը

Մինչ նախագծի ստեղծումը համակարգը ստուգում է նշված պարամետրերը, ըստ որոնց արդեն իսկ որոշում է արդյոք բավարարում են կամ անհրաժեշտ են լրկալ ռեսուրսները, կայացնում է նախագծի տեղաբաշխման վճիռը, որից հետո կառուցում է հիբրիդ ամպի ռեսուրսների սխեման, այնուհետև ընտրում պահուստային տարբերակի ստեղծման պայմաններն ու առաջնահերթությունները: Վերոհիշյալ բոլոր գործողությունները կատարելուց հետո՝ համակարգը ցուցադրում է հաղորդագրություն, որտեղ նշվում է, որ նախագիծը պատրաստ է շահագործման համար:

Որևէ նախագծի էջ մուտք գործելու ժամանակ օգտատերը տեսնում է նկ. 3.30-ում ներկայացված էջը, որտեղ երևում է պրոցեսորների, ցանցի և հիշողության տարածքի ծանրաբեռնվածությունը, ինչպես նաև առանձին էջերից հնարավորություն է ստանում կարգավորել ցանցը, հիշողության տարածքը, ստուգել ընթացիկ գրանցումները և կանչել կոնսոլը, որտեղից կառավարվում է տեղադրված օպերացիոն համակարգը:



Նկ. 3.30 Նախագծի էջը

4.2 Եզրակացություններ 4-րդ գլխի վերաբերյալ

«ՀԱՊՀ ամպային լուծումներ» համակարգի շահագործման համար AngularJS ծրագրային լեզվով, MySQL տվյալների բազայի կառավարման համակարգի կիրառ-

մամբ մշակվել է ավտոմատացման համակարգի վեբ-ինտերֆեյսը: Այն հնարավորություն է տալիս գրանցված օգտատերերին պատվիրել ու ստանալ նախընտրելի պարամետրերով տեղեկատվական ենթակառուցվածքներ ու ծառայություններ:

Մշակվել են օգտատերերի քայլերն իրականացնող երկխոսային պատուհաններ: Օգտատիրոջ նշած պարամետրերի անհամաձայնությունների դեպքերում համակարգի ադմինիստրատորից օգտատերն ստանում է հաղորդագրություններ իրավիճակի ու դրա կարգավորման մասին:

ԵԶՐԱՀԱՆԳՈՒՄ

Ուսումնասիրվել են ամպային տեխնոլոգիաները, դրանց հիմնական տիպերը և մատուցվող ծառայությունների տարատեսակները, ամպային տեխնոլոգիայի տրամադրման համար առկա գործիքամիջոցները: Կատարված ուսումնասիրությունների հիման վրա որոշվել է, որ միջինից մեծ կազմակերպության համար, որտեղ ամպը շահագործվելու է տարբեր տիպի լուծումներ տրամադրելու նպատակով ամենահարմարը հիբրիդային ամպի շահագործումն է: Տվյալների ապահով պահպանման համար ընտրվել են մեծածավալ տվյալների գրանցման/ընթերցման համակարգերը:

Հիբրիդային ամպի ստեղծման ճանապարհով մշակվել և ներդրվել է տեղեկատվատեխնոլոգիական ենթակառուցվածքների ու ծառայությունների տրամադրման ավտոմատացված համակարգ: ՀԱՊՀ-ում հիբրիդային ամպի մատուցման համար ընտրվել է Amazon Web Services հանրային ամպը՝ դրա առաջին տարին անվճար լինելու, մեծ ֆունկցիոնալ ունենալու և Cloudstack-ի հետ ինտեգրման համար նախատեսված API-ների առկայության համար: Ընտրությունը հիմնավորվել է ինչպես ֆինանսական շահավետությամբ, այնպես էլ բազմաֆունկցիոնալ հնարավորությունների առկայությամբ: Ավտոմատացման համակարգում կիրառվել է Ceph օբյեկտային տվյալների պահպանման համակարգը: Ամպային ծառայությունների մատուցման համար ընտրվել է CloudStack-ը: Վերոհիշյալ համակարգերի շահագործումն ապահովվելու է վիրտուալացման հիման վրա, մասնավորապես VMware ընկերության կողմից տրամադրվող ESXi համակարգի միջոցով:

Համակարգում պատվիրված նախագծերի օպտիմալ տեղաբաշխումն ապահովելու նպատակով կիրառվել է վճիռների կայացման կշռային գործակիցների եղանակը, ընդ որում կշռային գործակիցների օպտիմալ արժեքների որոշման համար ձևակերպվել են ոչ գծային ծրագրավորման խնդիրներ: Մշակվել է ալգորիթմ

նախագծի պատվիրված տվյալների և համակարգում առկա ռեսուրսների միջոցով կշռային գործակիցների ստացման, ինտեգրալ գնահատականների հաշվման և վճռի կայացման համար:

Հայաստանի ազգային պոլիտեխնիկական համալսարանում ավտոմատացված համակարգի ներդրման նպատակով մշակվել է ցանցի ճարտարապետությունը, ստեղծվել է հիբրիդային ամպ, որի տարբեր տիպերի ցանցերը տարանջատվել են VLAN-ների միջոցով՝ ցանցի անվտանգությունն ու արագագործությունն ապահովելու նպատակով: Ֆիզիկական սերվերի վրա տեղադրվել է ESXi 6.0 վիրտուալացում ապահովող օպերացիոն համակարգը, որի վրա արդեն տեղադրվել են ներքոհիշյալ համակարգերը որպես վիրտուալ մեքենաներ: Համակարգի անվտանգությունն ապահովելու համար շահագործվել է նաև pfSense ծրագրային firewall-ը, որն ապահովում է նաև երթուղավորում ցանցի մեջ: Տեղադրվել է Ceph օբյեկտային կառավարմամբ տվյալների պահպանման համակարգը՝ իր կլիենտների հետ ներառյալ: Տեղակայվել է Cloudstack համակարգը, որն ապահովում է հիբրիդային ամպի ծառայության մատուցումը:

Հիբրիդային ամպի շահագործման համար ստեղծվել է ավտոմատացման համակարգի վեբ-ինտերֆեյսը՝ օգտատերերի գրանցման, նախագծերի ստեղծման, կառավարման հնարավորություններով:

ԳՐԱԿԱՆՈՒԹՅԱՆ ՑԱՆԿ

1. Վիրապետյան Լ.Գ. Վիրտուալացման դերն ամպային տեխնոլոգիաներում // ՀՃԱ Լրաբեր: 2017. - Հ14, N1. - էջ 144-147
2. Վիրապետյան Լ.Գ. Տվյալների օբյեկտային պահպանման համակարգերի ուսումնասիրություն և ընտրություն // ՀԱՊՀ Լրաբեր. - 2017. - Մաս 1. - էջ 116-122.
3. **Вагнер Г.** Основы исследования операций. В 3-х томах. - М.: Мир 1972.
4. **Лаврентьев А.В.** Оптимизация многокомпонентных приложений в среде облачных вычислений с несколькими провайдерами // Вестник РГГУ. Серия: Документоведение и архивоведение. Информатика. Защита информации и информационная безопасность. 2014. № 11 (133). С. 130-144.
5. **Майзер Х., Эйджин Н., Тролл Р., Моудер Дж., Элмаграби С.** Исследование операций. В 2-х томах. Том 1, - М.: Мир, 1981, - 712с.
6. **Мелузов А.** Сценарии использования облачных сервисов: максимальный эффект от внедрения // Журнал сетевых решений LAN. 2014. № 9. С. 30-31.
7. **Ногин В.Д.** Принятие решений при многих критериях. - СПб.: ЮТАС, 2007. -104 с.
8. **Олексюк В.П.** Внедрение технологий облачных вычислений как составляющих ИТ-инфраструктуры вуза // Информационные технологии и средства обучения. 2014. Т. 41. № 3. С. 256-267.
9. **Паскова А.А., Бутко Р.П.** Гибридные облака в IT-инфраструктуре предприятия // Символ науки. 2016. № 10-2 (22). С. 73-75.

10. **Постников В. М.1 , Спиридонов С. Б.** Методы выбора весовых коэффициентов локальных критериев 1 // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2015. № 06. С. 267–287
11. **Постников В.М., Спиридонов С.Б.** Подход к выбору варианта модернизации сервера ЛВС // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2013. № 2. С. 255-272.
12. **Постников В.М., Черненький В.М.** Методы принятия решений в системах организационного управления. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2014. - 205 с.
13. **Саати Т.Л.** Принятие решений. Метод анализа иерархий: пер. с англ. - М.: Радио и связь, 1993. 316 с.
14. **Сидоренко В.С., Галета А.В.** Условия формирования гибридного облака для современного высшего учебного заведения //В сборнике: Актуальные научные вопросы и современные образовательные технологии сборник научных трудов по материалам Международной научно-практической конференции: в 7 частях. 2013. С. 125-126.
15. **Сморodinский С.С., Батин Н.В.** Оптимизация решений на основе методов и моделей математического программирования Минск: БГУИР, 2003, - 138с.
16. **Таха А.Х.** Введение в исследование операций. -7-е изд.-М.: Вильямс. - 2005. - 912 с.
17. **Царегородцев А.В., Лось А.Б., Сорокин А.В.** Построение гибридной защищенной облачной среды ИТ-инфраструктуры организации //Промышленные АСУ и контроллеры. 2015. № 11. С. 26-31.

18. **Чадин А.С.** Динамическая оптимизация нагрузки на вычислительных узлах частных, публичных и гибридных облаков //Труды Института системного программирования РАН. – М.: 2015, том 27, вып. 6, стр. 307-314.
19. "The NIST Definition of Cloud Computing" (PDF). *National Institute of Science and Technology*. 2011
20. **A. Marinos, G. Briscoe**, Community cloud computing, *Cloud Computing* (2009) 472–484.
21. **A. Paradowski, L. Liu and B. Yuan**, "Benchmarking the Performance of OpenStack and CloudStack," *2014 IEEE 17th International Symposium on Object/ Component/Service-Oriented Real-Time Distributed Computing*, Reno, NV, 2014, pp. 405-412.doi: 10.1109/ISORC.2014.12
22. **Abels T, Dhawan P, Chandrasekaran B** An overview of xen virtualization (2009). <http://www.dell.com/downloads/global/power/ps3q05-20050191-Abels.pdf>
23. Baron Schwartz, Peter Zaitsev, Vadim Tkachenko *High Performance MySQL: Optimization, Backups, and Replication* // O'Reilly, 2012
24. **Ben Belgacem M., Chopard B.** A hybrid HPC/cloud distributed infrastructure: Coupling EC2 cloud resources with HPC clusters to run large tightly coupled multiscale applications //FUTURE GENERATION COMPUTER SYSTEMS-THE INTERNATIONAL JOURNAL OF ESCIENCE, Volume: 42, - P.11-21.
25. **Bittencourt LF, Senna CR, Madeira ERM** (2010) Enabling execution of service workflows in grid/cloud hybrid systems. In: International workshop on cloud management (Cloudman). IEEE Computer Society Press, Osaka
26. Brad Green, Shyam Seshadri *AngularJS*, O'Reilly Media, 2013
27. Chandrasekaran, Arun, Dayley, Alan). *Critical Capabilities for Object Storage*. Gartner Research, 2014.

28. **Cho K.M., Tsai, P.W., Tsai C.W., Yang C.S.** A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing //NEURAL COMPUTING & APPLICATIONS. 2015, Volume: 26, Issue: 6, - P. 1297-1309.
29. **Delimitrou C., Kozyrakis C.** HCloud: Resource-Efficient Provisioning in Shared Cloud Systems // ACM SIGPLAN NOTICES, Volume: 51, Issue: 4. - P. 473-488
30. **Edwin S.H. Hou, Ninvan Ansari, Hong Ren** A Genetic Algorithm for Multiprocessor Scheduling // IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. VOL. 5, NO. 2, FEBRUARY 1994. P. 113-120
31. **Gabriella Laatikainen, Oleksiy Mazhelis, Pasi Tyrvaainen** “Cost benefits of flexible hybrid cloud storage: Mitigating volume variation with shorter acquisition cycle” Future Generation Computer Systems Volume 67, February 2017, Pages 35–46
32. **Gandhi, R ;Liu, HH ;Hu, YC ;Lu, GH ;Padhye, J ;Yuan, LH ;Zhang,** Cloud Scale Load Balancing with Hardware and Software // ACM SIGCOMM COMPUTER COMMUNICATION REVIEW, Volume: 44, Issue: 4, P. 27-38
33. **Giovanni Parmigiani, Lurdes Inoue** Decision Theory: Principles and Approaches. John Wiley & Sons, 2014, 372 p.
34. **Hui He, Yana Feng, Zhigang Li, Zhenguang Zhu, Weizhe Zhang, Albert Cheng** “Dynamic Load Balancing Technology for Cloud-oriented CDN”, Computer Science and Information Systems 12(2):765–786
35. **Hyun Jung La and Soo Dong Kim**, A Systematic Process for Developing High Quality SaaS Cloud Services p. 281
36. **J. Li, Y. Jia, L. Liu, T. Wo**, "CyberLiveApp: A Secure Sharing and Migration Approach for Live Virtual Desktop Applications in a Cloud Environment", *Future Generation Computer Systems Elsevier Science*, vol. 29, pp. 330-340, 2013.

37. **Jinho Hwang** “Towards Beneficial Transformation of Enterprise Workloads To Hybrid Clouds” IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, VOL. 13, NO. 2, JUNE 2016
38. **Kai Zheng, Lin Wang, Baohua Yang, Yi Sun, and Steve Uhlig** “LazyCtrl: A Scalable Hybrid Network Control Plane Design for Cloud Data Centers” IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 28, NO. 1, JAN. 2017 1
39. **Kirakosyan L.G., Nazaretyan Kh.A. The cloud system deployment and management in National Polytechnic University of Armenia // NAS RA and NPUA SERIES OF TECHNICAL SCIENCES, 2016. Vol. 69, N4. – P. 439-446.**
40. **Kirakosyan L.G., Avetisyan A.G., Automation system of the university information services //Proceedings of NPUA, Series Information Technologies, Electronics, RadioEngineering, 2017, N1, p.**
41. **L. Liu, D.-A. DaSilva, N. Antonopoulos, Z. Ding, Y. Zhan,** "Achieving Green IT Using VDI in Cyber Physical Society", *Journal of Internet Technology*, vol. 14, pp. 413-424, 2013.
42. **L. Liu, O. Masfary, N. Antonopoulos,** "Energy Performance Assessment of Virtualization Technologies Using Small Environmental Monitoring Sensors", *Sensors*, vol. 12, pp. 6610-6628, 2012.
43. **Karan Singh** Learning Ceph - Packt Publishing Ltd, 2015.- p. 244.
44. **Li, S ; Zhou, YF; Jiao, L ; Yan, XY ; Wang, X ; Lyu, MRT** Towards Operational Cost Minimization in Hybrid Clouds for Dynamic Resource Provisioning with Delay-Aware Optimization // IEEE TRANSACTIONS ON SERVICES COMPUTING, Volume: 8, Issue: 3, Pages: 398-409.
45. **Luiz Fernando Bittencourt, Edmundo Roberto Mauro Madeira** “HCOC: a cost optimization algorithm for workflow scheduling in hybrid clouds”

46. **Mahjoub, M., Mdhaffar, A., Halima, R.B., Jmaiel, M.** "A Comparative Study of the Current Cloud Computing Technologies and Offers" Network Cloud Computing and Applications (NCCA), 2011 IEEE
47. **Mann, ZA** Allocation of Virtual Machines in Cloud Data Centers-A Survey of Problem Models and Optimization Algorithms // ACM COMPUTING SURVEYS. Volume: 48, Issue: 1, Article Number: 11
48. **Martin Peterson** An Introduction to Decision Theory. Cambridge University Press, 2009, 617p.
49. **Mesnier, Mike; Gregory R. Ganger; Erik Riedel** (August 2003). "Object-Based Storage" . IEEE Communications Magazine: 84–90.
50. **Michael Menzel, Rajiv Ranjan, Lizhe Wang, Samee U. Khan, Jinjun Chen** "CloudGenius: A Hybrid Decision Support Method for Automating the Migration of Web Application Clusters to Public Clouds" JOURNAL OF LATEX CLASS FILES, VOL. 6, NO. 1, JANUARY 2007
51. **MOHAMMAD HAMDQA AND LADAN TAHVILDARI** Cloud Computing Uncovered: A Research Landscape, page 44
52. **Oleksyuk V.** Experience of the organization of virtual laboratories on the basis of technologies of cloud computing //Інформаційні технології в освіті. 2014. № 20. С. 127-137.
53. **Padala P.**, "Resource Management in VMware Powered Cloud: Concepts and Techniques," *2013 IEEE 27th International Symposium on Parallel and Distributed Processing*, Boston, MA, 2013, pp. 581-581.
54. **Phillips C, Stein C, Wein J** (1997) Task scheduling in networks. SIAM J Discrete Math 10(4):573–598
55. **Pinedo ML** (2008) Scheduling: theory, algorithms, and systems. Springer, Berlin

56. **Porter De Leon, Yadin; Tony Piscopo** Object Storage versus Block Storage: Understanding the Technology Differences, **2016**.
57. **Pradeep Padala, Kai-Yuan Hou, Kang G. Shin, Xiaoyun, Mustafa Uysal, Zhikui Wang, Sharad Singhal, Arif Merchant** “Automated Control of Multiple Virtualized Resources”
58. **Qiu, XJ; Li, HX; Wu, C ; Li, ZP ; Lau,** FCM Cost-Minimizing Dynamic Migration of Content Distribution Services into Hybrid Clouds //IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. Volume: 26, Issue: 12, Pages: 3330-3345
59. **Renato L.F. Cunha, Eduardo R. Rodrigues, Leonardo P. Tizzei, Marco A.S. Netto** “Job placement advisor based on turnaround predictions for HPC hybrid clouds” Future Generation Computer Systems Volume 67, February 2017, Pages 35–46
60. **Sangwon Lee, Kwang-Kyu Seo** “A Hybrid Multi-Criteria Decision-Making Model for a Cloud Service Selection Problem Using BSC, Fuzzy Delphi Method and Fuzzy AHP” Wireless Pers Commun DOI 10.1007/s11277-015-2976-z
61. **Soodeh Farokhi, Pooyan Jamshidi, Ewnetu Bayuh Lakewc, Ivona Brandic, Erik Elmroth** “A hybrid cloud controller for vertical memory elasticity: A control-theoretic approach” Future Generation Computer Systems Volume 65, December 2016, Pages 57–72
62. **Sushil Bhardwaj, Leena Jain, Sandeep Jain.** CLOUD COMPUTING: A STUDY OF INFRASTRUCTURE pp 60-63 International Journal of Engineering and Information Technology, 2010
63. Taxonomy, Classification & Implementation of open source cloud computing platforms; CS 692 R&D Project Report; Arpit Malani (10305901); May 16, 2012 pdf

64. **Thomas Sandholm and Dongman Lee** Notes on Cloud computing principles, p 2, *Journal of Cloud Computing: Advances, Systems and Applications* 2014, **3**:21
65. **Youk M.A.** Cloud Computing Issues Research and Implementations // *Journal of Computing and Information Technology*. – 2008. -16.- P. 235–246
66. **William Y. Chang, Hosame Abu-Amara, Jessica Feng Sanford**, *Transforming Enterprise Cloud Services*, London: Springer, 2010, pp. 55-56
67. **Yoji Yamato** “Cloud Storage Application Area of HDD–SSD Hybrid Storage, Distributed Storage, and HDD Storage”, *IEEJ TRANSACTIONS ON ELECTRICAL AND ELECTRONIC ENGINEERING IEEJ Trans* 2016; **11**: 674–675
68. **Yu Sun, Jules White, Bo Li Michael Walker, Hamilton Turner** “Automated QoS-oriented cloud resource optimization using containers” *Autom Softw*
69. **Zhou, JJ ; Yao, XF** A hybrid artificial bee colony algorithm for optimal selection of QoS based cloud manufacturing service composition // *INTERNATIONAL JOURNAL OF ADVANCED MANUFACTURING TECHNOLOGY*. Volume: 88, Issue: 9-12, Pages: 3371-3387
70. <https://www.dropbox.com>
71. <https://azure.microsoft.com/en-us/?b=17.06>
72. https://aws.amazon.com/solutions/?nc2=h_ql_ny_livestream_blu
73. <https://cloud.google.com/>
74. <http://searchdatacenter.techtarget.com/tip/Advantages-and-drawbacks-to-public-cloud-services>
75. <http://searchcloudcomputing.techtarget.com/definition/private-cloud>
76. http://blogs.gartner.com/lydia_leong/2011/11/28/private-clouds-arent-necessarily-more-secure/
77. <https://www.pwc.co.uk/assets/pdf/2015-isbs-technical-report-blue-03.pdf>

78. https://en.wikipedia.org/wiki/Health_Insurance_Portability_and_Accountability_Act
79. https://en.wikipedia.org/wiki/Data_Protection_Directive
80. <http://www.forbes.com/sites/ciocentral/2012/01/19/storing-data-in-the-cloud-raises-compliance-challenges/#574e666c664a>
81. <https://www.ibm.com/blogs/cloud-computing/2014/01/what-is-hybrid-cloud/>
82. <https://www.idc.com/getdoc.jsp?containerId=prUS25558515>
83. <https://support.rackspace.com/white-paper/understanding-the-cloud-computing-stack-saas-paas-iaas/>
84. <https://xenserver.org/about-xenserver-open-source.html>
85. <https://www.virtualbox.org/>
86. <https://www.linux-kvm.org/page/Documents>
87. <http://www.vmware.com/ru/products/cloud-foundation.html>
88. <https://cloud.google.com/compute/>
89. <http://www.infoworld.com/article/2638433/virtualization/windows-azure-s-hypervisor-isn-t-hyper-v.html>
90. <http://ceph.com/>
91. <https://www.gluster.org/>
92. <http://docs.ceph.com/docs/hammer/architecture/>
93. <http://gluster.readthedocs.io/en/latest/Quick-Start-Guide/Architecture/>
94. <https://aws.amazon.com/what-is-aws/>
95. <https://aws.amazon.com/pricing/>
96. <http://docs.aws.amazon.com/AWSEC2/latest/APIReference/Welcome.html>
97. [https://en.wikipedia.org/wiki/Firewall_\(computing\)](https://en.wikipedia.org/wiki/Firewall_(computing))
98. <https://www.pfsense.org/>
99. <https://pfsense.org/products/>

100. <http://www.infoworld.com/article/2861574/network-security/you-should-be-running-pfsense-firewall.html>
101. <http://www.datacenterknowledge.com/archives/2015/02/23/openstack-vs-cloudstack-the-platforms-and-the-cloud-apis/>
102. https://en.wikipedia.org/wiki/Apache_CloudStack
103. <http://kangaroot.net/blog/cloudstack-open-source-cloud-computing>
104. <https://software.intel.com/en-us/articles/the-advantages-of-using-virtualization-technology-in-the-enterprise>
105. <http://www.serverwatch.com/server-trends/slideshows/top-10-virtualization-technology-companies-for-2016.html>
106. <https://www.vmware.com/pdf/virtualization.pdf>
107. https://www.vmware.com/pdf/vi_architecture_wp.pdf
108. <http://docs.cloudstack.apache.org/projects/cloudstack-installation/en/4.9/hypervisor/vsphere.html>
109. <http://www.intel.com/content/www/us/en/virtualization/virtualization-technology/intel-virtualization-technology.html>