



LATVIJAS DATU GRID

PROJEKTA OTRĀ GADA REZULTĀTI

Dokumenta faila vārds:	DatuGridv1.doc
Aktivitāte:	3.aktivitāte "Lietojumprogrammu un lielu datu apjomu izmantošanas risinājumu izstrāde Grid vidē"
Projekta numurs:	VPD1/ERAF/CFLA/05/APK/2.5.1./000055/027
Organizācija:	Latvijas Universitātes aģentūra "Latvijas Universitātes Matemātikas un informātikas institūts"
Autori:	Dana Ludviga, Kaspars Krampis, Baiba Kaškina, Katrīna Sataki

Anotācija:

Šis dokuments apraksta Latvijas Datu Grid (turpmāk tekstā Datu režģis) izveides iniciatīvu, procesus un mērķus. Dokumentā apkopota informācija par Latvijas Nacionālās izpētes un izglītības tīkla SigmaNet sadarbību ar TERENA dibināto datu režģa izveides grupu TF-Storage, šīs sadarbības rezultātā iegūto pieredzi un tehniskos veikumus izvēlētā izkliedēto datu glabātuvju risinājumu *Cleversafe* ieviešanā un testēšanā.

Saturs

1. IEVADS	3
2. PAŠREIZĒJĀ EIROPAS INICIATĪVA DATU REŽĢA IZVEIDĒ	4
3. DATU REŽĢIS	6
4. WUALA	7
5. CLEVERSAFE.....	11
6. IZANALIZĒTO PRODUKTU SALĪDZINĀJUMS.....	20
7.CLEVERSAFE UZSTĀDĪŠANA	22
8. SECINĀJUMI UN DATU REŽĢA TURPMĀKĀ ATTĪSTĪBA	32

1. IEVADS

Zinātnei strauji attīstoties, ir ievērojami pieaudzis cilvēcei nozīmīgo datu apjoms. Tiek lēsts, ka pasaules kopējais radītās un apkopotās informācijas apjoms gadā veido aptuveni 5 - 10 eksabaitus (turpmāk tekstā EB). Tieši šis dinamiskais datu pieaugums ir iemesls, lai radītu Datu režģi - izplatītu, drošu, plaši pieejamu un replicētu dalīto datu krātuvi, kas nodrošinātu zinātniekiem iespēju koplietot un uzglabāt savu pētījumu rezultātus.

Projekta otrā gada darbības ietvaros veicot Datu režģa izveides iespējas izpēti, Latvijas Nacionālās izpētes un izglītības tīkls SigmaNet aktīvi ņēma dalību Eiropas pētniecības un izglītības tīklu asociācijas TERENA izveidotajā darbu grupā TF-Storage. Balstoties uz iegūto informāciju, SigmaNet izstrādāja konkrētas Datu režģa prasības, izpētīja vairākus izvirzītajām prasībām atbilstošus dalītas datu glabātuves risinājumus, no kuriem visveiksmīgākais tika arī realizēts un iztestēts, tādā veidā veicot pirmos soļus Datu režģa izveidei un ieviešanai Latvijā.

2. PAŠREIZĒJĀ EIROPAS INICIATĪVA DATU REŽĢA IZVEIDĒ

Šajā nodaļā tiek izklāstīti novērojumi darbojoties TERENA vadītajā datu glabātuvju sadarbības grupā TF-Storage. Nodaļa sniedz ieskatu Eiropā aktuālākajās problēmās, jaunumos un iniciatīvās izklaidēto datu glabātuvju jomā.

2.1. PAŠREIZĒJĀ SITUĀCIJA

Attīstoties lielaudas pieslēgumiem, paveras plašas iespējas zinātnieku resursu koplietojamībai. Skaitļošanas režģa izveidotā infrastruktūra ļauj zinātniekiem veikt savus pētījumus un liela apjoma aprēķinus uz režģī saslēgtajām skaitļošanas mašīnām. Zinātnieki paši apliecina, ka šāda veida infrastruktūra ļauj viņu fantāzijai vaļū, līdz ar to uzlabojot pētījumu rezultātus.

Nākamais Eiropas iniciatīvas solis ir, balstoties uz šo jau izveidoto infrastruktūru, nodrošināt ne tikai skaitļošanas, bet arī drošas datu uzglabāšanas un koplietošanas iespējas. Šādu infrastruktūru dēvē par Datu režģi (*Data Grid*).

Pasaulē ir veiksmīgi realizēti jau vairāki Datu režģa projekti, taču pārsvarā tie ir bijuši radīti kāda noteikta zinātniska pētījuma projekta ietvaros, līdz ar to ir izmantojami tikai vienas nozares vai pat viena atsevišķa projekta vajadzībām.

Datu režģa definīcija un uzbūve vēl joprojām nav precīzi definētas, atkarībā no konkrētās izveides nepieciešamības tās mēdz krasi atšķirties. Tādēļ Eiropas pētniecības un izglītības tīklu asociācija TERENA šā gada 25. februārī ir radījusi glabātuves sadarbības darba grupu (*Storage Collaboration task force*), kurā sadarbojas vairāki Eiropas valstu nacionālie pētniecības un izglītības tīkli (*NREN National Research and Education Network*). Darba grupas mērķis ir definēt Eiropas zinātnieku datu uzglabāšanas un koplietošanas prasības, mērķus, kā arī izpētīt tehniskos risinājumus kopīgas, starptautiskas datu glabātuves nodrošināšanai. Šādu Eiropas iniciatīvu atbalsta arī Latvijas pētniecības un izglītības tīkls (*SigmaNet*), veicot eksperimentus un pētījumus dalītu datu glabātuvju sistēmu ieviešanā.

Eiropas interese par kopīga Datu režģa izveidi ir izrādīta salīdzinoši nesen. Pirmās divas vēl tikai domubiedru sanāksmes notika:

- Pirmā glabātuves sadarbības sanāksme (*Collaboration on Storage Services*), Amsterdamā, 29.06.2007;
- Otrā glabātuves sadarbības sanāksme (*Collaboration on Storage Services 2nd meeting*), Amsterdamā, 14.12.2007;

Latvija savu iniciatīvu un ieinteresētību šāda Datu režģa izveidē izrādīja, piedaloties jau otrā glabātuves sadarbības sanāksmē. Balstoties uz plašo domubiedru klāstu 25.02.2008 TERENA akceptēja un noformēja atsevišķu glabātuves darba grupu, nodēvējot to par TF-Storage. Pirmā darba grupas sanāksme (*1st TF-Storage Meeting*) notika Helsinkos, 08.04.2008.

Šajā sanāksmē Latvijas izrādītā ieinteresētība tika novērtēta, akceptējot nākamās kopējās darba grupas sanāksmes norises vietu Rīgā, LU MII telpās (12.09.2008). Sīkāku informāciju par šo sanāksmi var iegūt, apskatot mājas lapu tfstorage.lumii.lv.

Iepriekš minētajās sanāksmēs tika apskatīti vairāki vispārēji Eiropas datu glabātuves un datu centru problemātiskie jautājumi. Tādi kā:

-
- Lielais enerģijas patēriņš;
 - Datu aprakstīšana (meta dati);
 - Autorizācijas un autentifikācijas infrastruktūra (turpmāk tekstā AAI);
 - Nepārtrauktības nodrošināšana;
 - Datu koplietošanas politika.

Problemātiskie jautājumi, pie kuriem strādā arī Latvijas pārstāvji ir:

- Datu glabātuvju testēšanas etalonmodeļa izveide - testējot ieviesto infrastruktūru, izstrādātājs nevar būt pilnībā pārliecināts, vai tā strādā tik labi, cik spēj. Līdz ar to rodas jautājums, vai tiešām mēs varam uzticēties ierīču izstrādātāju veikspējas rādītājiem, un vai tie nebūtu pirms lietošanas arī jānotestē? Šī izveidotā darba grupa nodarbojas ar vadlīniju izstrādāšanu datu infrastruktūras ātrdarbības testēšanai. Tās mērķis ir izanalizēt un salīdzināt apkopotos datus, lai izstrādātāji varētu veiksmīgāk novērtēt aparatūras veikspēju;
- Glabātuves taksonomijas izstrāde - apkopojot ekspertu vērtējumus par dažādu datu glabātuves sastāvdaļām, izveidot profesionāli izvērtētu datu glabātuves klasifikāciju (analizējot dCache, iRods, SRM, SamFS, u.c.).

3. DATU REŽĢIS

Izvērtējot Eiropas Datu režģa vīziju, var teikt, ka tā ir vēlme izveidot starptautisku, visā Eiropā izdalītu datu glabātuves sistēmu, kas ļautu visu Eiropas zinātnieku nozaru pārstāvjiem veiksmīgāk sadarboties. Atšķirībā no standarta tīmekļa servera Datu režģis ir zinātniekiem veltīta dalīta datu glabātuve, kas domāta lai savietojot ar skaitļošanas režģi savas mērījumu iekārtas vai pētījumu datus, veidotu ērtu un, galvenais, drošu pētījumiem veltītu izpildes infrastruktūru. Lai veiksmīgi apmierinātu visu zinātnieku vēlmes pētījumu datu uzglabāšanā, dalītai datu glabātuvei ir jābūt drošai, pieejamai un ērti lietojamai.

Šajā nodaļā, balstoties uz pašreiz neskaidro Eiropas vispārējo Datu režģa formulējumu, ir izveidotas un attēlotas mūsu definētās nepieciešamās Datu režģa prasības, pēc kurām tika piemeklēti jau realizēti risinājumi, kas būtu par pamatu Datu režģa izveidei.

3.1. DATU REŽĢA PRASĪBAS.

1. Sadalīta un satur replikas - viens no svarīgākajiem aspektiem ir datu drošība. Lai to nodrošinātu, glabātuvē ievietotais fails, neskatoties uz to cik tas būtu apjomīgs, nedrīkst tikt uzglabāts tikai uz viena servera. Tam ir jābūt sadalītam pa vairākām aparatūrām un jā satur replikas, lai nelaimes gadījumā, ja viena vai vairākas iekārtas iziet no ierindas vai vienkārši nav sasniedzamas, fails vēl joprojām būtu pieejams un nebojāts;

2. Droši šifrēta - datu drošība nevar būt iedomājama bez šifrēšanas. Šifrēšana var būt droša tikai tad, ja šifrēšanas atslēga nepamet lietotāja datoru, proti, tā notiek lokāli uz lietotāja aparatūras. Tādā gadījumā arī sadalītais fails, nokļūstot nepareizās rokās, tā oriģinālo saturu neatklāj;

3. Plaši pieejama - ņemot vērā šīs datu glabātuves plašo pielietojamību, tai ir jābūt viegli pieejamai no dažādiem lietojumiem. Dati var tikt iesūtīti datu bāzē tieši no pētniecības iekārtas vai jau no eksistējošas datu bāzes. Tai ir jābūt pieejamai no jebkuras vietas;

4. Kontrolēta, droša datu koplietošana - datu glabātuvē ievietotie šifrētie dati var tikt arī koplietoti. Proti, datu īpašnieks var atļaut zināmai lietotāju grupai piekļūt viņa datiem (piemēram, daloties ar savu draudzības šifrēšanas atslēgu);

5. Veiktspēja – ņemot vērā plašo datu pieejamību un pielietojumu, datu izgūšana un ievietošana datu glabātuvē ir jāveic pēc iespējas ātrāk, vēlams diska ātruma robežās;

6. Saskaņotība kā failu sistēmai - tā kā lietotāju pulks šādai datu glabātuvei ir ļoti plašs un ne vienmēr tie ir informācijas tehnoloģijas pārstāvji, datu pievienošanai un apskatei ir jābūt pēc iespējas vienkāršākai. Apjomīgu datu pievienošana varētu aprobežoties ar cietā diska piemontēšanu. Saskaņotībai ir jābūt līdzīgai standarta failu sistēmai;

7. Atvērtā koda - izveidotajam risinājumam ir jābūt atvērta koda, lai to varētu lēti ieviest un ar to eksperimentēt, kā arī nepieciešamības gadījumā arī pielabot un pielāgot.

8. Sertifikācijas pārvaldes (CA - *Certification Authority*) atbalsts (neobligāts) - ņemot vērā jau skaitļošanas režģa izstrādāto autorizācijas infrastruktūru datu piekļuvei varētu tikt izmantota publiskā sertifikāta ierobežošana, tādā veidā veidojot kopīgu datu režģa un skaitļošanas režģa autorizācijas sistēmu.

Pēc vairāku iespējamo dalītās datu glabātuves risinājumu izpētes analīzei un salīdzināšanai tika izvēlēti divi jauni projekti – *Wuala* un *Cleversafe*, kuru apraksts un salīdzinājums ir attēlots sekojošās nodaļās.

4. WUALA

Šajā nodaļā tiek izklāstīta dalītās tiešsaistes datu glabātuves projekta un produkta *Wuala* uzbūve un analīze.

Projekta *Wuala* mērķis ir nodrošināt lielu, uzticamu un drošu dalītu tiešsaistes datu krātuvi, kas var izmantot lietotāju datoru brīvos resursus. Tā ir bezmaksas programmatūra, kas balstīta uz Java, līdz ar to darbojas uz dažādām platformām - *Windows*, *Mac* un *Linux*, nodrošinot ērtu un drošu tiešsaistes datu glabātuvi. *Wuala* pamatā ir daļēji centralizēts vienādranga (*peer to peer*) tīkls, kas piedāvā noteikta apjoma bezmaksas tiešsaistes cieto disku, pie kura lietotājs var piekļūt jebkurā laikā.

Izmantojot *Wuala*, lietotājs:

1. var ielādēt jebkuru failu, neskatoties uz to, cik tas būtu apjomīgs, un dalīties ar to (ar draugiem, draugu grupām vai pat publiski ar visu pasauli);
2. piekļūt failiem no jebkuras vietas.

Kad produktu sāk izmantot kāds jauns lietotājs, viņam(-ai) tiek piešķirta un nodrošināta 1 gigabaitu (turpmāk tekstā GB) apjomīga tiešsaistes datu krātuve. Taču ja lietotājs šo apjomu vēlas palielināt, to var veikt, nomainot glabātuvi. Glabātuves maiņa notiek šādi - lietotājs apmaiņā pret lielāku tiešsaistes atmiņas apjomu piedāvā koplietošanai sava lokālā diska apjomu. Tomēr ir obligāts nosacījums: lietotājam ir jābūt tiešsaistē vismaz 17% no laika, kas ir aptuveni 4 stundas dienā. Lemesis šai prasībai - iemainītā lokālā diska vieta kļūst neizmantojama, ja tā reti atrodas tiešsaistē.

Veidojot publisku, dalītu tiešsaistes datu krātuvi ir jāņem vērā šādi aspekti:

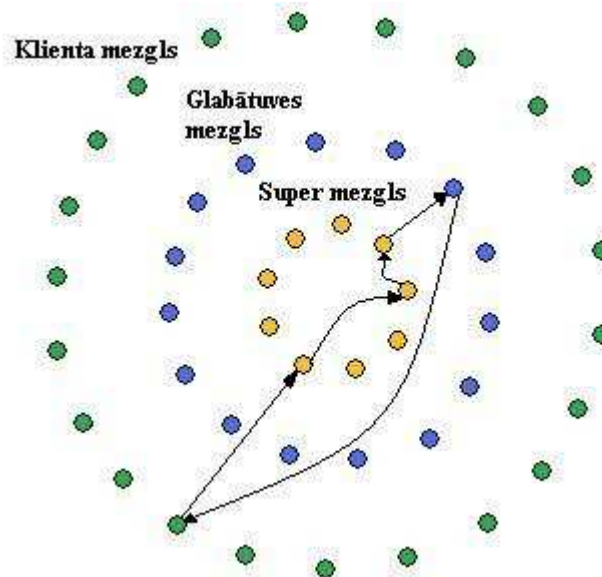
1. **Privātums** - fiziskas personas vai organizācijas tiesības kontrolēt vai noteikt, kādu informāciju par to drīkst uzkrāt un saglabāt un kam šo informāciju ir atļauts izmantot. *Wuala* šo problēmu risina izmantojot datu šifrēšanu:
 - visi faili tiek šifrēti uz lietotāja datora;
 - lietotāja parole nekad nepamet datoru;
 - nevienam nav pieejas un līdz ar to arī nojausmas, kādus failus lietotājs glabā un dalās ar saviem draugiem.
2. **Datu pieejamība** - programmas vai lietotāja potenciālā spēja lasīt datus. *Wuala* šo problēmu risina, izmantojot redundanci¹. Tiek izmantoti speciāli "*erasure codes*", kas nodrošina apjoma optimālu datu redundanci. Tradicionāli ir tikuši izmantoti sakaru sistēmās, taču nesenāk arī glabātuvju sistēmās, kā alternatīva replikācijai. Ar šī algoritma palīdzību ir iespējams šifrēt faila m fragmentus n -tos gabalos, kur $m < n$. Lai atjaunotu šādu failu, ir nepieciešami jebkuri m no n . Šos algoritmus izmanto arī turpvērstā kļūdu labošanā. Eksistē vairāki šī algoritma veidi, *Wuala* izmanto Rīda-Solomona² (*Reed-Solomon*) grupai piederošos.

¹ Dažāda rakstura papildelementu kopums, ko iestrādā sistēmā tās darbības uzticamības paaugstināšanai. Izmanto, piemēram, aparatūras, informacionālo un algoritmisko redundanci. Aparatūras redundanci var realizēt, dublējot sistēmas aparatūras elementus.

² Svarīga lineāro blokkodu saime, ko parasti izmanto sprādzienkļūdu labošanai. RīdaSolomona kodi ir viens no ciklisko kodu paveidiem.

3. **Maršrutēšana** - maršrutēšanas princips vienādranga tīklā ir ļoti būtisks ātrdarbības un drošības faktors. Vairāki jau realizēti projekti ir cietuši neveiksmi tieši nepareizi izvēlētas maršrutēšanas principa dēļ. Lai izvairītos no tādām maršrutēšanas problēmām kā centralizēšanas un pieprasījumu apraide, tika aizsākta super mezglu (*super nodes*) attīstība. Super mezgli ir atbildīgi par kaimiņu mezglu maršrutēšanu un vaicājumu pārvaldību. Sīkāk par tā darbību un specifiku tiek izklāstīts programmatūras *Wuala* arhitektūras aprakstā.
4. **Arhitektūra** - liela daļa programmatūras *Wuala* veiksmīgās darbības slēpjas tā arhitektūrā. Tīkls sastāv no dažādiem mezgliem (*nodes*):
 - super mezgli (*super nodes*), kuri ir atbildīgi par maršrutēšanu;
 - glabāšanas mezgli (*storage nodes*), uz kuriem glabājas faili. Katrs glabāšanas mezgls ir pieslēgts tā tuvākajam super mezglam;
 - klienta mezgli (*client nodes*), šiem mezgliem nav nekādu pienākumu. Tie ir tikai lai lietotu datus kas ir atrodami tīklā.

Kad tiek veikts pieprasījums no klienta datora, tas tiek pārsūtīts tuvākam super mezglam, kas šo pieprasījumu nodod tālāk mērķa super mezglam, kas savukārt to nosūta tuvākam glabāšanas mezglam. Glabāšanas mezgls pieprasīto failu tad pārsūta klientam (skatīt attēlu 4.1.).



4.1 att. **Wuala arhitektūras skice (pieprasījuma apstrādes shēma)**

5. **Godīgums** - tīkla efektīvu darbību nodrošina arī godīgi klienti. Taču rodas jautājums, kā novērst '*free riding*' klientus, kas daudz izmanto, bet maz dod, proti, neveic gandrīz nekādu ieguldījumu. *Wuala* izmanto formulu, pēc kuras novērtē klienta godīgumu, tādā veidā aprēķinot piešķiramo tiešsaistes glabātuves apjomu. Katru lietotāju raksturo šādi rādītāji:

- lokālais diska apjoms;
- klienta tiešsaistes laiks;
- augšupielādes tīkla caurlaidspēja (*upload bandwidth*).

Ņemot vērā minētos rādītājus, tiešsaistes glabātuves apjoms tiek aprēķināts šādi:

$$t_a = l_a * t_l$$

t_a = tiešsaistes glabātuves apjoms;

l_a = lokālais diska apjoms;

t_l = tiešsaistes laiks.

Godīgumu nodrošina arī princips - jo lielāku augšupielādes tīkla caurlaidspēju klients nodrošina, jo ātrāks būs tā ielādes ātrums. Tā kā mēdz būt arī asimetriska ieinteresētība - lietotājs pieprasa kaut ko no glabātuves mezgla, taču glabātuves mezglam tajā mirklī no lietotāja nekas nav vajadzīgs – princips "dots pret dotu" (*tit for tat*), kā tas tiek darīts programmatūrā *BitTorrent*, nevar tikt pielietots.

Šīs problēmas risināšanai *Wuala* izmanto dalīto slavas sistēmu (*distributed reputation system*) *Havelaar*, kura nav pakļauta nekādiem viltus ziņojumu un krāpšanas draudiem. Klientu datoru reputācijas novērošana programmatūrā *Wuala* notiek šādi:

1. Katrs dators veic novērojumus. Katru reizi, kad tas veic kādu transakciju, tas piefiksē, ar kādu tīkla caurlaidspēju tā tiek veikta, un rezultātus saglabā lokālā vektorā;
2. Katru noteiktu laiku, piemēram, nedēļu, klients pārsūta šo vektoru noteiktam pulkam kaimiņu datoru (tas tādēļ, lai kāds ļaunprātīgs lietotājs nevarētu noslogot visu tīklu, bet gan tikai tā daļu);
3. Saņemot vektorus, tiek izdzēsta visa informācija par sūtāmo klientu (lai izvairītos no safabricētiem datiem, tādā veidā lietotājam atstājot iespēju safabricēt datus tikai par draugiem);
4. Taču tā kā sistēma saņem ļoti daudz statistikas vektoru, tā aprēķina vidējos rādītājus, samazinot krāpšanas iespēju līdz minimumam;
5. Šādā veidā tiek veikti vairāki vektoru pārsūtīšanas pārgājieni;
6. Reputācija vai slava tiek atjaunota katram mezglam lokāli, un kad nonāk laiks līdz atlīdzināšanai tiek izsniegta tīkla caurlaidspēja ņemot vērā ieguldījuma proporcijas.

Galvenās šādas reputācijas sistēmas priekšrocības ir:

- akurāts lietotāju reālo ieguldījumu novērtējums;
 - stingrs nodrošinājums pret savtīgiem lietotājiem;
 - augsta efektivitāte, samērojams ar lielo transakciju skaitu.
6. **Noslodze** - *Wuala* programmatūrā, līdzīgi kā tas notiek *BitTorrent*, pārlietu lielas noslodzes gadījumā klienta pieprasījums tiek pārsūtīts tālāk. Proti, ja glabātuves mezgls uzglabā kādu ļoti populāru faila gabalu un pie tā regulāri pieslēdzas miljoniem klientu, kas pieprasa tieši šo failu, glabātuves mezgls pieprasījumus pārsūta tālāk klientam, kas jau ir ielādējis šo failu. Tādā veidā klienta mezgli var apmainīties ar datiem savā starpā.



-
7. **Uzturēšana** - lai pārlicinātos, vai fragmenti, kas bija pieejami, kad fails tika ievietots tīklā, būs pieejami arī pēc mēneša, tiek veiktas periodiskas pārbaudes. Tās nepieciešamas, lai uzzinātu, vai faila pieejamība ir tiešām pietiekama, kā arī, vai daži datori nav pametuši tīklu uz visiem laikiem. Ja šāds apstāklis tiek konstatēts, programmatūra rekonstruē trūkstošos fragmentus un ielādē tos tīklā.

Sīkāk izpētot un salīdzinot šo dalītās tiešsaistes datu glabātuves risinājumu ar jau ieviestiem un plašāk iztestētiem līdzīgiem produktiem, nākas secināt, ka daudzām izvirzītajām prasībām šis risinājums neatbilst. Ņemot vērā arī faktu, ka šis risinājums ir komerciāls, tā izmantošana un pielāgošana kļūst apgrūtināta.

5. CLEVERSAFE.

Šajā nodaļā tiek izklāstīta izklaidēto datu glabātuvju risinājuma *Cleversafe* detalizēta analīze.

Drošas un ilgstošas datu uzglabāšanas un rezerves kopiju veikšanas ziņā veiksmīgs risinājums ir *Cleversafe*. No angļu valodas tulkojumā "gudri drošs". *Cleversafe* sevi dēvē par vadošo izklaidēto glabātuvju tehnoloģiju ziņā. Kompānija tika dibināta 2004. gadā un, lai spētu pēc iespējas veiksmīgāk un ātrāk šo produktu nostandartizēt, ir ieviesusi un sponsorē arī atvērtā koda „cleversafe.org” projektu. Sākot ar 2008. gada februāri, *Cleversafe* ir kļuvis arī komerciāls, taču atvērtā koda projekts „cleversafe.org” vēl joprojām tiek turpināts.

Cleversafe mērķis ir datu uzglabāšanu padarīt vienkāršu, lētu un drošu. Tas tiek panākts, izmantojot izklaidēto glabātuvju tehnoloģiju, kas ir pierādīta esam daudz izdevīgāka par ierasto replicēšanu. Veicot replicēšanu un izveidojot tiešu faila dublikātu, lietotājs ir spiests uzturēt par 100% lielāku glabātuves apjomu katrai duplikācijai. Tas nozīmē lielākus izdevumus, sarežģītāku uzturēšanu un lielāku risku. Ja rodas kāda negaidīta katastrofa un gan serveris, gan atspoguļotā failu glabātuve iziet no ierindas, nepieciešamie faili vairs nav atgūstami.

Lai izbēgtu no šādas situācijas, lietotājs var izveidot vēl papildus citā vietā novietotu failu atspulgu serveri, kas vēl par 100% palielinās jau tā augstās izmaksas. Šāds risinājums izpalīdzēs, ja tiks bojāti tikai divi serveri, taču ja tiks bojāts arī trešais? Līdz ar to var secināt, ka šāda papildu servera uzturēšana šo problēmu tik vienkārši neatrisinās.

Izmantojot izklaidēto glabātuves tehnoloģiju, kura būtībā datus sadala un izplata starp serveriem, tos saliekot kopā tikai tad, kad fails tiek pieprasīts, izmaksas un riski ievērojami samazinās. Šāda veida faila uzglabāšanai ir nepieciešams tikai par 60% lielāks papildu diska apjoms. Ja kāds skaits serveru iziet no ierindas, lietotājs vēl joprojām var iegūt sev nepieciešamā faila pilnu un nebojātu kopiju, ja vien ir pieejams minimālais sadalītā faila gabalu skaits.

Cleversafe datu sadalīšanai izmanto *Cauchy-Reed-Solomon* informācijas izklaidētus algoritmus, kuri, izmantojot drošu Interneta pieslēgumu, tiek tālāk pārsūtīti vairākiem serveriem izklaidētajā glabātuves tīklā. Šis process ir vizuāli attēlots attēlā Nr. 5.1. Datu pārsūtīšana un uzglabāšana ir privāta un droša, to nodrošina šādi aspekti:

- 1) Neviena pilna faila kopija neatrodas uz vienas aparatūras;
- 2) Individuāli datu gabali nesatur pietiekami daudz informācijas, lai neautorizēts skatītājs spētu izprast oriģinālā faila saturu.

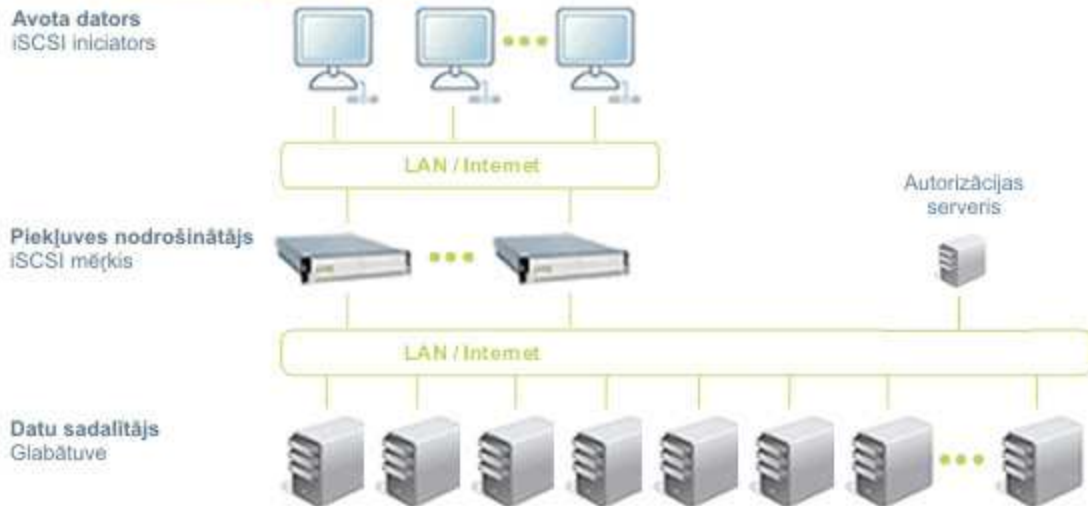


5.1.attēls **Datu uzglabāšanas un atgūšana Cleversafe tīklā.**

Šādas izkliedētās glabātuves sistēmas uzturēšanai un darbībai ir nepieciešami trīs pamatelementi jeb slāņi - Avota dators (*Source Computer*), Piekļuves nodrošinātājs (*Accesser*) un Datu sadalītājs (*Slicestor*), skatīt attēlu Nr. 5.2. Katram no šiem elementiem ir sava darbības specifika un mērķis:

- **Avota dators** - darbojas kā iSCSI iniciators. Tas tiek izmantots, lai, pieslēdzoties piekļuves nodrošinātājam, izmantojot iSCSI saskarni, iesniegtu, atgūtu vai izdzēstu lietotāja datus. Avota datori veic standarta FS operācijas (piemēram: kopēšanu, atjaunošanu un dzēšanu), lietotāji var izveidot, izmainīt un dzēst failus izmantojot jebkuru faila balstītu programmatūru. Diemžēl atvērtā koda risinājumā nav dokumentēta un testēta iespēja, kā vairākiem iSCSI iniciatoriem pieslēgties vairākiem iSCSI mērķiem.
- **Piekļuves nodrošinātājs** - ir iSCSI mērķis. Būtībā tas ir kā izkliedētās glabātuves maršrutētājs, kas noglabā, izgūst un sadala, digitālo informāciju lietotāja definētos gabalos izmantojot informācijas izkliedētos algoritmus (turpmāk tekstā – IIA, *IDA – Information Dispersal Algorithms*). Pašlaik *Cleversafe* ir implementēta optimizēta *Rīda-Solomona* algoritmu versija zināma kā *Cauchy-Reed-Solomon* algoritmi, taču *Cleversafe* programmatūra ir pielāgojama un atļauj izmantot arī citas jaunākas IIA implementācijas.
- **Datu sadalītājs** - sadalītie datu gabali tiek izdalīti datu sadalītājiem, kuri ir tīklā saistīti taču var būt arī ģeogrāfiski izkliedēti.

Izkliedētās glabātuves infrastruktūra



5.2 att. Izkliedētās glabātuves infrastruktūra

Atsevišķos *Cleversafe* fiziskās arhitektūras raksturojumos ir minēta vēl viena komponente - autorizācijas serveris, kurš tiek saslēgts vienā tīklā ar Datu sadalītājiem un Piekļuves nodrošinātājiem. Tas nozīmē, ka, iespējams, komerciālais produkts piedāvā arī sava veida autorizāciju, kura laikā tiek veikta piekļuve minētajam autorizācijas serverim.

Atšķirībā no atvērtā koda produkta, komerciālais produkts vēl papildus satur arī Pārvaldnieku (*Manager*). Šī iekārta pārbauda visu izkliedētā glabātuvju tīkla komponentu darbību un nelaimes gadījumos automātiski uzsāk trūkstošo gabalu pārbūvi. Tā atbalsta standarta SNMP protokolus.



5.3 att. Piekļuve izkliedētam glabātuves tīklam.

5.1. PROGRAMMATŪRAS DARBĪBA

Datu izkliedēšana un saglabāšana *Cleversafe* izkliedētajā glabātuves tīklā var tikt iedalīta četrās fāzēs:

1. Pirmssadalīšanas fāze - dati tiek saņemti kā atsevišķi datu avoti. Piekļuves nodrošinātājs katram datu avotam veic rindu ar kodēšanas darbībām. Šādos procesos iekļaujas kompresija, šifrēšana vai integritātes pārbaude (kā CRC kontrolsummas, digitālie paraksti u.c.), lai gan neviena no tām netiek obligāti pieprasīta;
2. Sadalīšanas fāze - katrs datu avots tiek padots IIA;
3. Pēcsadalīšanas fāze - jau vienreiz sadalīts, katrs gabals var tikt tālāk apstrādāts. Viens piemērs varētu būt integritātes (CRC, digitālā paraksta utt.) pārbaude, lai uzturēšanas process varētu pamanīt datu bojājumus;
4. Glabāšanas fāze - tiklīdz visas pirms izkliedēšanas darbības ir veiktas, fails tiek pārsūtīts datu sadalītājiem.

Cleversafe sistēma ir projektēta tādā veidā, lai tā varētu būt pieejama kā jebkura cita glabātuves sistēma, proti, relatīvi vienkārši jau no eksistējošiem lietojumiem.

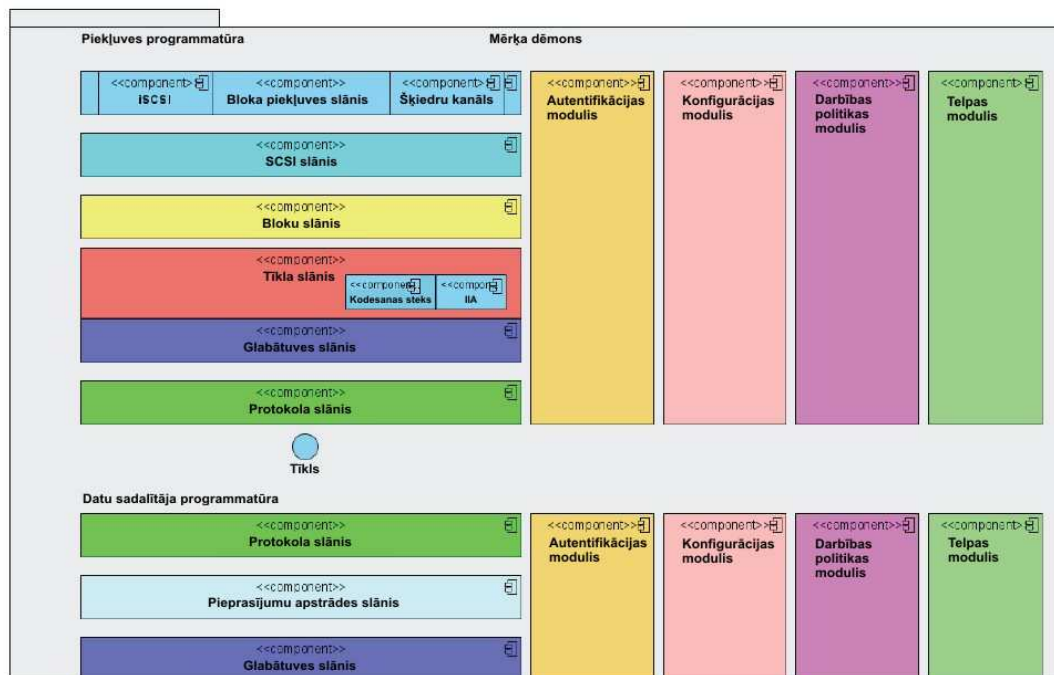
5.2. CLEVERSAFE PROGRAMMATŪRAS ARHITEKTŪRA

Cleversafe programmatūras arhitektūra ir izveidota, lai vairums tā rādītāju, tādu kā IIA (*IDA*), kā arī kodēšanas un dekodēšanas algoritmu būtu maināmi un konfigurējami. Sīkāk par maināmo rādītāju īpašībām un iespējām tiek izklāstīts *Cleversafe* uzstādīšanas nodaļā.

Cleversafe programmatūras arhitektūra, vadoties pēc tās darbības specifikas, var tikt iedalīta - piekļuves un pamata slāņos. Sīkāk par katru no programmatūras slāņiem un moduļiem (to darbībām un īpašībām) tiek izklāstīts sekojošās apakšnodaļās.

5.2.1 Piekļuves slāņi

Piekļuves slānis ir izklaidētā tīkla klienta puse, kas ir atbildīga par klienta pieprasījumu tulkošanu tīkla slānim. Tehniski šis slānis neiederas pamata arhitektūrā, jo nav noteiktas nekādas specifikācijas vai saskarnes ko tam ir jāievieš.



4.5 att. Programmatūras arhitektūra

Šajā izklaidētajā glabātuves tīkla infrastruktūrā lietotāja konta glabātuve tiek dalīta vienā vai vairākās telpās (*vault*). Telpa nav *Cleversafe* fiziskās arhitektūras sastāvdaļa, tas ir kā virtuāls diskdzīnis, kura dati tiek šifrēti, sagriezti un izklaidēti pa vienu vai vairākām fiziskām datu glabātuvēm. Šī telpa ir arī līmenis, kurā tiek noteikts šifrēšanas sliekšnis. Lietotājs ar šo telpu darbojas kā ar bloka iekārtu, līdzīgi cietajam diskam. Pašlaik *Cleversafe* ir ieviesta bloka³ līmeņa piekļuve, taču nākotnē *Cleversafe* plāno ieviest sīkāku līmeņu piekļuvi (piemēram, failu līmeņa, kur katrs adresējamais datu avots tiktu attēlots kā fails):

- **Bloka piekļuves slānis (*Block Device Access Layer*)** - Bloka līmeņa piekļuves slānis attēlo lietotāja konta telpu kā bloku, kas ir pieejams, izmantojot SCSI bāzētus protokolus kā iSCSI un šķiedru kanālu protokolu (*FCP*). Pamatuzdevums šim slānim ir ieviest šos standartus un pārsūtīt SCSI komandu aprakstošos blokus (*CDB - Command Descriptor Blocks*) SCSI līmenim. iSCSI iniciators tiek ielādēts uz klienta datora, kur tā lietotāja konta telpa tiek attēlota kā standarta bloka ierīce, piemēram,

³ Bloks - ierakstu kopums, ko apstrādā vai pārsūta kā vienotu veselumu.



uz *Windows* datora, tā var parādīties kā disks F: vai uz *Linux* /dev/sdb. Šīs ierīces sadalīšanai partīcijās vai formatēšanai var tikt izmantota jebkura standarta programmatūra. Lietotājs var formatēt disku izmantojot vēlamo failu sistēmu, kā arī veikt lasīšanas, rakstīšanas un dzēšanas darbības.

- **Bloku slānis (Block Layer)** - Bloku slānis atklāj telpu izklieģētājā tīklā kā saistītu bloku masīvu, katram blokam ir noteikts, iepriekš definēts izmērs. Primārā bloka slāņa atbildība ir izpildīt vairāku bloku lasīšanas un rakstīšanas darbības. Lai apmierinātu pieprasījumus no vairākiem avotu tipiem, ieskaitot SCSI slāņa vai īpašnieka ierīces draivera, tā saskarne tiek attēlota kā tipisks cietais disks. Atšķirībā no augstākiem slāņiem bloka slānim ir pieeja informācijai par izklieģēto tīklu un tā darbībām, taču noteiktas glabātuvju detaļas ir slēptas.
- **SCSI slānis (SCSI layer)** - SCSI slānis ir atbildīgs par SCSI CDB skaidrošanu un pārtulkošanu Bloku slāņa darbībās. Kad no Bloka slāņa tiek saņemta kāda atbilde vai arī kļūdas paziņojums, SCSI slānis to pārveido par SCSI CDB un pievieno paketei, kura tiek pārsūtīta bloka piekļuves slānim. Šim slānim nav nekādu zināšanu par piekļuves protokolu vai izklieģētā tīkla glabātuvju struktūru. Kā jau vairākums standarta SCSI tiešās glabātuvju piekļuves ierīces, SCSI slānis izpilda visus nepieciešamos SCSI bloka komandas -2 (*SBC2*) un SCSI primārās komandas - 3 (*SPC-3*) standartus. Dažos gadījumos komanda var tikt apstrādāta, tikai izmantojot SCSI līmeni, bez komunikācijas ar zemākajiem slāņiem.
- **iSCSI** - viena šāda izklieģēto tīklu saskarne ir iSCSI *Initiator-Target* (RFC3720). iSCSI protokols nodrošina mehānismu lietotāju telpas atklāšanai un datu pārsūtīšanai.
 - Atklāšanas sesija - Atklāšanas posma laikā mērķis nodrošina sarakstu ar konkrētā lietotāja autorizētām telpām.
 - Darba sesija - tiek apstrādāti iSCSI pieprasījumi, tādi kā lasīt un rakstīt, kā arī pamata SCSI komandu aprakstošie bloki tiek pārsūtīti SCSI slānim. Izklieģētā tīkla atbildes un kļūdas paziņojumi tiek pārsūtīti iniciatoram.

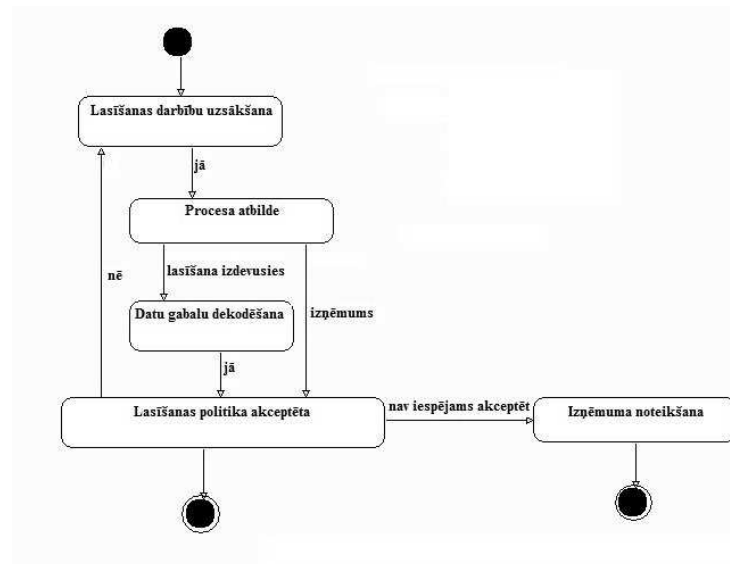
5.2.2. Pamata slāņi

- **Tīkla slānis (Grid Layer)** - izklieģētā glabātuves tīkla datu pārvaldības darbi tiek atstāti tīkla slāņa kompetencē. Tīkla slānis ir atbildīgs par daļīto transakcijas uzsākšanu, izpildi un kļūdas paziņojumu apstrādi vairākām glabātuvēm, kad tiek saņemts kāds pieprasījums. Pirms rakstīšanas darbības pabeigšanas tīkla slānis pārbauda, vai datu avots atrodas atbilstošā, atļaujamā stāvoklī, tādā veidā nodrošinot sava veida konkurences kontroli. Tīkla slāņa darbības iekļauj sevī rakstīšanu, lasīšanu, dzēšanu un sarakstu (*read, write, delete and list*). Tīkla slānis ir arī atbildīgs par datu kodēšanu - šifrēšanu un saspiešanu. Specifisko kodeku saraksts katrai telpai tiek iepriekš noteikts un uzstādīts telpas konfigurācijā. Pēc datu kodēšanas, tīkla slānis veic arī informācijas izklieģēšanu, veidojot vairākus datu gabalus, kas tiek saglabāti uz vairākām glabātuves ierīcēm. Datu rakstīšanas laikā tīkla slānis darbojas kā multipleksors, taču kā demultipleksors, kad dati tiek lasīti:
 - **Lasīšanas darbība** (sk. 5.6 att.) - Lasīšanas darbības laikā tīkla slānis cenšas nolasīt un savākt datu avotu no visām glabātuves iekārtām telpā. Ņemot vērā izklieģētās informācijas īpašības, dati var tikt arī savākti, ja dažas no glabātuves

iekārtām nedarbojas. Izklidētais tīkls atrodas pieļaujamā bojājumu stāvoklī (*acceptable outage state*), ja datu avotu ir iespējams veiksmīgi savākt.

Viena no tīkla slāņa optimizācijām paredz, ka datu lasīšanas laikā netiek izmantotas visas telpas glabātuves ierīces, bet gan izklidēto datu gabaliņu sliekšnis (minimālo glabātuvju ierīču skaits, kas nodrošina veiksmīgu datu atgūšanu).

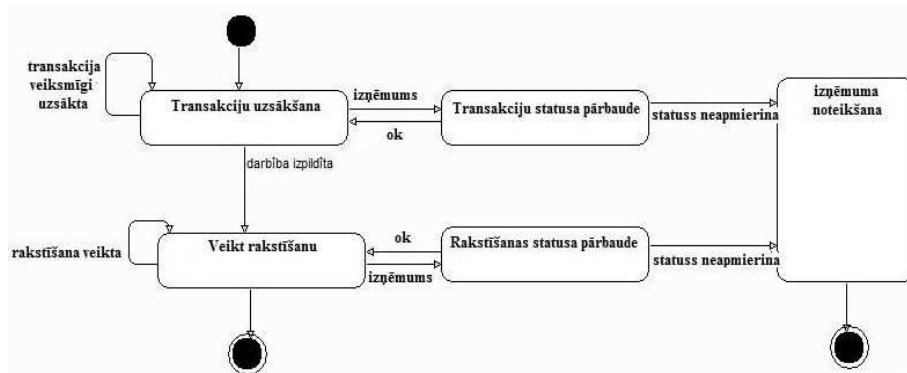
Tā kā visas rakstīšanas darbības tiek izpildītas transakcijā ar transakcijas identifikatoru, dažreiz ir iespējams arī atjaunot iepriekšējo datu versiju, pat tad, ja pašreizējā versija nevar tikt iegūta. Tādēļ tīkla slānim ir jāspēj noteikt pēdējā iegūstamā datu avota versija.



5.6 att. Lasīšanas stāvokļu diagramma

- **Rakstīšanas darbība** (sk. 5.7 att.) - Rakstīšanas darbības laikā tīklu slānis cenšas saglabāt datus vairākās glabātuvju ierīcēs, lai vēlāk dati varētu tikt veiksmīgi savākti. Rakstīšanas darbības tiek veiktas dalītā transakcijā un tīklu slānim ir jābūt pārliecinātam vai visas rakstīšanas darbības ir beigušās veiksmīgi, lai varētu transakciju veiksmīgi nobeigt (*commit*). Katrs datu avots tiek rakstīts ar savu transakcijas identifikatoru, kas tiek vēlāk izmantots izklidēto datu gabalu saistīšanai.

Tā kā datu lasīšanai nav nepieciešami visas telpas izklidētajā tīklā esošās glabātuves ierīces, tad arī rakstīšanas darbībām tās nav vajadzīgas. Taču pirms rakstīšanas darbības izpildes tīkla slānis pārbauda, vai izklidētais tīkls atrodas pieļaujamā bojājumu stāvoklī.



5.7 att. Rakstīšanas stāvokļu diagramma

- **Datu kodēšanas steks** - kodeki tiek izmantoti secīgi lai pārvērstu oriģinālos datus vai datu gabalus pirms to rakstīšanas izkliedētajā glabātuvju tīklā. Kodeku pielietojumu secībai telpā ir jābūt noteiktai, taču var tikt ieviesti arī jauni kodeki.
- **Glabātuves slānis (*Storage Layer*)** - šis slānis nošķir fizisko glabātuvu no izkliedētajiem datu gabaliem. Izmantojot glabātuves slāņa saskarni, Tīkla slānis var izpildīt visas darbības (lasīt, rakstīt, dzēst utt.), bez zināšanām par pamata glabātuves realizāciju. Šāda veida struktūra atvieglo arī testēšanu.
- **Protokola slānis (*Protocol layer*)** - šis slānis izpilda protokola ziņojumu kodēšanas un dekodēšanas darbības, izmantojot bināro protokolu. Visus pieprasījumus, atbildes un kļūdas ziņojumus kodē sūtītājs izmantojot abstraktās sintakses notāciju 1 (ASN.1) un dekodē saņēmējs.

5.2.3. Datu sadalītāja programmatūras slāņi (*Slicestor Software Layers*)

Sadalītāja konfigurācija satur:

- Autentifikācijas mehānismu;
- Versiju informāciju;
- Atbalstāmos telpu tipus;
- Atbalstāmos datu glabātuvju parametrus.

Sadalītājs ir mezgls izkliedētajā glabātuves tīklā. Datu sadalītāja arhitektūra ir stipri līdzīga piekļuves (*Accesser*) arhitektūrai, un daudzas aplikācijas tiek izmantotas kopīgi. Arī ielādes pakotne abiem ir viena.

- **Protokola slānis (*Protocol layer*)** - šis līmenis tiek koplietots ar klientu un veic tās pašas operācijas. Kad no klienta tiek saņemts bināra protokola pieprasījums datu sadalītājs to dekodē un iesniedz pieprasījumu Pieprasījumu apstrādes slānim. Līdzīgi notiek arī ar atbildes vai kļūdas paziņojumu apstrādi, Pieprasījumu apstrādes slānis tās kodē un pa tīklu pārsūta klientam.
- **Pieprasījumu apstrādes slānis (*Request handler layer*)** - tas ir vienkāršs mehānisms, kurš maršrutē pieprasījumus no protokola slāņa uz pamata apakšsistēmu (piemēram glabātuves slāni vai autentificēšanās moduli). Saņemtais

atbildes vai kļūdas paziņojumi tiek tālāk maršrutēti protokola slānim, kas tos pārsūta klientam.

- **Glabātuvju slānis (*Storage Layer*)** - glabātuvju slānis tiek koplietots starp datu sadalītāju un klienta pusi.

5.2.4. Moduļi

- **Konfigurācijas modulis (*Configuration module*)** - *Cleversafe* izklaidētās glabātuves tīkla programmatūra ir veidota, pievēršot plašu uzmanību tās iespējamai attīstībai un uzlabošanai. Tas nozīmē, ka pamata struktūra ir pazīstama tikai ar izklaidētā tīkla koncepta (tādu kā IDA, kodeki/transformācijas) saskarnēm un jebkuri nākotnē īstenotie jauninājumi var tikt ieviesti *Cleversafe* bez nepieciešamības mainīt pamata kodu. Šī īpašība paver ļoti plašas iespējas programmatūras attīstībai un ilglaicīgai izdzīvošanai.
- **Darbības politikas modulis (*Policy module*)** - darbības politika ir atsevišķi konfigurējams rādītājs. Darbības politika ir iespēja pielāgot esošās klases uzvedību izmantojot iepriekš definētas saskarnes. Piemēram, tīkla slānim ir noteikts rakstīšanas sliksnis (veiksmīgi pieejamo datu gabalu skaits, kurus izmantojot var tikt atgūts datu avota oriģinālais saturs), šo rādītāju var noteikt arī sistēmas administrators (sīkāka informācija lēlādes nodaļā).
- **Telpas modulis (*Vault module*)** - katrai telpai ir noteikti konfigurējamie parametri, tādi kā bloka izmērs u.c.

Ņemot vērā programmatūras atvērtā koda priekšrocības, tā darbības pielāgojamību un veiksmīgo arhitektūru, *Cleversafe* varētu būt ļoti labs kandidāts Latvijas Datu režģa implementācijai. Šī risinājuma īpašības atbilst lielākai daļai iepriekš definēto Datu režģa prasību. Balstoties uz pieejamo uzbūves aprakstu un veikto analīzi šis produkts nodrošina ātru, ērtu un drošu failu uzglabāšanu un izgūšanu, taču lai tas atbilstu visām nepieciešamībām, tas tomēr vēl ir jāuzlabo. Lai pārliecinātos par produkta atbilstību visām vai vairākiem iepriekš izvirzīto prasību, ir jāveic abu produktu (*Wuala* un *Cleversafe*) salīdzinājums.

6. IZANALIZĒTO PRODUKTU SALĪDZINĀJUMS

Šajā nodaļā tiek attēlots iepriekš izanalizēto izklaidēto datu glabātuvju risinājumu salīdzinājums atkarībā no to atbilstības iepriekš izvirzītajām Datu režģa prasībām. Skatīt tabulu Nr- 6.1.

6.1. tabula

Dalīto datu glabātuvju risinājumu salīdzinājums

Prasības	Wuala	Cleversafe
Sadalīta un satur replikas	+	+
Droši šifrēta	+	+
Plaši pieejama	-	+
Kontrolēta, droša datu koplietošana	+	+/-
Veiktspēja	-	+
Saskarne kā failu sistēmai	+	+
Atvērtā koda	-	+
CA atbalsts	-	+

- Sadalīta un satur replikas** - abi produkti datu drošai uzglabāšanai izmanto līdzīgu pieeju - informācijas izklaidētos algoritmus (Rīda-Solomona grupai piederošos). Tie tīklā saglabāto failu sadala, šifrē un izplata pa dažādām glabātuves iekārtām, tādā veidā piekļuve failam tiek nodrošināta 99% (ņemot vērā nelaimes gadījuma iespējas). Katrs atsevišķs faila gabals ir šifrēts un neatklāj oriģinālā faila saturu.
- Droši šifrēta** - abos gadījumos šifrēšana tiek veikta lokāli uz lietotāja datora, līdz ar to šifrēšanas atslēgas un citi rādītāji nevar tikt nelegāli atklāti.
- Plaši pieejama** - *Wuala* risinājuma gadījumā tīklā ievietotie faili ir pieejami tikai tad, ja lietotājs ir ielādējis uz savas aparatūras *Wuala* klienta programmatūru. Taču *Cleversafe* ir projektēta tā, lai būtu relatīvi vienkārši pieejama jau no eksistējošiem lietojumiem (skat. attēlu Nr. 5.3).
- Kontrolēta, droša datu koplietošana** - *Wuala* lietotājs, izmantojot *Cryptree*⁴, dalās ar savu draudzības atslēgu ar lietotājiem, kuriem atļauj piekļūt jau šifrētiem failiem. *Cleversafe* atvērtā koda gadījumā šāda datu koplietošana pēc noklusējuma nav paredzēta, taču, uzlabojot telpas piekļuves autorizāciju, šāda iespēja var tikt ieviesta. Ļoti iespējams, ka komerciālais *Cleversafe* produkts šādu iespēju jau ir realizējis.
- Veiktspēja** - *Wuala* risinājuma gadījumā, darbības notiek uz lietotāja datora uzstādītās *Wuala* klienta programmatūras, datu ielādes procesi stipri noslogo

⁴ *Cryptree* - kriptogrāfiska koka struktūra, kas atvieglo pieejas kontroli neuzticamu datu glabātuvju failu sistēmās.

procesoru, atmiņu un arī disku, tādēļ ielādes ātrums mēdz būt ļoti lēns. Taču *Cleversafe* gadījumā datu apstrāde notiek tīkla veiktspējas robežās.

6. **Saskarne kā failu sistēmai** - vizuāli *Wuala* lietotāja interfeiss līdzinās standarta failu sistēmai, taču vairāku apjomīgu datu ielāde ir sarežģīta un notiek manuāli - katru failu ielādējot atsevišķi. Turklāt *Cleversafe* gadījumā ir iespējams tīklam pieslēgt veselu cieto disku ar neierobežotu datu apjomu, kas uzreiz tiek pievienots un ir ērti pieejams.
7. **Atvērtā koda** - *Wuala* ir bezmaksas programmatūra taču ne atvērtā koda, savukārt, neskatoties uz to, ka *Cleversafe* ir uzsācis arī savas komerciālās darbības, atvērtā koda projekts „cleversafe.org” vēl joprojām tiek turpināts. Kā arī, ņemot vērā atvērtā koda licenci (*GNU General Public License 2.0 (GPL)*), lietotājam ir atļauja kopēt un izplatīt uzlabotu programmatūras versiju.
8. **CA atbalsts** – *Wuala* šādu iespēju neparedz, savukārt *Cleversafe* ir iespēja standarta infrastruktūrai pielāgot autorizācijas servera pievienošanu. Šāda veida serveris varētu atvieglot *Cleversafe* savietojamību ar skaitļošanas režģi.

Ņemot vērā produktu salīdzināšanas rezultātus, tika nolemts dzīvē ieviest atvērtā koda risinājumu *Cleversafe*, jo tas izrādījās daudz efektīvāks un atbilstošāks noteiktajām prasībām.

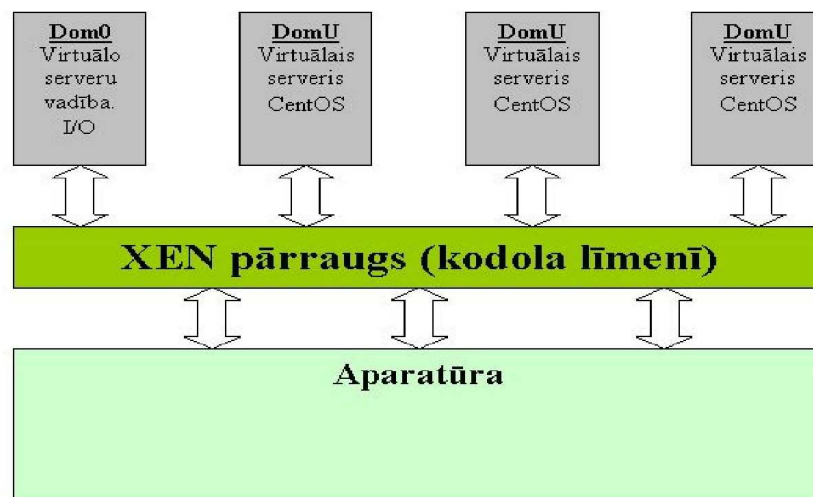
7. CLEVERSAFE UZSTĀDĪŠANA

Šajā nodaļā tiek secīgi aprakstīti *Cleversafe* uzstādīšanai un testēšanai nepieciešamie darbi. Tiek attēloti veltītā servera virtualizācijas darbi, kā arī katra virtualizētā servera uzstādīšana, ņemot vērā iedalīto lomu (piekļuves nodrošinātājs un datu sadalītāji).

7.1. Veltītā servera virtualizācija, izmantojot „XEN HYPERVISOR”

Atvērtā koda risinājuma infrastruktūras konfigurācija nosaka, ka katrs no iepriekš minētajiem līmeņiem (avota dators, piekļuves nodrošinātājs un datu sadalītāji) ir jāizvieto uz atsevišķa servera. Šāds obligātais nosacījums sadārdzina iecerēto programmatūras testēšanu. Taču risinājums būtu, izmantojot atklātā pirmkoda virtualizācijas sistēmu *Xen® hypervisor* (tulkojumā no angļu valodas *XEN* pārraugs) sadalīt serveri vairākās virtuālās mašīnās⁵, tādā veidā imitējot vairāku serveru izmantošanu.

XEN pārraugs ir infrastruktūras virtualizācijas risinājums, kas atbalsta ļoti plašu virkni ar viesu OS, ieskaitot *Windows*, *Linux*, *Solaris*, kā arī vairāku veidu *BSD OS*.



7.1 att. Servera virtualizācija, izmantojot XEN pārraugu

Izmantojot *XEN* virtualizāciju, starp servera aparatūru un OS tiek izvietots papildu programmatūras slānis, tā dēvētais *XEN* pārraugs. Tas nodrošina abstrakcijas slāni, kas atļauj katrā fiziskā serverī efektīvi strādāt vairākiem "virtuāliem serveriem" (skat. attēlu Nr. 7.1).

Minētajā attēlā ir redzami jauni termini (*Dom0* un *DomU*), tie ir domēni – izpildes konteksti, kas satur virtuālo mašīnu:

- *Dom0* (*Domain 0*) vai domēns nulle – pārvalda sistēmu. Tas ir pirmais *XEN* pārrauga domēns saknēšanā (*boot*). Tam ir īpašas privilēģijas, piemēram, jaunu domēnu startēšana, un tieša piekļuve aparatūrai.

⁵ Virtuālā mašīna – vide, kurā strādā mitināta operētājsistēma un kas piedāvā izdalīta datora vīziju.

- DomU – viesis. Dom0 kopija vai dublikāts, pēc noklusējuma neprivilģēts domēns bez piekļuves aparatūrai. DomU kodols nāk no Dom0 failu sistēmas, nevis no FS kura strādā DomU.

Lai gan var tikt izmantota jebkura OS, kas darbojas XEN kā DomU, taču tikai *Linux* satur rīkus un kodola ielāpus, kas nepieciešami Dom0 izpildei. Tādēļ arī šajā virtualizācijas gadījumā, Dom0 tiek izmantota *Ubuntu 7.10 server*.

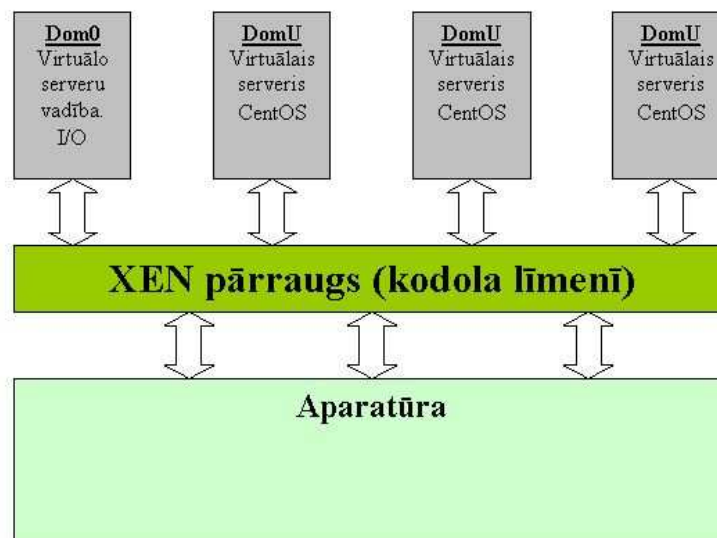
Pašlaik *Cleversafe* atvērtā koda risinājuma dokumentācijā ir minēts, ka programmatūra atbalsta tikai vienu OS - *CentOS 5.0*. Tādēļ uz visām virtuālām mašīnām (DomU) tika ielādēta minētā OS (taču testēšanas laikā atklājās, ka Datu Sadalītāji var izmantot arī citas *Linux OS*).

Servera virtualizācijai tika izmantota pēdējā stabilā XEN versija 3.2.0. Darbs sākās ar Dom0 izveidošanu, kas sastāv no šādiem soļiem:

- *Ubuntu 7.10* servera uzstādīšana,
- Repozitoriju aktivizēšana,
- *XEN* sagatavošana kompilācijai (kodola parametru maiņa),
- *XEN* kompilācija,
- *XEN* uzstādīšana, kodola izveide.

Pēc atsāknēšanas var sākt virtuālās OS uzstādīšanu. Tika izmantots papildu dators, uz kura tika ielādēta vajadzīga *CentOS OS*. Datora cietā diska satura saknes direktorija tika pārkopēta izveidojot vēlamo OS prototipu. Uz servera tika izveidoti 3 nodalījumi un papildu 3 maiņas nodalījumi. Katrā nodalījumā tika iekopēts OS prototips, ar kuru veiktas nepieciešamās izmaiņas (nosaukuma maiņa, tīkla konfigurācija, u.c.).

Servera uzbūve pēc virtualizācijas vizualizēta attēlā Nr. 7.2 . Tika izveidots Dom0 - *Ubuntu 7.10* un trīs DomU - *CentOS 5.2* .



7.2 att. Servera uzbūve pēc virtualizācijas

7.2. CLEVERSAFE UZSTĀDĪŠANAS PRASĪBAS.

Sadalīto virtuālo serveru lomas tika iedalītas šādi:

- virtuālais serveris - Piekļuves nodrošinātājs (*Accesser*);
- virtuālais serveris - Datu sadalītājs, glabātājs (*Slicestor*);
- virtuālais serveris - Datu sadalītājs, glabātājs (*Slicestor*).

Lai gan pašas programmatūras konfigurācija sevī ietver trīs lomas, to skaitā arī avota datoru, izvēlētas ārējo serveru uzstādīšanai ir tika divas (piekļuves nodrošinātājs un datu sadalītājs), kā avota dators var tikt izmantota jebkura lietotāja darba stacija, kas vēlas pieslēgties minētai izkliedēto datu glabātuvei. Tam nepieciešams tikai ielādēt iSCSI iniciatoru. Tā ielāde un uzstādīšana ir vienkārša un automatizēta gan *Windows*, gan *Linux* platformām. Zemāk ir attēlotas katrai lomai noteiktā nepieciešamā programmatūras platforma (sk. tabulas 7.1 – 7.3)

7.1 tabula

Klienta iekārtas platforma

Linux (ieteicams)	CentOS 5.0
	Open-iSCSI Initiator
Windows	Windows XP
	Microsoft iSCSI Software Initiator v2.05

7.2 tabula

Piekļuves nodrošinātāja platforma

Operētājsistēma	CentOS 5.0
Cita programmatūra	Java Runtime Environment (JRE 5.0)
	Linux-PAM

7.3 tabula

Datu sadalītāja platforma

Operētājsistēma	CentOS 5.0
Cita programmatūra	Java Runtime Environment (JRE 5.0)
	Linux-PAM

Pēc instalācijas programmatūra aizņem tikai 20MB, līdz ar to lielas brīvā apjoma prasībās šīs programmatūras testēšanai nav nepieciešamas (aptuveni 40MB).

7.2.1. Klienta iekārtas uzstādīšana

Klienta iekārtas uzstādīšanai nepieciešams iSCSI iniciators (*Open-iSCSI Initiator*), kurš tika ielādēts no interneta - <http://www.open-iscsi.org> mājas lapas. Iniciatora ielāde uz *Linux* platformas tika veikta izmantojot komandu - `yum install iscsi-initiator-utils`. Ielādes process attēlots 7.3 att.

```
[dana@centos1 ~]$ sudo yum install iscsi-initiator-utils
Loading "installonlyn" plugin
Setting up Install Process
Setting up repositories
Reading repository metadata in from local files
Parsing package install arguments
Resolving Dependencies
--> Populating transaction set with selected packages. Please wait.
--> Downloading header for iscsi-initiator-utils to pack into transaction set.
iscsi-initiator-utils-6.2.0.865-0.8.el5 11 kB 00:00
--> Package iscsi-initiator-utils.i386 0:6.2.0.865-0.8.el5 set to be updated
--> Running transaction check

Dependencies Resolved

=====
Package                Arch      Version           Repository        Size
=====
Installing:
iscsi-initiator-utils  i386     6.2.0.865-0.8.el5 base              510 k

Transaction Summary
=====
Install      1 Package(s)
Update      0 Package(s)
Remove      0 Package(s)

Total download size: 510 k
Is this ok [y/N]: y
Downloading Packages:
(1/1): iscsi-initiator-ut 100% |=====| 510 kB 00:00
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing: iscsi-initiator-utils      ##### [1/1]

Installed: iscsi-initiator-utils.i386 0:6.2.0.865-0.8.el5
Complete!
```

7.3 att. iSCSI iniciatora ielāde

7.2.2. Piekļuves nodrošinātāja uzstādīšana

Piekļuves nodrošinātāja (1. virtuālais serveris - centos2) uzstādīšana tika veikta secīgi pa soļiem:

- 1) **Uzstādīts Java atbalsts** - no interneta ielādēts un uz servera uzlikts, atspiests uzstādīšanas `.bin` fails. *Java* uzstādīšana tika uzsākta izmantojot komandu: `rpm -ivh jre-1_5_0_15-linux-i586-rpm`. Lai pabeigtu darbu, tika pievienoti arī nepieciešamie ceļi `/etc/profile.d/java-jre.sh` failā. Veiktās izmaiņas:

```
export JAVA_HOME=/usr/java/jre1.5.0_15
```

```
export PATH=$JAVA_HOME/bin:$PATH
```

- 2) **Ielādēta Cleversafe pakotne** - arī šajā gadījumā tika no interneta (www.cleversafe.org) ielādēts un uz servera pārvietots izklaidētās glabātuves tīkla uzstādīšanas .rpm fails. Pakotnes uzstādīšana tika veikta izmantojot komandu - `rpm -ivh dsgrid-0.7.8-0.i386.rpm`, skatīt 7.4 att.

```
[dana@centos2 ~]$ sudo rpm -ivh dsgrid-0.7.8-0.i386.rpm
Password:
Preparing...                               ##### [100%]
 1:dsgrid                                   ##### [100%]
```

7.4 att. Izklaidētā glabātuvju tīkla uzstādīšana

Gan piekļuves nodrošinātāja, gan datu sadalītāja ielādei tiek izmantots viens un tas pats uzstādīšanas `dsgrid-0.7.8-0.i386.rpm` fails (tas skaidrojams ar vairāku programmatūras slāņu koplietošanu). Taču lai serveris zinātu kāda loma viņam jāpilda ir nepieciešamas papildus izmaiņas attiecīgajos konfigurācijas failos. Piekļuves nodrošinātājs izmanto `/etc/dsgrid/dsgrid-iscsi.conf` konfigurācijas failu.

- 3) **Veiktas nepieciešamās izmaiņas piekļuves nodrošinātāja konfigurācijas failā** (`/etc/dsgrid/dsgrid-iscsi.conf`). Kā arī ievadīts tīkla lietotāja lietotājvārds un parole. Veiktās izmaiņas:
- `bind_portal_host = 0.0.0.0`
 - `bind_portal_port = 3260`
 - `adv_portal_host = 127.0.0.1`
 - `adv_portal_port = 3260`
 - `account_path = /etc/dsgrid/accounts`
 - `vault_path = /etc/dsgrid/vaults`
- 4) **Pievienots lietotājvārds lietotāju grupai - *dsgridusers***, izmaiņas veiktas iekš `/etc/group: dsgridusers:x:503:dana`
- 5) **Izveidoti piekļuves nodrošinātāja lietotāja akreditācijas dati**, izmantojot komandu - `dsgrid-create-account --username=... --password=****` (skat. attēlu 7.5).

```
[root@centos2 ~]# dsgrid-create-account --username=cleversafe --password=cleversafe
Created new account.
Grid Account Username:   cleversafe
Grid Account Identifier: 11e19c6d-432e-3dbd-8291-cb9b75b98953
```

7.5 att. Lietotāja akreditācijas datu izveide

- 6) **Telpas konfigurācijas faila izveidošana** - lai izveidotu telpu ir nepieciešams izveidot speciālu telpai veltītu konfigurācijas failu. Par paraugu var ņemt noklusēto telpas konfigurācijas failu, kas atrodas `/usr/share/dsgrid/conf/vaults/` zem nosaukuma `sample-vault-descriptor.xml`. Vadoties pēc sniegtā parauga tiek izveidots jauns telpas konfigurācijas fails - `sigmanet.xml`. Šis fails nosaka vairākus telpas parametrus, kas ietekmē tā īpašības (piemēram, telpas izmēru, datu sadalītāju skaitu un to IP adreses, nepieciešamo gabalu skaitu veiksmīgai datu atgūšanai, u.c.). Attēlā Nr. 7.6 tiek attēlots konfigurācijas faila satura izraksts.

```
<!-- Total number of slices to be used to disperse data. -->
<eval:param name="num-slices">
<eval:literal type="int" value="8"/>
</eval:param>

<!-- Number of slices that must exist in order for data to be successfully
retrieved. -->
<eval:param name="threshold">
<eval:literal type="int" value="6"/>
</eval:param>
```

7.6 att. Telpas konfigurācijas maināmo parametru piemērs

Ja tiek pievienots vēl kāds papildu datu sadalītājs, kurā mēs vēlamies saglabāt šīs telpas sadalītos datu gabalus, tad jaunā servera informācija (IP adrese un ports) ir jānorāda šajā failā (skat. 7.7 att.). Pēc noklusējuma *Cleversafe* ielādes pakotnē katrai telpai ir norādīti 8 datu sadalītāji (līdz ar to fails tiek sadalīts 8 gabalos, no kuriem tikai 6 ir nepieciešami veiksmīgai faila satura atgūšanai), taču testa versijā, tiek izmantoti tikai divi.

```
<!-- Slice store type used. -->
<vault:slice-store kind="SliceStore" referral="remote">
  <eval:param name="connection">
  <eval:reference interface="org.cleversafe.layer.communication.Connector">

  <!-- Host name for Slice Server -->
  <eval:param name="host">
  <eval:literal type="string" value="127.0.0.1"/>
  </eval:param>

  <!-- Port number for Slice Server -->
  <eval:param name="port">
  <eval:literal type="int" value="5001"/>
  </eval:param>
  <eval:param name="manager">
  <eval:reference interface="org.cleversafe.layer.communication.ConnectorManager" />
  </eval:param>
  </eval:reference>
  </eval:param>
  <eval:param name="vault-identifier">
  <eval:method object="acl" name="getVaultIdentifier"/>
  </eval:param>
  <eval:param name="credentials">
  <eval:reference object="credentials"/>
  </eval:param>
</vault:slice-store>
```

7.7 att. Telpas datu sadalītāja informācija

- 7) **Telpas izveide** - telpas izveide veicama izmantojot komandu - `dsgrid-create-vault --username=data --password=storage --descriptor=sigmanet.xml`, skatīt attēlu Nr. 7.8.

```
Created new vault.
New vault identifier:      8ab9bffe-9d68-42fd-9330-682383e7d900
Grid account of vault owner: 11e19c6d-432e-3dbd-8291-cb9b75b98953
```

7.8 att. Jaunas telpas izveide

- 8) **Piekļuves nodrošinātājā startēšana** izmantojot komandu - `etc/init.d/dsgrid-iscsi start` (skat. attēlu 7.9).
- 9)

```
[root@centos2 ~]# /etc/init.d/dsgrid-iscsi status
Dispersed Storage Grid iSCSI Target is running (7303).
```

7.9 att. Piekļuves nodrošinātāja startēšana

7.2.3. Datu sadalītāju uzstādīšana

Cleversafe uzstādīšanas prasības, datu drošības apsvērumu dēļ, ietver sevī minimālo datu sadalītāju skaitu, kas ir divi. Arī mūsu gadījuma tika izveidoti un uzstādīti divi datu sadalītāji. Šo abu serveru uzstādīšana tika veikta secīgi pa soļiem:

1) **Uzstādīts Java atbalsts**, skatīt att. Nr.7.10.:

```
[root@centos3 ~]# rpm -ivh /home/dana/jre-1_5_0_15-linux-i586.rpm
Preparing...                               ##### [100%]
 1:jre                                       ##### [100%]
```

7.10 att. Java uzstādīšana

Tiek pievienoti papildus ceļi /etc/bashrc:

```
export JAVA_HOME=/usr/java/jre1.5.0_15
export PATH=$JAVA_HOME/bin:$PATH
```

2) **Ielādēta un uzstādīta Cleversafe pakotne** - dsgrid-0.7.8-0.i386.rpm. Šinī gadījumā Datu sadalītājs izmanto divus konfigurācijas failus:

- /etc/dsgrid/dsgrid-server.conf - ir vienkāršs teksta fails uz administratoru orientētām konfigurācijām;
- /etc/dsgrid/slice-server.xml - XML fails priekš zemāka līmeņa konfigurācijām.

3) **Veiktas izmaiņas datu sadalītāja konfigurācijas failā** (/etc/dsgrid/dsgrid-server.conf). Veiktās izmaiņas:

- host = 0.0.0.0
- port = 5000
- jaas_profile = pam-authenticate
- slice_data_path = /var/dsgrid

Tādā veidā norādot, datu sadalītāja IP adresi, iSCSI pieprasījumu klausīšanās portu, autentifikācijas rādītājus u.c.

4) **Lietotāju izveidošana** - dsgrid-create-account --username=data --password=storage

```
Created new account.
Grid Account Username: data
Grid Account Identifier: 8d777f38-5d3d-3ec8-815d-20f7496026dc
```

7.11 att. Lietotāja izveidošana

5) **Pievienots lietotājs lietotāju grupai dsgridusers** (/etc/group)

Datu sadalītāja sāknēšana, izmantojot komandu - /etc/init.d/dsgrid-server start, pēc kā datu sadalītājs veiksmīgi iestartējās.



```
[root@centos3 ~]# /etc/init.d/dsgrid-server start
Starting Dispersed Storage Grid Slice Server...
[root@centos1 ~]# /etc/init.d/dsgrid-server status
Dispersed Storage Grid Slice Server is running (8030).
```

7.12 att. Datu sadalītāja uzsākšana

7.3.CLEVERSAFE IZMANTOŠANA UN TESTĒŠANA

Lai pēc iespējas pilnīgāk pārbaudītu *Cleversafe* darbību, tā priekšrocības un trūkumus, tika izvēlēti vairāki testēšanas scenāriji, kuru laikā *Cleversafe* sistēmā tika augšupielādēti un lejupielādēti dati, reģistrējot augšupielādēšanas un lejupielādēšanas ātrumus, kā arī piekļuves nodrošinātāja noslodzi. Klienta un datu sadalītāju programmatūras radītā noslodze tika novērota zemāka par 5% no sistēmas resursiem, tādēļ šajā testēšanas posmā tam netika pievērsta papildu uzmanība. Lai arī minimālās *Cleversafe* prasības ir 4 datori (1 klients, 1 piekļuves nodrošinātājs un 2 datu sadalītāji), reālā vidē datu sadalītāju skaits būs ievērojami lielāks. Izstrādātāju un esošo lietotāju ieteiktā tipiskā konfigurācija ir 1 klienta platforma, 1 piekļuves nodrošinātājs un 8 datu sadalītāji, no kuriem vismaz 6 jābūt pieejamiem, lai klients spētu piekļūt datiem, t.i., augšupielādējot datus, katrs fails tiek sadalīts 8 gabalos. Lai atjaunotu pilnu failu, klientam nepieciešams iegūt jebkurus 6.

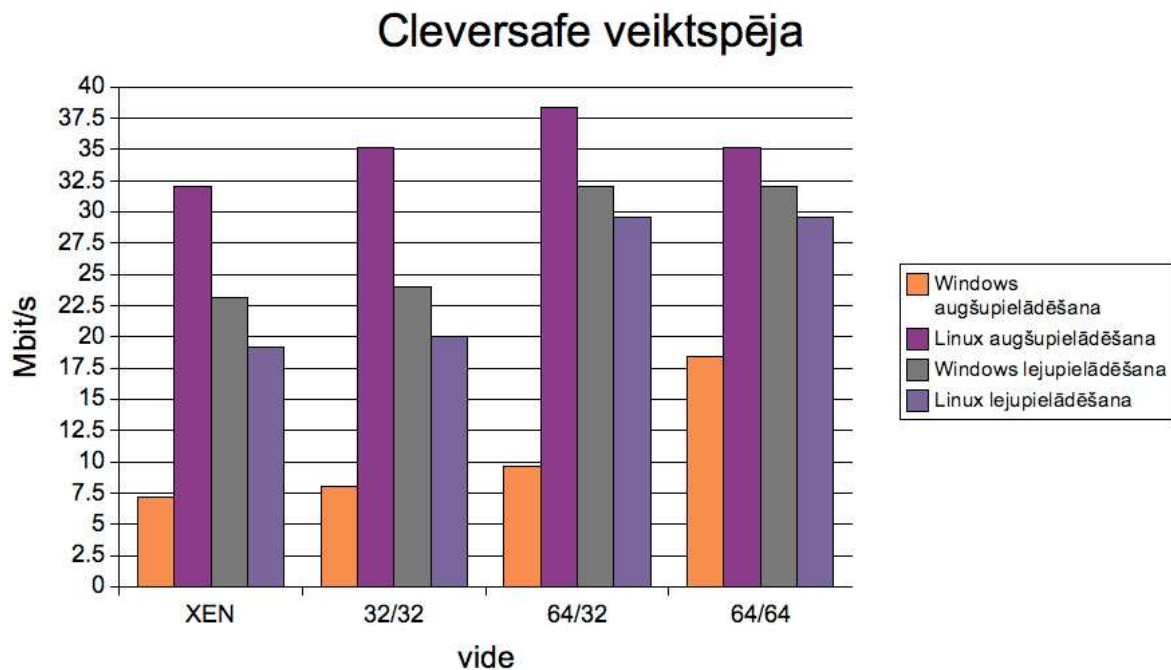
Testēšanas scenāriji:

1. Līdzīgi kā sākotnējā izmēģinājuma gadījumā, arī šajā tika izmantotas virtualizācijas tehnoloģijas piedāvātās iespējas. Uz viena fiziska datora tika uzstādīts XEN pārraugs ar 8 virtuālajiem serveriem, kur katrs no tiem pildīja datu sadalītāja funkciju. Datu piekļuves nodrošinātājam tika atvēlēts dators neizmantojot virtualizāciju. Lai pilnīgāk pārbaudītu *Cleversafe* darbību, tika izvēti divi dažādi klienti – viens ar *Microsoft Windows XP* operētājsistēmu un iSCSI Initiator programmatūru, otrs *Ubuntu Linux* ar Open iSCSI programmatūru. Gan piekļuves nodrošinātājs, gan datu sadalītāji tika darbināti, izmantojot 32-bit operētājsistēmas.
2. Lai vēl vairāk testa vidi pietuvinātu reālajai, šajā gadījumā datu sadalītāji katrs tika uzstādīts uz atsevišķa servera, neizmantojot virtualizāciju. 8 atsevišķi datori datu sadalītājiem, 1 dators datu piekļuves nodrošinātājam un 2 dažādi iSCSI klienti. Gan piekļuves nodrošinātājs, gan datu sadalītāji tika darbināti, izmantojot 32-bit operētājsistēmas.
3. Tā kā datu piekļuves nodrošinātājs ir ļoti svarīga *Cleversafe* sistēmas komponente un tās ātrdarbība tiešā veidā ietekmē visas sistēmas veiktspēju, tika nolemts, ka jāpārbauda, cik liela nozīme ir tam, vai tiek izmantota 32-bit vai 64-bit operētājsistēma. Datu sadalītāji tika izmantoti tie paši, kas iepriekšējā gadījumā. 8 atsevišķi datori datu sadalītājiem, 1 dators datu piekļuves nodrošinātājam un 2 dažādi iSCSI klienti. Piekļuves nodrošinātājs darbojās izmantojot 64-bit operētājsistēmu, savukārt datu sadalītāji tika darbināti izmantojot 32-bit operētājsistēmas.
4. Lai noskaidrotu datu sadalītāju operētājsistēmas ietekmi uz kopīgo sistēmas veiktspēju, gan piekļuves nodrošinātājs, gan datu sadalītāji tika darbināti, izmantojot 64-bit operētājsistēmas. Kā datu sadalītāji tika izmantoti 8 atsevišķi datori, 1 dators datu piekļuves nodrošinātājam un 2 dažādi iSCSI klienti.

Rezultāti:

Cleversafe veikspēja				
	Windows augšupielādēšana	Linux augšupielādēšana	Windows lejupielādēšana	Linux lejupielādēšana
XEN	7.2	32	23.2	19.2
32/32	8	35.2	24	20
64/32	9.6	38.4	32	29.6
64/64	18.4	35.2	32	29.6

7.4 tabula Clevesafe veikspēja (Mbit/s)



7.13 att. Cleversafe veikspēja

No iegūtajiem rezultātiem var secināt, ka ir ievērojama atšķirība starp *iSCSI Windows* un *Linux* klientu realizāciju – *Linux* klients visos gadījumos ir ātrāks pie datu augšupielādēšanas, savukārt pie datu lejupielādēšanas *Windows* klients darbojas ātrāk. Var secināt arī to, ka 64-bit operētājsistēmas (un *Java* atbalsta) izmantošana dod ievērojamus (teju vai divas reizes) uzlabojumus *Windows* augšupielādēšanas gadījumā un mērenus abu klientu lejupielādēšanas gadījumos. Datu piekļuves nodrošinātāja noslodze visos gadījumos bija robežās no 50% - 70% (piekļuves nodrošinātāja procesa radītā slodze datora procesoram). Tā ir samērā ievērojama slodze un ieviešot *Cleversafe* reālā vidē visticamākais, ka būs jāmeklē risinājumi piekļuves nodrošinātāja slodzes balansēšanai, piemēram, izmantojot vairākus piekļuves nodrošinātājus. Vēl jāņem vērā, ka piekļuves nodrošinātājs gan datu augšupielādēšanas brīdī, gan lejupielādēšanas laikā vienlaicīgi datu saņem (vai sūta) no klienta un sūta (vai saņem) datu sadalītājiem. Tādējādi, lai piekļuves nodrošinātājs spētu 40 Mb/s augšupielādes (vai lejupielādes) ātrumu klientam, tam pašam jānodrošina vismaz 80 Mb/s datu plūsmas caurlaidība. Ieviešot *Cleversafe* reālā darba vidē, jāņem vērā šis apstāklis un vismaz datu piekļuves serveris jānodrošina ar jaudīgu tīkla aprīkojumu (vismaz 1Gb/s pieslēgumu).

8. SECINĀJUMI UN DATU REŽĢA TURPMĀKĀ ATTĪSTĪBA

Cleversafe tehnoloģija ir ar lielu nākotnes potenciālu un plašām pielietojšanas iespējām, tomēr, plānojot to ieviest, ir jāņem vērā prasības pret aparatūru – gan serveru jaudas, gan tīkla aprīkojums un pat klienta datora aprīkojums var ievērojami ietekmēt *Cleversafe* izmantošanu. Pozitīvi ir tas, ka *Cleversafe* datu nogādāšanai pie klienta izmanto iSCSI protokolu, kas nodrošina piekļūšanu datiem, izmantojot *TCP/IP* tīkla protokolus, un sniedz iespēju strādāt ar datiem tā, it kā klienta datorā fiziski atrastos papildu cietais disks. Tieši pateicoties iSCSI protokolam, *Cleversafe* izmantošanas iespējas ir ļoti plašas – sākot ar parastu datu glabāšanas vietu kā papildinājumu datorā esošajiem resursiem (piemēram, rezerves kopijām, datu arhivēšanai) un beidzot ar iespēju uzstādīt operētājsistēmu uz iSCSI diska un darbināt to, izmantojot iSCSI sniegtās iespējas.