



KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Kompiuterių tinklų katedra

Donatas Sandonavičius

GRID teikiamų servisų efektyvumo analizė

Magistro darbas

Darbo vadovas:
doc. dr. Gytis Vilutis

Kaunas, 2007



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ TINKLŲ KATEDRA

TVIRTINU
Katedros vedėjas
prof. dr. R. Plėštys

GRID TEIKIAMŲ SERVISŲ
EFEKTYVUMO ANALIZĖ

Informatikos magistro baigiamasis darbas

Recenzentas

2007 05

Vadovas
doc. dr. G. Vilutis
2007 05

Atliko
IFM -1 gr. stud.
D. Sandonavičius
2007 05

KAUNAS, 2007

SUMMARY

Nowadays GRID computing issues become urgent. GRID quality of service (QoS) importance grows together with the development of GRID technology. The importance of QoS of the grid lies in at least three aspects: the concept of grid service, the multiplicity of users' demands, and the heterogeneity of grid resources (Pu Juhua, Xiong Zhang and Wu Zhenxing. The Research On QoS For Grid Computing.). Besides discussing GRID QoS parameters, there are also suggestions about the methods to obtain them.

The scope of this document is to research the GRID QoS aspects and the description of developed methodology for estimating QoS efficiency evaluating network and clusters parameters. The results of efficiency experiments done using the developed methodology are presented there.

Turinys

1.	ĮVADAS.....	7
2.	ANALIZĖ	10
2.1.	Siūlymas vertinti panašių sistemų QoS.....	12
2.2.	QoS tyrinėjimai GRID skaičiavimuose.....	18
2.3.	QoS problemos GRID skaičiavimuose	19
3.	GRID PROGRAMINĖ ĮRANGA.....	21
3.1.	GRID programinės įrangos palyginimas.....	22
4.	KURIAMOS METODIKOS BEI PROGRAMINĖS ĮRANGOS ANALIZĖ.....	26
4.1.	Projekto iniciatoriai.....	26
4.2.	Projekto Apribojimai.....	27
4.3.	Funkciniai reikalavimai.....	28
4.4.	Nefunkciniai reikalavimai.....	33
4.5.	Projekto išeiga.....	34
5.	SERVISŲ KOKYBĖS TYRIMAS	36
5.1.	Pasiekiamumas.....	36
5.2.	Prieinamumas.....	37
5.3.	Operatyvios parengties koeficientas	39
5.4.	Saugumas	40
5.5.	Vėlinimas	49
5.6.	Vėlinimo fliuktuacijos.....	51
5.7.	Pralaidumas	52
5.8.	Laisvi CPU.....	53
6.	EFEKTYVUMO ĮVERTINIMAS KLASTERIUI GRID KONTEKSTE	54
7.	BalticGrid VEIKIMO PRINCIPAS	56
8.	ATLIEKAMŲ TYRIMŲ REZULTATAI	57
	IŠVADOS	61
	LITERATŪRA.....	62

Lentelių sąrašas

<i>1 lentelė.</i> Interetinių paslaugų palyginimas.....	10
<i>2 lentelė.</i> GRID skaičiavimų resursų parametrai	14
<i>3 lentelė.</i> Resursų paskirstymo problemos sprendimas*.....	17
<i>4 lentelė.</i> UNICORE, Globus, Legion ir Gridbus palyginimas.....	22
<i>5 lentelė.</i> Veiklos įvykių sąrašas	30
<i>6 lentelė.</i> Skaičiavimo elementų palyginimas pagal darbinių mazgų skaičių.....	58
<i>7 lentelė.</i> Duomenų elementų palyginimas pagal saugyklai skirtos vietos dydį.....	58
<i>8 lentelė.</i> Skaičiavimo elementų apskaičiuotų pasiekiamumo, prieinamumo, vėlinimo, vėlinimo fliktuacijos ir pralaidumo parametrų lentelė.....	59
<i>9 lentelė.</i> Duomenų elementų apskaičiuotų pasiekiamumo, prieinamumo, vėlinimo, vėlinimo, vėlinimo fliktuacijos ir pralaidumo parametrų lentelė.....	59
<i>10 lentelė.</i> Klasterių QoS skaičiavimas pagal kelis užsiduotus parametrus.....	60
<i>11 lentelė.</i> Programos apskaičiuotų pasiekiamumo parametro lentelė	66
<i>12 lentelė.</i> Skaičiavimo elementų apskaičiuoto prieinamumo parametro lentelė	66
<i>13 lentelė.</i> Duomenų elementų apskaičiuoto prieinamumo parametro lentelė	67
<i>14 lentelė.</i> Klasterių apskaičiuoto vėlinimo parametro lentelė	67
<i>15 lentelė.</i> Klasterių apskaičiuoto vėlinimo fliktuacijų parametro lentelė.....	68
<i>16 lentelė.</i> Skaičiavimo elementų pralaidumo parametro lentelė	68
<i>17 lentelė.</i> Duomenų elementų pralaidumo parametro lentelė.....	69
<i>18 lentelė.</i> Laisvi CPU	69

Paveikslėlių sąrašas

1 pav. GRID Topologijos imitacija.....	8
2 pav. Imitacinio tinklo darbo rezultatų pavyzdys.....	14
3 pav. UNICORE, Globus, Legion ir Gridbus palyginimas	22
4 pav. Konteksto diagrama.....	29
5 pav. Panaudos atvejų diagrama.....	30
6 pav. Paslaugos pasiekiamumo dinamika	36
7 pav. Siūlomas pasiekiamumo skaičiavimo būdas	37
8 pav. Siūlomas prieinamumo skaičiavimo būdas	38
9 pav. Operatyvios parengties koeficiento vaizdinis modelis.....	39
10 pav. Saugumo įvertinimui sukurtos programos skaičiavimo pavyzdys	48
11 pav. Saugumo įvertinimui sukurtos programos rezultato pavyzdys	48
12 pav. Siūlomas paketinio vėlinimo skaičiavimo būdas	50
13 pav. Siūlomas uždavinio vėlinimo skaičiavimo būdas	50
14 pav. Paketinio vėlinimo fliktuacijos išsibarstymas apie vėlinimo vidurkį laike, skaičiuojant 1 val. intervalu	52
15 pav. Paketinio vėlinimo fliktuacijos išsibarstymas apie vėlinimo vidurkį laike, skaičiuojant 1 s. intervalu.....	52
16 pav. BalticGRID middleware veikimo principas.....	56
17 pav. BalticGrid tinklo topologija.....	57

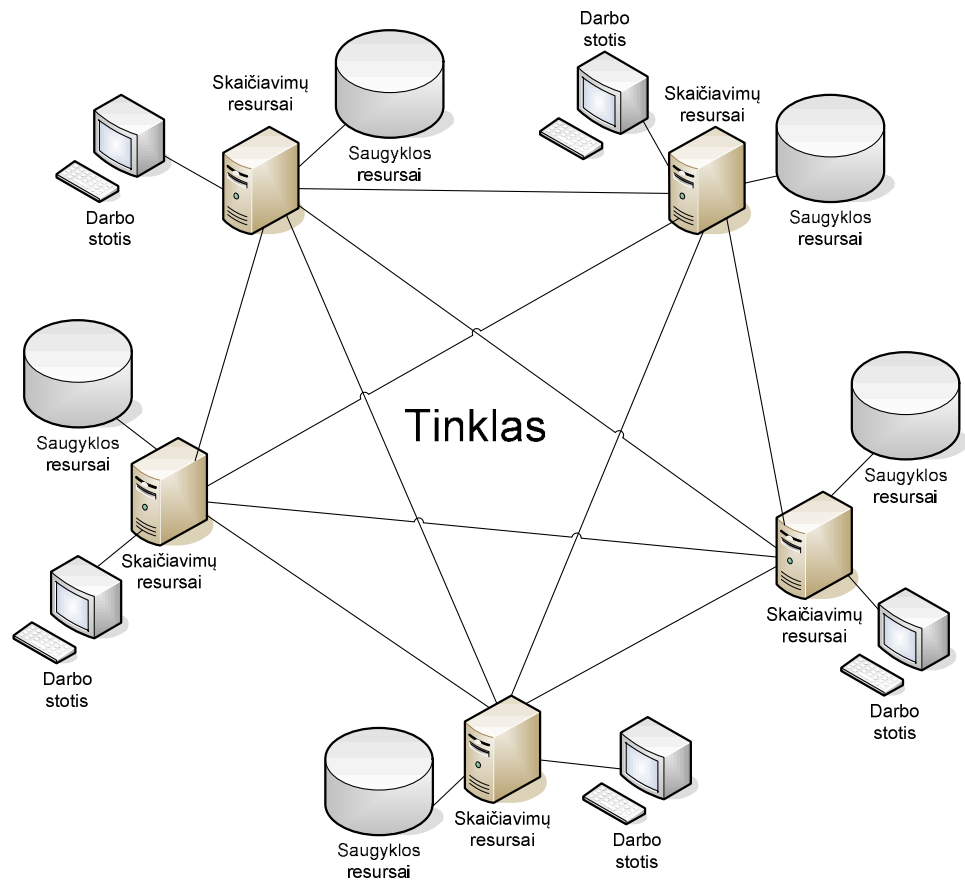
1. ĮVADAS

Sparčiai vystantis technologijoms, išskyla vis didesnis poreikis didelius skaičiavimus atliekančioms mašinoms – superkompiuteriams. Tuo tikslu yra kuriami GRID tinklai. Mokymo procesui kuriamos virtualios organizacijos ir virtualūs universitetai sėkmingai įgyvendinami GRID tinklų pagrindu.

GRID skaičiavimai – tai technologinis modelis, apjungiantis skaičiavimo resursus. GRID apjungia skirtingus kompiuterius ir jų sistemas (klasterius) į vieną didelę sistemą. Apjungimas gali būti atliekamas viename padalinyje (*Department Grids*), vienoje įstaigoje (*Enterprise Grids*) ar tarp kelių įstaigų arba net visame globaliame tinkle (*Global Grids*). Tokia sistema gali būti naudojama spręsti uždavinius, kurių negali apdoroti atskiri kompiuteriai ar jų sistemos. Taip pat GRID teikia priėjimą prie didžiulių duomenų kiekių, esančių paskirstytose duomenų bazėse ar failų sistemose bei leidžia jas matyti kaip vientisą duomenų bazę ar failų sistemą. Be to, GRID sudaro galimybę lengviau (paprasciau) bendradarbiauti ir pasidalinti vertingus resursus tarp organizacijų ar jų padalinių. Taip kaip interneto vartotojas mato vientisą turinio vaizdą per pasaulinį žiniatinklį, taip ir GRID vartotojas mato vieną didžiulį, galingą virtualų kompiuterį [30]. GRID tinklai visiškai nepriklauso nuo tinklo topologijos išsidėstymo.

GRID tinkle resursų ir vartotojų gali būti labai daug (1 pav.), todėl resursų savininkams yra labai sunku spręsti, kaip paskirstyti resursus galimiems vartotojams. Siekiant to išvengti gali būti kuriamos VO (virtualios organizacijos). Kiekvienas vartotojas priskiriamas tam tikrai VO, kurios viduje VO administratoriai priskiria tam tikras galimybes ar resursus vartotojams. Resursai tam tikrai VO suteikia papildomas galimybes.

GRID aplinką paprastai kuria programinis sluoksnis įdiegtas kompiuteriniuose, duomenų saugojimo ar kituose resursuose. Šis programinis sluoksnis angliškai vadinamas „*middleware*“. Kitaip tariant tai yra tam tikra operacinė sistema padedanti apjungti ir dalinti resursus. Šiuo metu nėra daug GRID „*middleware*“ operacinių sistemų. Pagrindiniai yra Legion, UNICORE, Globus Toolkit. Labiausiai populiarėja Globus Toolkit dėl savo plataus pritaikomumo ir dėl to, kad jis yra atviro kodo [31].



1 pav. GRID Topologijos imitacija

Kadangi GRID tinklų skaičius auga bei plečiasi virtualios GRID organizacijos, vartotojas turi vis didesnę pasirinkimą, kuriuo klasteriu ar GRID tinklu geriau pasinaudoti sprendžiant vieną ar kitą uždavinį. Didėjant uždavinių apimčiai ir greito rezultatų gavimo poreikiui, vis aktualesnis vartotojui tampa GRID teikiamų servisų efektyvumas (*QoS*). Siekiant įvertinti, kuris GRID klasteris, ar pats GRID tinklas veikia efektyviau, reikia apskaičiuoti klasterio ar GRID teikiamų servisų kokybės efektyvumą.

Šiame darbe tyrinėjami GRID servisų kokybės parametrai, taip pat kuriama metodika šioms parametrų apskaičiuoti bei metodika, kaip apskaičiuoti GRID teikiamų servisų kokybės efektyvumą.

Tyrimo objektas – GRID tinklo ar jo klasterio teikiamų servisų kokybės efektyvumo tyrimas.

Darbo tikslas – sukurti metodiką, kurios pagalba būtų įmanoma analizuoti GRID servisų kokybę (*QoS*) bei atlikti tiek *QoS* atskirų parametru skaičiavimus, tiek leistų apskaičiuoti GRID servisų kokybės parametrus ir įvertinti efektyvumą.

Darbo uždaviniai:

- Išanalizuoti esamus GRID QoS skaičiavimų būdus, ar bent siūlymus tai skaičiuoti;
- Sukurti metodiką GRID QoS efektyvumui skaičiuoti įtraukiant į skaičiavimus tinklo bei serverių parametrus;
- Atlikti bandomuosius skaičiavimus pagal sukurtą metodologiją.

Darbo metu sukaupta medžiaga buvo aprobuota dviejose mokslinėse konferencijose IT 2006 ir IT 2007. Atspausdintas straipsnis konferencinėje medžiagoje IT 2006 bei priimtas kitas straipsnis į tarptautinę konferenciją ITI 2007 Kroatijoje.

Ši medžiaga taip pat buvo panaudota ESF projekte (Virtualios mokslo ir studijų organizacijos kūrimas, vystymas ir diegimas GRID technologijų pagrindu. Nr. BPD-182/F450-BPD-182) pravedant mokymus pirmai tislinei grupei.

2. ANALIZĖ.

GRID tinklas – tai vėliausiai sukurta ir labiausiai išbaigta, išplėtota sistema, savo struktūra ir keliamais tikslais panaši į paskirstytus skaičiavimus, *web*, *peer-to-peer* skaičiavimus ir virtualizavimo technologijas [30]. GRID tinklo palyginimas su kitomis panašaus tipo internetinėmis sistemomis pateikiamas 1 lentelėje.

1 lentelė. Interetinių paslaugų palyginimas

	Panašumai	Skirtumai
<i>WEB</i> technologija	GRID tinklai paslepia nuo vartotojo sudėtingumus; vienos virtualios organizacijos VO vartotojai mato vientisą sistemą	WEB leidžia tik komunikaciją, o GRID palaiko pilną bendradarbiavimą
<i>peer-to-peer</i> technologija	GRID leidžia vartotojams keistis failais	GRID leidžia keistis ne tik failais, bet ir kitais resursais
Panašumai su klasteriais ir paskirstytais skaičiavimais	GRID jungia skaičiavimo resursus į vieną sistemą	Klasterių sudedamosios dalys privalo turėti fizinius ir veikimo panašumus, o gridai gali būti geografiškai išskaidyti ir įvairiarūšiai

GRID tipai

Kadangi GRID architektūra vis dar vystymosi stadijoje, yra daug galimybių modifikuoti GRID tipus. GRID tipai dažniausiai pagrįsti kažkieno poreikiais ir supratimu [29].

Kolkas nėra aiškių GRID kategorizavimo standartų. Daug tyrinėtojų analitikų, IT prekeivių ir kompiuterių mokslininkų klasifikuoja GRID ir jo variacijas, paremtas jų pačių supratimu ir vizijomis. Vieni klasifikavimo kriterijumi parenka funkcionalumą, kiti remiasi architektūra, tretieji klasifikuoja pagal įterptus komponentus. Galime pastebėti, kad dauguma organizacijų turi skirtingus požiūrius, pagal kuriuos gaunamos skirtingos klasifikacijos. GRID duomenų apdorojimas gali būti atliekamas įvairiais būdais, kad išspręstų įvairių rūšių užduotis. Dažnai GRID kategorizuojami pagal užduočių sprendimo tipus, kuriuos jie geriausiai atitinka.

Taigi nėra griežtų ir tvirtų taisyklių ar ribų tarp GRID tipų, ir dažnai GRID gali būti dviejų ar daugiau tipų junginys.

Šiuo metu yra trys dominuojantys GRID tinklų tipai:

- Skaičiavimo GRID tinklas – tai tinklas, kuris apjungia skaičiavimo resursus sudėtingų užduočių sprendimui;

- Duomenų GRID tinklas – tai tinklas apjungiantis skaičiavimo resursus stambių paskirstytų duomenų bazių analizei.
- Bendradarbiavimo GRID tinklas – tai tinklas naudojamas didelių modelių apsikeitimui tarp įstaigos darbuotojų, atliekančių vieną projektą būdami skirtingose vietose.

GRID tinklai taikomi tam, kad [30]:

- geriau išnaudotų esamus skaičiavimo resursus, tuo pačiu sumažinant administravimo išlaidas, susijusias su paskirstytų, įvairialypių informacinių sistemų valdymu;
- žymiai greičiau apdorotų didelius užduočių kiekius;
- suteiktų saugų priėjimą prie duomenų dideliame paskirstytame tinkle;
- lengviau bendradarbiautų su vidinėmis ir išorinėmis organizacijomis;
- sumažintų kompiuterinių sistemų, talpyklų, tinklo valdymo sudėtingumą ir kainas.

Šiuo metu GRID tinklai naudojami:

- aeronautikoje ir automobilių gamyboje (patogesniai projektavimui ir modeliavimui);
- energetikoje (naftos ir dujų telkinių aptikimui);
- biologijoje ir antropologijoje (ypatingai vaistų kūrime);
- nuosekliuose skaičiavimuose (daugiau skaičiavimų atliekama per trumpesnę laiką);
- duomenų vizualizacijoje;
- serverių/talpyklų grupavimui (išnaudojamumo padidinimui) ir daugelyje kitų sričių.

Atsižvelgiant į GRID taikymą, kiekviena GRID sistema privalo išlaikyti 3 esminius parametrus:

1) koordinuoti resursus, kurie nevaldomi centralizuotai;

GRID apjungia ir koordinuoja resursus ir vartotojus kurie valdomi skirtingai.

(A Grid integrates and coordinates resources and users that live within different control domains - for example, the user's desktop vs. central computing; different administrative units of the same company; or different companies; and addresses the issues of security, policy, payment, membership, and so forth that arise in these settings. Otherwise, we are dealing with a local management system.)

2) naudoti standartinius, atvirus, bendros paskirties protokolus ir vartotojo sąsajas;

GRID paremtas universaliomis sąsajomis ir protokolais, teikiančiais autentifikavimo, teisių suteikimo, resursų suradimo ir priėjimo prie resursų paslaugas.

3) teikti apibrėžtą servisų kokybę;

GRID leidžia naudotis bendrais resursais užtikrinant skirtingą servisų kokybę (*QoS*), vertinant tokiais parametrais kaip reakcijos laikas, pralaidumas, prieinamumas, saugumas ir skirtingų resursų užtikrinimas skirtingiems vartotojams.

Apibendrinant galima išskirti tris GRID tinklų privalumus:

1. Optimizuota infrastruktūra

GRID tinklų modelis leidžia optimaliai išnaudoti kompiuterių sistemas, paskirstydamas darbų apkrovą ir suteikdamas reikalingą resursų kiekį, kai tik to pareikalaujama. Sumažinus apdorojimo laikus, pagreitinus rezultatų pateikimo laiką ir padidinus rezultatų tikslumą padidėja bendras produktyvumas. Be to, GRID leidžia rikiuoti užduotis į eilę bei išspręsti problemas, kurios seniau buvo neišsprendžiamos.

2. Didesnis priėjimas prie duomenų ir laisvesnis bendradarbiavimas

Bendradarbiavimas apjungia informacinių technologijų resursus ir žmones. Išskirstyti padaliniai (įmonės, universiteto) ar organizacijos gali kurti virtualias organizacijas, kurios galėtų tarpusavyje keistis resursais. Suteikiant vartotojams priėjimą prie skaičiavimo, duomenų bei saugojimo resursų, kai jiems to reikia, ir leidžiant jiems efektyviai bendradarbiauti, gali pastebimai padidinti produktyvumą.

3. Lanksti, plačiai prieinama infrastruktūra

Resursų ir duomenų srautų prieinamumas 24 valandas 7 kartus per savaitę yra daugelio universitetų reikalavimas. GRID leidžia kurti lanksčias veikiančias sistemas, teikiančias resursus, kai tik to prireikia. Be to, GRID turi galimybę reaguoti į gedimus ir atstatyti vieną ar kelis jo komponentus, nesukeliant rūpesčių vartotojams.

2.1. Siūlymas vertinti panašių sistemų QoS

Plataus masto GRID'ai yra labai sudėtingos sistemos, susidedančios iš labai daug ir skirtingų tinklo komponentų, priklausančių skirtingiems IPT. QoS užtikrinimas yra pakankamai sunkus uždavinys tokiose sistemose. GRID servisų kokybė svarbesnė negu įprastiniuose WEB servisuose, kadangi GRID „*middleware*“ programinį sluoksnį sudaro keletas sudedamųjų dalių, tokių kaip vartotojo sąsaja, autorizacija, resursų paskirstymas, duomenų persiuntimo sistema, rezultatų pateikimas. Kiekviena dalis turi skirtingus

vėlinimus ir skirtingus apdorojimo laikus, priklausančius nuo GRID taikymo srities. Įprastiniuose tinkluose kokybės tyrimas atliekamas taškas į tašką (klientas-tiekėjas). GRID tinkluose tiekėjų gali būti labai daug. Todėl paslaugos kokybės tyrimas turėtų būti atliekamas iš kliento pusės peržiūrint visus kliento virtualios organizacijos resursus. Uždavinį apsunkina tai, kad vartojai ir resursai gali bet kada keistis savo apimtimi.

Kadangi GRID aplikacijos turi daugybę pritaikymų, tokių kaip video konferencijos, nutolusi vizualizacija, realaus laiko paveikslėlių analizė, nutolusios techninės įrangos analizė, kliento-serverio persiuntimai, duomenų bazių replikavimas, didelių duomenų rinkinių skirstymas, todėl siūloma tirti visą spektrą parametrų. GRID tinkluose paslaugų kokybę galima tirti pagal tuos pačius parametrus, kokie naudojami kitiems WEB servisams tirti. Tačiau tie parametrai gali būti taikomi skirtingiems lygmenims, pavyzdžiui, aplikacijų ir fiziniame lygmenyse tiriant skirtingus aspektus.

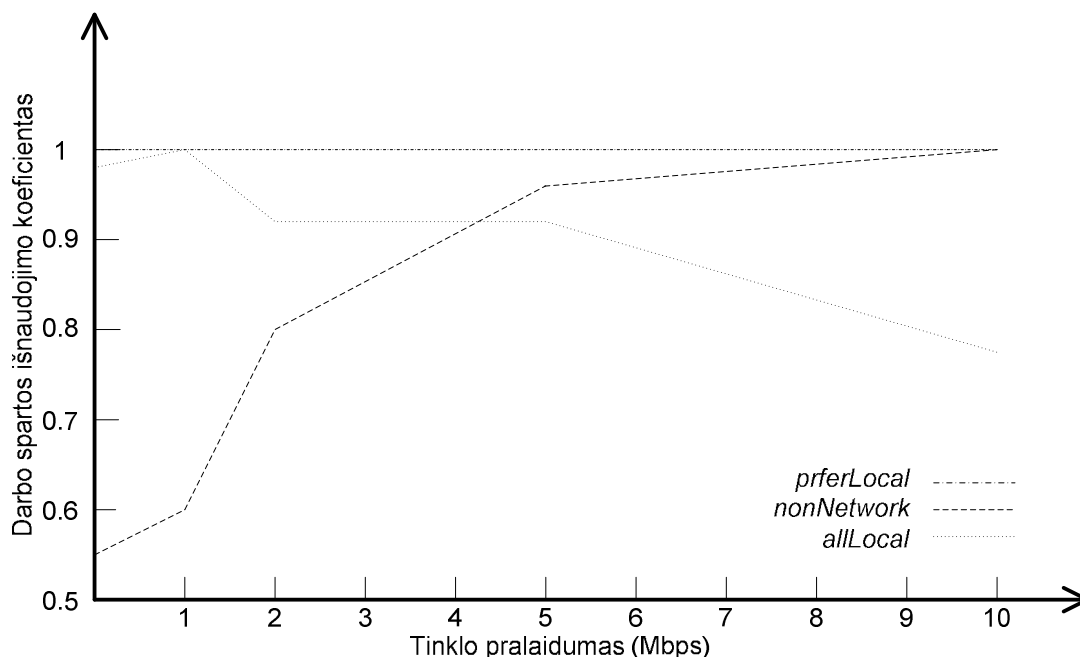
Šiuo metu GRID tinkluose nėra įgyvendinto QoS mechanizmo ir jie veikia „geriausių pastangų“ (*best effort*) principu, t.y. visiems srautams naudojami vienodi prioritetai. Vienas iš praktinių sprendimų būtų pasiūlyti tokius QoS mechanizmus kurie leistų paslaugų tiekėjams suskirstyti paslaugas pagal prioritetą ir ekonominę naudą. Taip yra padaryta GRID *Quality of Service Management (G-QoS)* projekte. Jame siūloma servisų kokybę užtikrinti sesijos pradžioje, vartotojui prisijungus prie sistemos, o vartotojui atsijungus paslaugos valdymo mechanizmas nutraukiamas, ir resursai atlaisvinami kitiems vartotojams. Tas užtikrinama dinaminio resursų planuotojo pagalba, kuris parenka optimaliausius maršrutus pagal tai, koku tikslu naudojamas GRID.

Servisų kokybę siūloma vertinti dviem aspektais – kokybiniu ir kiekybiniu. Kokybinis požiūris – tai paslaugos patikimumas, saugumas ir vartotojų pasitenkinimas, o kiekybinis kriterijus taikomas tinklams, CPU (procesoriams), duomenų saugykloms yra vėlinimas, paketų praradimas, pralaidumas ir t.t. Taip pat juos galima grupuoti į tokius punktus, kaip nuo kliento priklausantys, nuo tinklo, nuo protokolo ir nuo serverio. Nuo kliento priklauso tokie punktai, kaip „GET“ užklauso išsiuntimo bei priėmimo sparta. Skirtingoms naršyklėms ir operacinėms sistemoms yra skirtingi užklauso apdorojimo laikai. Tinklo parametrams priklauso pasiekiamumas, vėlinimas, paketų praradimas. Nuo protokolo priklauso pralaidumas. Pagrindinis jo punktas – tai kadro dydžio parinkimas, klaidų korekcija. Serverių žinioje yra „GET“ užklauso apdorojimas ir rezultatų pateikimas.

GRID tinklo topologijos pavyzdys:

1 paveiksle pavaizduota GRID topologijos imitacija susidedanti iš 5 kompiuterių, kurių kiekvienas iš jų turi skaičiavimų ir saugyklai skirtų resursų. Paskirstytas procesų

darbų vykdymas galimas dėl skirtingų GRID sistemų sujungtų per tinklą. Galimas tinklo pralaidumas riboja darbų vykdymo dažnį.



2 pav. Imitacinio tinklo darbo rezultatų pavyzdys

2 paveiklėse pavaizduoti imitacijos rezultatai parodo GRID efekto svarbą, pavyzdžiui, kai darbo resursai viršija lokalaus kompiuterio darbo galimybes: algoritmas, kuris vykdo visus darbus lokaliai („AllLocal“) buvo palygintas su dviem algoritmais, sugebančiais planuoti darbus paskirstytoje sistemoje. Aišku, kad esant mažam tinklo pralaidumui, tinklo būseną apskaičiuojantis algoritmas („PreferLocal“) grąžina geresnius darbų rezultatus nei algoritmas („NoNetwork“), kuris nesigilina į tinklo būseną jungdamasis prie įvairių resursų, reikalingų darbams atlikti.

Resursų paskirstymas

SLA yra kontraktas tarp paslaugos tiekėjo ir vartotojo, apibrėžiant priimtina QoS nustatytų matų lygį [34]. Tai paprastai sieja kainą su norimu serviso paslaugos lygiu. Svarbus ir sudėtingas dalykas GRID aplinkoje yra tai, kaip stebėti ir palaikyti SLA sutartyje nustatytus matus, kuomet daug vartotojų dalinasi tais pačiais resursais, o ypač, todėl, kad svarbiausia dalis GRID aplinkoje apibrėžia tai, jog ji teikia netrivialią QoS.

2 lentelė. GRID skaičiavimų resursų parametrai

Resursas	Greitis (milijonai ciklų per sekundę)	Kaina (Litai per sekundę)
1	1 000	0.10
2	2 000	0.25
3	3 000	0.60

Aplikacijų lygio SLA matai turėtų būti pažymėti resursų lygio SLA matuose. Nustatinėjant SLA su žemesnio lygio resursais, GRID globalių resursų planuotojai turi suprasti, kad skirtingų resursų tiekėjai gali siūlyti servisus skirtingo lygio su skirtinga kaina. Taigi, optimizavimo problema GRID aplinkoje yra servisų ir servisų tiekėjų parinkimas, kad kryptingai pasiekti globalinį SLA su mažiausia kaina.

Panagrinėsime pavyzdį, tarkime, kad RAM nustatymas reikalauja NC milijono procesoriaus ciklų ir turi būti pabaigtas per daugiausiai T_{\max} laiko vienetų, ir kad skaičiavimų kaina neviršytų C_{\max} litų. Tarkime, kad daugiausiai trys skaičiavimo resursai gali būti paskirstyti analizei. Pažymime s_i , ($i = 1, 2, 3$) greičiu resurso i milijone ciklu per sekundę ir pažymime c_i , ($i = 1, 2, 3$) kaina litais per sekundę naudojant resursą i . Mes išskaidysime nustatymus į tris lygiagrečius nepriklausomus uždavinius. Taigi, trys skaičiavimų resursai turi septynis galimus paskirstymo variantus: (1), (2), (3), (1, 2), (1, 3), (2, 3) ir (1, 2, 3).

Tegul T ir C bus vykdymo laikas ir kaina atitinkamai duoto paskirstytumo. Kai kurie paskirstymai gali būti neįmanomi, kad tenkintų globalų SLA maksimalų laiką T_{\max} ar maksimalią kainą C_{\max} . Jei tinkamiausias resursas i yra panaudotas, tada

$$T = \frac{NC}{s_i} \leq T_{\max}, i = 1, 2, 3, \quad (1)$$

$$C = c_i \times \frac{NC}{s_i} \leq C_{\max}, i = 1, 2, 3. \quad (2)$$

(1) lygybė rodo, kad visas resurso i vykdymo laikas negali išnaudoti globalaus SLA T_{\max} laiko. (2) lygybė yra kainos apribojimas. Laiko ir kainos apribojimai susidaro, kai du resursai i ir j yra naudojami. Tuomet

$$T = \max \left\{ \frac{NC_i}{s_i}, \frac{NC_j}{s_j} \right\} \leq T_{\max}, \quad (3)$$

$$C = c_i \times \frac{NC_i}{s_i} + c_j \times \frac{NC_j}{s_j} \leq C_r, \quad (4)$$

$$NC_i + NC_j = NC. \quad (5)$$

Jei visi trys resursai yra panaudoti, tada

$$T = \max \left\{ \frac{NC_1}{s_1}, \frac{NC_2}{s_2}, \frac{NC_3}{s_3} \right\} \leq T_{\max}, \quad (6)$$

$$C = \sum_{i=1}^3 c_i \times \frac{NC_i}{s_i} \leq C_{\max}, \quad (7)$$

$$\sum_{i=1}^3 NC_i = NC. \quad (8)$$

Jei atmestume kainos apribojimą, sprendinys, kuris minimizuoja RAM vykdymo laiką, yra vienintelis, kuris pasiskirsto proporcingai resursų greičiui. Taigi,

$$N_i = NC \times \frac{s_i}{\sum_{j=1}^3 s_j}. \quad (9)$$

Pažvelkime į bendrą atvejį, kuriame mes imame maksimalų vykdymo laiką ir maksimalią kainą. Viena iš optimizavimo problemų yra išspręsti GRID resursų paskirstymo mechanizmą, tai yra, rasti galimą paskirstymą, kuris patenkintų veikimą su apribojimais, t.y. $T \leq T_{\max}$ su minimalia kaina. Dviguba problema yra rasti galimą paskirstymą, kuris minimizuotų vykdymo laiką T ir tenkintų kainos apribojimą $C \leq C_{\max}$.

Panagrinėkime kitą skaitmeninį pavyzdį:

$NC = 10^7$ milijonų ciklų, $T_{\max} = 4800$ s, $C_{\max} = 1500$ Lt. 2 lentelėje duotas greitis ir kainos trims skaičiavimų resursams.

Kiekvienam galimam resursui paskirstyti mes sprendžiame optimizavimo problemą, ieškodami ciklų paskirstymo, kuris minimizuotų vykdymo laiką T , kol tenkiname kainos apribojimą $C \leq C_{\max}$. 3 lentelė parodo sprendinius: negalimas paskirstymas žymimas pasviru šriftu, galimas paskirstymas žymimas paryškintu šriftu. Paskutinė eilutė atitinka optimalų paskirstymą, kuris vykdo RAM per 2 593 sekundes ir kainuoja 1.352Lt (3 lentelė).

3 lentelė. Resursų paskirstymo problemos sprendimas*

Eil. Nr.	Paskirstymas	Ciklo paskirstymas (NC_i) milijone ciklų			Vykdymo laikas (s.)	Kaina (Lt.)
		1	2	3		
		1	(1)	10 000 000		
2	(2)	-	10 000 000	-	5 000	1 250
3	(3)	-	-	10 000 000	3 333	2 000
4	(1, 2)	3 333 333	6 666 667	-	3 333	1 167
5	(1, 3)	4 800 000	-	5 2000 000	4 800	1 520
6	(2, 3)	-	6 666 667	3 333 333	3 333	1 500
7	(1, 2, 3)	2 592 593	5 185 185	2 222 222	2 593	1 352

* paryškintas paskirstymas yra įvykdomas, pasviręs neįvykdomas.

Pirmas iš dviejų paskirstymų yra neįvykdomas todėl, kad pažeidžia vykdymo laiko apribojimą (maksimalus 4 800 s). Trečias ir penktas paskirstymai yra neįvykdomi dėl kainos apribojimo pažeidimo (maksimali kaina 1 500 Lt). Keliuose paskirstymuose (tokiuose kaip paskutinis), greitesnis ir brangesnis mazgas – 3 mazgas šiuo atveju – yra naudojamas tik dalyje vykdymo laiko, dėl kainos apribojimo. Faktiškai 2 222 222 milijonų ciklų yra paskirstyti į jį ir jo greitis 3000 milijonų ciklų per sekundę, šis mazgas yra naudojamas 741 sekundę, ar 28.6 procentų vykdymo laiko.

Pastaba

Jei mes turime n resursų, ir jei skaičiavimas gali būti suplanuotas į bet kokių skaičių resursų nuo 1 iki n , galimų paskirstymų visas skaičius:

$$\sum_{k=1}^n \binom{n}{k} = 2^n - 1. \quad (10)$$

Taigi, tai galėtų būti brangu analizuoti visus galimus paskirstymus ir spręsti optimizavimo problemas kiekvienam atskirai. Šiuo atveju GRID resursų sąrašas gali naudoti euristiką, kad pasiektų artimą optimaliam sprendimui labiau efektyviu būdu.

Tikra aplinka yra labiau komplikauta nei pavyzdys, kurį apžvelgėme: skaičiavimai gali turėti struktūrą, kuri atspindi abipusę priklausomybę tarp užduočių ir komunikacijų užlaikymą duomenų persiuntimui bei aukštesnio lygio koordinavimą. Vis

dėlto, aptartas pavyzdys akcentuoja du svarbius klausimus, kurie išskyla kai planuojami skaičiavimai GRID mazguose – būtent, SLA sutikimas ir kainos apribojimas.

Patikimumo skaičiavimas

Tarkime, pateikti keli serveriai, kuriuose yra išskaidyta informacija, reikalinga užduoties sprendimui. Informacija serveriuose nesidubliuoja. Taigi tikimybė, kad uždaviny bus išspręstas:

$$P = \sum_{i=1}^k (P_i \cdot \eta_i); \quad (11)$$

čia η_i informacijos kiekis i -tajame serveryje ($0 \leq \eta_i \leq 1$), P_i - tikimybė, kad i -tasis serveris bus pasiekiamas.

Jei uždaviniams skirta informacija serveriuose dubliuojasi, tai patikimumo formule atrodys taip:

$$P = \sum_{i=1}^I \eta_i \cdot \left[1 - \prod_{k=1}^K P_k \right]; \quad (12)$$

čia P_k tikimybė, kad informacija k -me serveryje bus nepasiekama, η_i - informacija i -tajame serveryje patalpinta, I – visi serveriai, kuriuose patalpinta informacija, K – visi serveriai kuriuose nepatalpinta informacija.

Šio QoS skaičiavimo trūkumai tie, jog čia kalbama tik apie kompiuterių procesorių spartas ir kainas, tačiau neatsižvelgiama į tai, jog priėjimas prie klasterių gali nekainuoti, jog servisų gali būti daug, neatsižvelgiama į tinklo parametrus, kas taip pat aktualu tam tikrų uždavinių sprendimui.

2.2. QoS tyrinėjimai GRID skaičiavimuose

Augant visuomenės poreikiams, vieno superkompiuterio skaičiavimų technologijos dabar netinka kai kurioms modernioms didelės apimties programinėms įrangoms. Taigi plinta pasaulyje GRID skaičiavimai, technologija, kuri koordinuotų resursų dalinimąsi ir dinamišką problemų sprendimą virtualiose organizacijose [25].

GRID buvo pristatytas maždaug 1995m. Ir pasiekė didelį progresą paskutiniaisiais metais. Tačiau šioje srityje yra vis dar daug problemų tarp kurių QoS yra ypač svarbus ir

palyginti mažai iširtas. QoS svarbumas GRID glūdi mažiausiai trijuose aspektuose: GRID servisų koncepcija, vartotojų gausos reikalavimas ir įvairiarūšiai GRID resursai.

1. GRID serviso koncepcija. GRID išsivystė iš skaičiuojamųjų GRID, duomenų GRID ir kitų GRID dar vadinamų „servisų GRID“. Būtent GRID yra plačiai paskirstytas serveris, kuriame viskas veikia kaip servisai. Bendravimas tarp vartotojo ir GRID yra paremtas užklausų servisu. Bendravimas gali įvykti tiesiogiai tarp servisų. Taigi QoS yra vienas iš svarbiausių GRID savybių.

2. Vartotojų gausos reikalavimas. GRID yra atviresnis vartotojams nei internetas, kuris gali bendradarbiauti su įvairiais vartotojų reikalavimais. Vartotojai tikisi paprastai ir lengvumai gauti pageidaujamą servisą ir jų norai labai įvairūs. Kai kuriuos vartotojus domina tik serviso funkcijos, kai kuriems svarbiausias yra funkcionalumas, kiti kelti reikalavimus kainai, o dar kiti galbūt tikisi visko. Taip pat, tas pats vartotojas gali turėti skirtingus QoS reikalavimus skirtingoms programoms. Taigi QoS yra itin svarbu GRID tinkluose.

3. Įvairiarūšiai GRID resursai. GRID yra sukurtas ant įvairiarūšių platformų, skirtingų resursų, kategorijų ir išdėstymo. Tačiau jis turi teigti vienos sistemos įvaizdį vartotojams ir nematomą resursų koordinavimą, kad išpildytų vartotojų norus. Iš to išplaukia, kad QoS tiesiogiai susijęs su GRID veikimu.

2.3. QoS problemos GRID skaičiavimuose

GRID atvirumas ir dinamika kuria esmines QoS problemas GRID aplinkoje. Atvirumas suteikia įvairiarūšių servisų ir jų gausumo, vartotojų ir jų poreikių augimą. Dinamika, iš kitos pusės, leidžia GRID resursams, vartotojams, elgsenai keistis savarankiškai. Tačiau [26]:

1. Reikalingi aprašymai ir GRID QoS žymėjimas. Serviso užklausa gali būti patvirtinta kartu su ją atitinkančiais servisų kokybės parametrais, kad įsitikintų, jog užduotis buvo įvykdyta taip puikiai, kaip vartotojas tikėjosi. Mechanizmas, aprašantis vartotojo QoS reikalavimus, neturėtų būti tik lankstus ir ekspresyvus, kad patenkintų įvairius vartotojų skonius, bet taip pat turi būti lengvai naudojamas. Galutinio vartotojo QoS reikalavimai yra dažniausiai grubūs ir dėmesys sutelkiamas tik į serviso rezultatus. Vis dėlto GRID turi atsižvelgti į konkrečius resursus ir reikalingi išsamūs QoS parametrai. Visgi, kaip sumodeliuoti gerai aprašytą reikalavimų mechanizmą ir grubius QoS reikalavimus perversi į išsamius, GRID'ui suprantamus parametrus, yra sunki ir svarbi užduotis.

2. Servisų aprašymas. GRID yra ekstremaliai atvira ir dinamiška aplinka su savo dažnai pridedamais ir keičiamais servisais, kurie ir reikalauja aiškaus servisų aprašymo mechanizmo. Tinklo servisams WSDL dirba kaip šios problemos sprendimas. Deja, jis fokusuojasi tik į tai, kaip pasiekti servisą ir tai, ką jis daro, išskyrus kitą informaciją, kuri naudinga GRID QoS. Taigi susipažinti su QoS, kaip serviso aprašymo mechanizmu, WSDL yra nepakankamas. Naujas mechanizmas turi įtraukti ne tik tradicinius QoS faktorius, tokius kaip tinklo pralaidumą, vėlavimą, ir laiką, bet taip pat ir patogumą, saugumą, perskirstymą, ir kitus faktorius būdingus GRID. Tai daro mūsų servisų aprašymą GRID aplinkoje labai patrauklų, tačiau prastai išbandytą.

3. Adaptyvus QoS. GRID'e servais gali būti pakeisti ir ištrinti bet kuriuo metu, su laisvai įdiegiamais naujais servisais, ir netgi aktyvūs servais gali sustabdyti save, palikdami neužbaigtą darbą. Taip pat tinklo topologija, perdavimo talpa ir platforma, kurioje veikia servais, gali savarankiškai pasikeisti. GRID turi atrasti pasikeitimus ir greitai teisingai sureaguoti, kad suteiktų vartotojui geriausią QoS. Dėlto reikalingas permatomas (*transparent*) adaptyvus QoS mechanizmas GRID aplinkoje.

4. Resursų rezervavimas ir planavimas. Jei keli resursai yra įtraukti į atsakymą vartotojo užklausiai, GRID turi atlikti išankstinę jų rezervaciją, kad pasiektų aukščiausią QoS. Rezervavimas turėtų atitikti maksimalų globalų GRID funkcionalumą. Šiame kontekste apsvarstomas darbo apkrovos balansavimas ir resursų derybos. Kadangi vartotojo darbas yra atliekamas žingsnis po žingsnio, teisinga rezervacija ir planavimo politikos tikslai turi pasiekti geriausią bendrą QoS. Išties resursų rezervacija, o ypač perrezervavimas GRID aplinkoje ir planavimas yra įdomi ir svarbi GRID QoS dalis.

5. QoS analizė ir grįžtamasis ryšys. GRID QoS poveikis turėtų būti matuojamas tik pagal tai, kiek vartotojas yra patenkintas, ir tai yra rodiklis projektuojant ir gerinant QoS mechanizmus. Tačiau prezentacijos kokybė QoP (multimedia QoS) verčia susitelkti į kokybės kriterijų ir funkcinį rodiklių plėtojimą, kad sukurtų ryšį tarp QoS ir atsakomosios reakcijos.

3. GRID PROGRAMINĖ ĮRANGA

GRID yra kompiuterių resursų, išdėstytų po visą pasaulį rinkinys, galintis priklausyti kokiai nors organizacijai (privaciai ir viešai), kuriuo dalinamasi tarp vartotojų bendruomenės pagal specifinius apribojimus.

GRID koncepcija išplaukia iš šių pasvarstymų:

- Daug CPU resursų yra nenaudojama nakties metu ar įvairiu kitu laiku;
- Modernūs projektai, kaip HEP eksperimentų duomenų apdorojimas, reikalauja intensyvių skaičiavimų;
- Modernūs PC siūlo aukštą veikimo lygį žemomis kainomis;
- Beveik visi kompiuteriai pajungti į tinklą ir turi priėjimą prie visų kitų kompiuterių tinkle;
- Interneto naršyklės leidžia vykdyti sudėtingas operacijas.

Dėlto sujungiant didelį kiekį kompiuterių, kuriuose veikia specializuota valdymo programinė įranga, vadinama *middleware*, mes galime įdiegti skaičiavimo GRID ir sukurti galingą virtualią skaičiavimo aplinką, kur kiekviena įstaiga ar pavieniai asmenys gali užsisakyti ir paleisti sudėtingus programinius paketus.

GRID *middleware* sukurtas pagal atitinkamuose lygmenyse bendraujančius paketus ir gali būti sujungtas naudojant skirtingus procesus iššaukiamus bendro API, taip, kad vartotojams nerūpėtų skirtinga sintaksė ir priėjimas prie specifinių paketų metodų. Labiausiai paplitęs uždavinys duotas darbo planuoklio, tai gali būti bet kuris iš daugiau ar mažiau sudėtingų produktų rinkinių kaip Unix fork, Condor, LSF, PBS ir kiti.

Skirtingos užduotys yra įvykdomos egzistuojančios programinės įrangos, kaip MRTG veikimo stebėjimui ir MySQL statistikos saugojimui. Apibendrinant pagrindinę *middleware* metodologiją reiktų pabrėžti ir šiuos punktus:

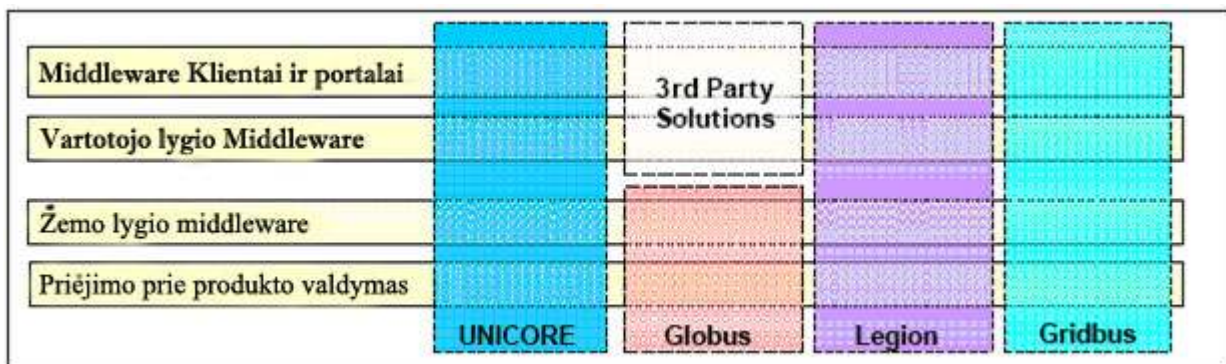
- GRID yra prie tinklo pririšta sistema, kuriai reikia efektyvių ir patikimų komunikacijų ir naršyklių su aukšto lygio saugumu.
- GRID yra susietas su duomenų bazėmis dėl šių funkcijų: GRID topologijos valdymo, vartotojų teisių ir sertifikatų nustatymų vietos ir resursų tikslios kopijos, bei dinamiškas veikimo valdymas ir vartotojų statistika.

- GRID yra susieta su efektyviais suderintais algoritmais: vartotojo sertifikato suderinimas, darbo resurso suderinimas su pirminiais duomenų objektais, pogrupių ir kopijų identifikavimas.
- GRID priklauso nuo tikslaus laikrodžio veikimo, sinchronizuojančio mazgus ir tiksliai valdantčio užduotis ir darbų planavimą.

Yra nemažai sukurta GRID *Middleware* sistemų. Paminėsiu kelias iš jų: Globus, Grasp, LCG, Systinet, IBM® Grid, UNICORE, Legion, Gridbus.

3.1. GRID programinės įrangos palyginimas

Palyginame GRID programinę įrangą (3 pav.) pagal servisų teikimą per GRID architektūros steką. INUCORE ir Legion yra tvirtai integruoti ir teikia abu serverio ir kliento komponentus. Globus teikia „servisų krepšelio“ (*bag-of-services*) priėjimą ir suteikia turtingą pagrindinių įrankių rinkinį, kurie gali būti panaudoti pasirinktinai konstruojant GRID sistemas. Gridbus komponentų rinkinys, kad ir išskleistas steke, nėra taip tvirtai integruotas, kaip UNICORE ar Legion ir gali būti panaudoti visiškai nepriklausomai vienas nuo kito.



3 pav. UNICORE, Globus, Legion ir Gridbus palyginimas

Palyginimas įvairių *middleware* sistemų, paremtų jų architektūra, realizavimo modeliais ir kategorijomis yra pateikta 4 lentelėje.

4 lentelė. UNICORE, Globus, Legion ir Gridbus palyginimas

<i>Middleware</i>	UNICORE	Globus	Legion	Gridbus
Ypatybės				
Dėmesio centras (koncentravimas)	Aukšto lygio programavimo modeliai	Žemo lygio servisai	Aukšto lygio programavimo modeliai	Abstrakcijos ir rinkos modeliai
Kategorija	Svarbiausiai pastovus darbo pateikimas ir stebėjimas	Bendras skaičiavimas	Bendras skaičiavimas	Bendras skaičiavimas

<i>Middleware</i> Ypatybės	UNICORE	Globus	Legion	Gridbus
Architektūra	Vertikali multi-eilių sistema	Sluoksninis ir modulinis įrankių komplektas	Vertikalčiai integruota sistema	Sluoksniniai komponentai ir pagalbiniai modeliai
Įgyvendinami modeliai	Abstraktūs darbo objektai (Abstract Job Object)	Smėlio laikrodžio modelis sisteminiame lygyje (Hourglass model at system level)	Objektiškai orientuota metasistema (Object-oriented Metasystem)	Smėlio laikrodžio modelis sisteminiame lygyje (Hourglass model at system level)
Įgyvendinamos Technologijos	Java	C ir Java	C++	C, Java, C# ir Perl
Veikimo platforma	Unix	Unix	Unix	Unix ir Windows su .NET (Alchemi)
Programavimo aplinka	Darbo eigos aplinka (Workflow Environment)	Unix & C bibliotekų keitimas. Specialios MPI bibliotekos Java, Python, CORBA, Matlab, Java Server Pages, Perl ir Web Servisai	Legion aplikacijų programavimo sąsaja (API). Komandinės eilutes paslaugos	Broker Java API XML-paremti parametrai-sweep language Grid Thread model via Alchemi.
Platinamas (Distribution) modelis	Atviro kodo (Open source)	Atviro kodo (Open source)	Ne atviro kodo. Galima komercinė versija	Atviro kodo (Open source)
Keletas vartotojų ir ir pritaikymų	EuroGrid [1], Grid Interoperability Project (GRIP) [3], OpenMolGrid [2], Ir Japanese NAREGI [4].	AppLeS [5], Ninf [7], Nimrod-G [6], NASA IPG [13], Condor-G [8], Gridbus Broker [9], UK eScience Project [10], GriPhyN [12], ir EU Data Grid [11].	NPACI Testbed [18], Nimrod-L [17], ir NCBioGrid [16]. Be to, tai buvo panaudota studijuojant ašių simetrijų pastovią srovę [15] ir proteino sudedamumo [14] aplikacijas.	ePhysics Portal [23], Belle Analysis Data Grid[22], NeuroGrid [21], Natural Language Engineering [24], HydroGrid [19], Ir and Amsterdam Private Grid [20].

Middleware Ypatybės	gLite	LCG
Dėmesio centras	Sutelkia dėmesį į produkciją, didelės apimties duomenų priežiūrą	Susitelkia analizėje
Kategorija	-	-
Architektūra	-	-
Įgyvendinami modeliai	-	-
Įgyvendinamos technologijos	-	-
Veikimo platforma	Linux, Unix ir Windows	-
Programavimo aplinka	-	-
Platinamas (Distribution) modelis	Atviro kodo (Open source)	-
Keletas vartotojų ir pritaikymų	-	-

Papildant aukščiau pateiktą palyginimą, *middleware* sistemos gali būti palygintos resursų valdymo pagrindu, duomenų valdymo, komunikavimo metodų ir saugumo. Globus ir Legion teikia brokerių servisus, tuo tarpu UNICORE ir Alchemi neteikia brokeriavimo (tarpininkavimo) tarp vartotojų užklausų. Alchemi atveju, Gridbus brokeris atlieka GRID tarpininko vaidmenį. Globus komunikacijų modulis yra paremtas Nexus komunikavimo bibliotekomis, INUCORE komunikacijų metodai yra paremti AJO modeliu ir nepalaiko sinchroninių pranešimų praėjimo. Legion palaiko RMI variacijas komunikavimui per LOIDs. UNICORE saugumas yra paremtas ant SSL (*Secure Socket Layer*) protokolo ir X.509V3 tipo sertifikatu. Globus teikia saugumo servisą per GSI, kuris vėlgi paremtas ant SSL ir X.509 sertifikatu. Gridbus' o Alchemi *middleware*, saugumo servisas yra išpirktas iš galingo įrankių rinkinio Microsoft .NET sistemos (*framework*). Kiti Gridbus komponentai daugiausiai išpirkti iš saugumo servisų teikiamų Globus GSI.

GRID sistema, kurioje dar daug neišspręstų klausimų:

- kol kas nėra vieno aiškaus GRID apibrėžimo, nėra bendro standarto;
- kai vartotojų ir resursų yra labai daug, tampa sudėtinga kurti ir sinchronizuoti visų vardus ir slaptažodžius;
- per maži tinklo pralaidumai;
- per silpni duomenų valdymo serveriai;
- skirtingos skaičiavimo ir saugyklų techninės charakteristikos;
- skirtinga vartotojų kvalifikacija;
- skirtingos taikymo sritys.

4. KURIAMOS METODIKOS BEI PROGRAMINĖS ĮRANGOS ANALIZĖ

4.1. Projekto iniciatoriai (*Project Drivers*)

4.1.1. Sistemos paskirtis

Metodika bei programų rinkinys suteiks galimybę GRID vartotojui įvertinti klasterio pajėgumus jo norimam uždaviniui spręsti.

Projekto kūrimo pagrindas (pagrindimas)

Didėjant servisų kokybės poreikiams, reikalinga metodika šiems kokybės parametrams apskaičiuoti, kad būtų galima klasterius ar GRID tinklus palyginti tarpusavyje kokybės atžvilgiu.

Sistemos tikslai (paskirtis)

Ši metodika bei programų rinkinys skirti GRID klasterių servisų kokybės tyrimams atlikti.

4.1.2. Užsakovai, vykdytojai ir kiti sistema suinteresuoti asmenys

Užsakovas. Tai Kauno technologijos universitetas, kuris koreguoja, prižiūri metodikos ir jos parašymo bei programų vystymo raidą, taip pat ginantis tezes įvertina šį darbą.

Vykdytojas. Studentas, kurio baigiamojo darbo tema „GRID teikiamų servisų efektyvumo analizė“.

Kiti sprendimus priimančys asmenys. Tai baigiamojo darbo vadovas bei Informatikos fakulteto tinklų katedra, kuri pakoreguoja darbo vykdymo eigą bei įvertina baigiamojo darbo vykdymo tarpinius etapus per tarpinius atsiskaitymus.

4.1.3. Vartotojai

Ši metodika bei programų rinkinys skirti naudoti vartotojams, kurie turi priėjimą prie kelių klasterių, ar GRID, ir gali pasirinkti kuri klasterį jiems naudoti. Ši metodika padeda apsispręsti ir įvertinti klasterius pagal vartotojui aktualius kriterijus.

4.2. Projekto apribojimai

Metodika bei programos sukurtos tirti BalticGrid skaičiuojamajam GRID mažos apimties uždavinių sprendimui. Ši metodika taikant kitokio tipo GRID gali būti plečiama, pridedant prie galutinės formulės papildomus parametrus.

4.2.1. Įpareigojantys apribojimai (Mandated constraints)

Apribojimai sprendimui

Sukurtos programos uždavinių vėlinimo skaičiavimui veikia tik ant gLite GRID programinės įrangos. Jei būtų naudojama kitokia programinė įranga, uždavinius tektų modifikuoti. Uždaviniai ant kitokios programinės įrangos netestuoti.

Diegimo aplinka

Metodika bei serverinės programos turėtų būti vykdomos kompiuteryje ar serveryje, iš kurio leidžiamas priėjimas prie tiriamo GRID ar norimos tirti klasterių grupės. Taip pat serveryje ar kompiuteryje turi būti leidžiamos vykdyti GRID komandos.

Bendradarbiaujančios sistemos

Programos atliekančios skaičiavimus bendradarbiauja su GRID programine įranga, tiksliau sukurtos programos tarsi GRID uždaviniai, kurie siunčiami į GRID, kad būtų sprendžiami.

Komerciniai specializuoti programų paketai

Servisų kokybės efektyvumo skaičiuoklė ir saugumo lygio skaičiuoklė veikia tik MS Windows aplinkoje. Parametrus nuimančios programos yra pritaikytos vykdyti skaičiavimus bei rinkti duomenis Linux aplinkoje, tačiau nesunkiai gali būti modifikuotos ir Windows operacinei sistemai.

Numatoma darbo vietos aplinka

Vartotojas prie programinės įrangos gali jungtis iš nutolusio kompiuterio, taigi darbo vieta mobili.

Sistemos kūrimo terminai

Metodika, programos bei dokumentacija turi būti baigtos aprašyti iki 2007 m. Gegužės mėnesio. Šis darbas naudojamas ginantis informatikos magistro laipsnį.

Sistemos kūrimo biudžetas

Metodikos kūrimui nėra numatyta išskirti lėšų bei finansavimo. Taip pat šis kūrimo procesas ir nereiklus investicijoms.

4.2.2. Svarbūs faktai ir prielaidos

Šiame darbe kuriama metodika apima pasiekiamumo, prieinamumo, operatyvios parengties koeficientą, vėlinimą, vėlinimo fliuktuacijų, pralaidumo, laisvų CPU parametrų apskaičiavimo būdus bei jų suvedimas į bedimensinius, t.y. sulyginamus tarpusavyje matmenis. Taip pat pateiktos ir programos šiems parametrams nuskaityti nuo sistemos.

Tačiau naudojant šias programas kai kuriems parametrams gali kilti problemų. Kaip, pavyzdžiui, pasiekiamumas, paketinis vėlinimas, paketinio vėlinimo fliuktuacijos. Tai gali būti dėl to, kad klasteris naudoja ugniasienę ir nepraleidžiančia PING paketų.

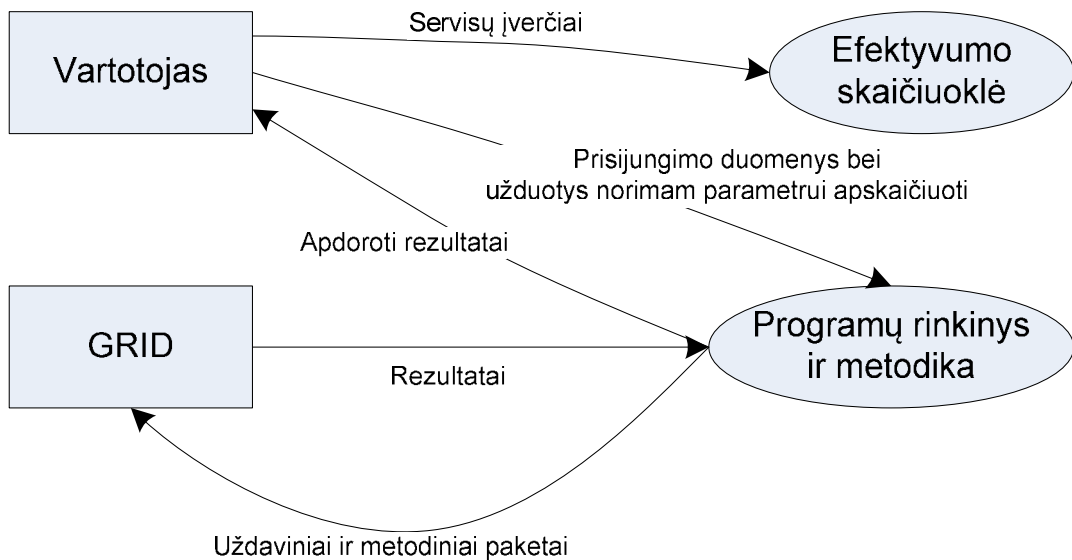
Taip pat vartotojas savo parinktam uždaviniui gali pareikalauti kitokių parametrų įvertinimų, kuriems apskaičiuoti programų nėra. Tuomet vartotojas turėtų rasti alternatyvių būdų šiems parametrams nuo sistemos nuskaityti. Tačiau rezultatų suvedimo į bedimensinius dydžius, kuriuos būtų galima lyginti tarpusavyje ir panaudoti efektyvumo skaičiavimo formulėje, skaičiavimui galima panaudoti metodikoje pateiktas formules, nes jos universalios ir pritaikytos visiems atvejams.

4.3. Funkciniai reikalavimai

4.3.1. Veiklos sfera

Veiklos kontekstas (pateikiama konteksto diagrama)

Nagrinėjamai veiklos sričiai apibrėžti naudojama „Konteksto diagrama“. Veiklos kontekstas apima plačiau, nei kuriamos metodinės sistemos atliekamos funkcijos. Veiklos kontekstas apibrėžia dominančią veiklą ir jos naudojamus bei formuojamus informacijos srautus.



4 pav. Konteksto diagrama

Vartotojas norėdamas tirti GRID klasterių efektyvumą naudojami programų rinkiniu bei metodika, kurioje nurodyta, kaip šiuo programų rinkiniu naudotis, ir ką reiškia, kuris rezultatas. Susirinkęs rezultatus, vartotojas įverčius surašęs į programą, ar tiksliau efektyvumo skaičiuoklę, gauna klasterių įverčius (klasterių efektyvumą).

Veiklos padalinimas

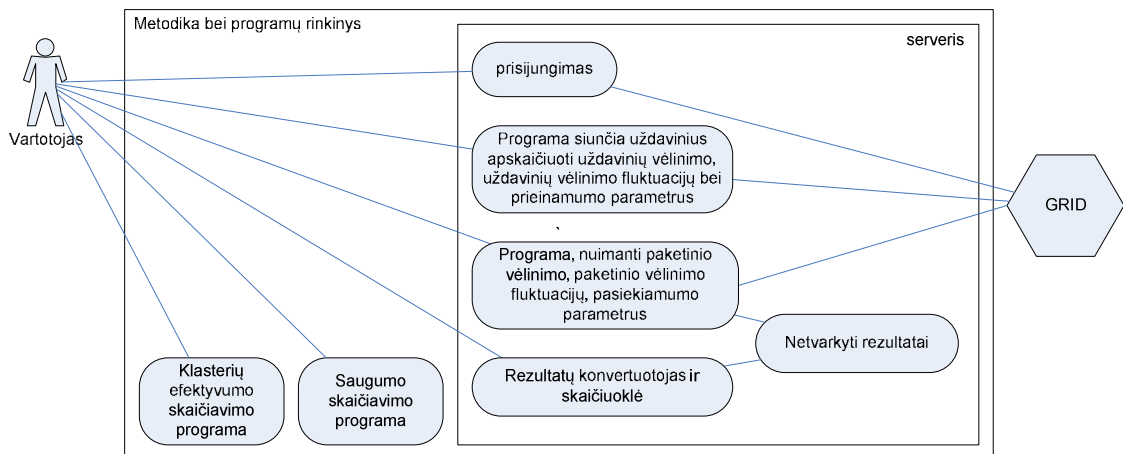
GRID efektyvumo tyrimas susideda iš vartotojo, jo norų, programų rinkinio ir metodikos, taip pat GRID tinklo. Žemiau pateikta, kaip šie dalyviai tarpusavyje pasidalina veiklas (5 lentelė).

Eil. nr.	Įvykio pavadinimas	Įeinantys/išeinantys informacijos srautai
1	Vartotojas norėdamas prisijungti prie serverio autorizuojamas.	Vartotojo prisijungimo duomenys (in)
2	Vartotojas norėdamas prisijungti prie GRID autentifikuojasi su savo sertifikatu.	Vartotojo sertifikatas ir slaptažodis duomenys (in)
3	Vartotojas užsiduoda parametrus, pagal kuriuos nori tirti GRID	Vartotojo užsiduoti duomenys (in)
4	Programos tiriančios užsiduotus servisu jungiasi prie GRID tinklo klasterių ir siunčia uždavinius ar užklausas parametrus nuimti.	Programų uždaviniai ar užklausas parametrų nuėmimui (out)
5	GRID klasteriai grąžina uždavinių rezultatus ar atsakymus į užklausas.	Uždavinių rezultatai ir atsakymai į užklausas (in)
6	Apdoroja GRID grąžintus uždavinius ir atsakymų į užklausas rezultatus bei apskaičiuoja tam tikrus parametrus.	Apdoroti GRID tinklo rezultatai (out)
7	Pildo efektyvumo skaičiuoklę apdorotais rezultatais ar paties vartotojo apskaičiuotais įverčiais, o programa grąžina efektyvumo įvertinimą.	Servisų įverčiai (in) Efektyvumo įvertinimai (out)

4.3.2. Produkto veiklos sfera

Sistemos ribos

Ribas tarp mano pateiktos metodikos ir vartotojo nusako panaudos atvejų diagrama.



5 pav. Panaudos atvejų diagrama

Panaudojimo atvejų sąrašas

Panaudojimo atvejų diagramoje pavaizduota kokiomis funkcijomis naudodamasis vartotojas ar vartotojų grupė gali tirti GRID servisų kokybės efektyvumą ir šių sukurtų funkcijų tarpusavio sąveika su vartotoju, tarpusavyje ir su GRID.

1. PANAUDOJIMO ATVEJIS: Vartotojo autorizacija bei autentifikacija

Vartotojas/Aktorius:	GRID vartotojas
Aprašas:	Apima procesą, vartotojas jungiasi prie GRID, kad galėtų naudotis GRID servisais.
Prieš sąlyga:	Asmuo nesusikūręs sertifikuoto laikinojo <i>proxy</i> ; Asmuo neautentifikuotas.
Sužadinimo sąlyga:	Vartotojas su savo sertifikatu kuria laikinąjį proxy serverį įvesdamas sertifikato slaptažodį.
Po sąlyga:	GRID autentifikavo vartotoją.

2. PANAUDOJIMO ATVEJIS: uždavinių vėlinimo, uždavinių vėlinimo fliktuacijų bei prieinamumo parametrų nuėmimas.

Vartotojas/Aktorius:	GRID vartotojas
Aprašas:	Apima procesą, kuomet vartotojas paleidžia programą skirtą nuimti GRID parametrus ar jų vertes, kurios reikalingos uždavinių vėlinimo, uždavinių vėlinimo fliktuacijų bei prieinamumo skaičiavimui.
Prieš sąlyga:	Vartotojas neturi jokių duomenų apie uždavinių vėlinimo, uždavinių vėlinimo fliktuacijų bei prieinamumo parametrus.
Sužadinimo sąlyga:	Vartotojas naudodamasis komandine eilute paleidžia programą, kuri siunčia uždavinius klasterių laiko nuėmimui į GRID.
Po sąlyga:	Vartotojas turi duomenis apie uždavinių vėlinimo, uždavinių vėlinimo fliktuacijų bei prieinamumo parametrus.

3. PANAUDOJIMO ATVEJIS: paketinio vėlinimo, paketinio vėlinimo fliuktuacijų, pasiekiamumo parametrų nuėmimas.

Vartotojas/Aktorius: GRID vartotojas

Aprašas: Apima procesą, kuomet vartotojas paleidžia programą skirtą nuimti GRID parametrus, ar jų vertes, kurie reikalingi paketinio vėlinimo, paketinio vėlinimo fliuktuacijų bei prieinamumo skaičiavimui.

Prieš sąlyga: Vartotojas neturi jokių duomenų apie uždavinių vėlinimo, uždavinių vėlinimo fliuktuacijų bei prieinamumo parametrus.

Sužadinimo sąlyga: Vartotojas naudodamasis komandine eilute paleidžia programą, kuri siunčia PING paketus į vartotojo sudarytą klasterio sąrašą, bei rezultatus gražina į dokumentą, kurį naudodama PHP programa apskaičiuoja vidutinį vėlinimą, vėlinimo fliuktuaciją ir pasiekiamumą.

Po sąlyga: Vartotojas turi sutvarkytus duomenis apie paketinio vėlinimo, paketinio vėlinimo fliuktuacijų bei prieinamumo parametrus.

4. PANAUDOJIMO ATVEJIS: saugumo parametro skaičiavimas.

Vartotojas/Aktorius: GRID klasterio administratorius

Aprašas: Apima procesą, kuomet klasterio administratorius programoje naudodamasis programa apskaičiuoja saugumo lygį.

Prieš sąlyga: Vartotojas nežino klasterio saugumo lygio skaitinės vertės.

Sužadinimo sąlyga: Vartotojas pažymi jo klasteryje naudojamus saugumo parametrus, bei įvertina juos svarbos svorio koeficientais, pagal kuriuos programa apskaičiuoja jo prižiūrimo klasterio saugumo lygį.

Po sąlyga:	Vartotojas turi sutvarkytus duomenis apie paketinio vėlinimo, paketinio vėlinimo fliktuacijų bei prieinamumo parametrus.
-------------------	--

5. PANAUDOJIMO ATVEJIS: efektyvumo skaičiavimas.	
Vartotojas/Aktorius:	GRID vartotojas
Aprašas:	Apima procesą, kuomet vartotojas pagal gautis rezultatus apskaičiuoja efektyvumą.
Prieš sąlyga:	Vartotojas nežino klasterių servisų kokybės efektyvumo.
Sužadinimo sąlyga:	Efektyvumo skaičiavimo programoje užpildo reikiamus laukus gautais rezultatais ir paspaudžia mygtuką skaičiuoti.
Po sąlyga:	Vartotojas turi kiekvieno tiriamo klasterio efektyvumo įvertinimą ir gali juos palyginti tarpusavyje.

4.4. Nefunkciniai reikalavimai

GRID servisų kokybės efektyvumo skaičiavimo metodika turi būti lengvai plečiama ir pritaikoma įvairių tipų GRID. Reikalinga galimybė prie skaičiavimo metodikos pridėti naujų parametrų, kurie gali atsirasti naudojantis kitais, ne skaičiavimo GRID.

4.4.1. Reikalavimai sistemos išvaizdai

Reikalavimų vartotojo sąsajai nėra. Iš programų rinkinio, vartotojo sąsają turi tik efektyvumo ir saugumo lygio skaičiuoklė. Kitos programos paleidžiamos bei skaitomi rezultatai naudojant komandinę eilutę.

4.4.2. Reikalavimai panaudojamumui

Metodika lengvai pritaikoma ir papildant parametrų sąrašą naujais. Metodikoje aiškiai aprašyta kaip suvesti parametrus į bedimensinius dydžius, kurie būtų lyginami tarpusavyje ir kuriuos būtų galima panaudoti galutinėje formulėje vertinant klasterio efektyvumą.

4.4.3. Reikalavimai vykdymo charakteristikoms

GRID servisų kokybės tyrimas atliekamas per paties vartotojo nustatytą laiko tarpą. Kuo ilgesnis tyrimo laiko tarpas, tuo reprezentatyvesnė informacija gaunama, tuo tikslesnis servisų kokybės efektyvumo įvertis gaunamas.

4.4.4. Reikalavimai veikimo sąlygoms

Kad atliktų tyrimą pagal metodiką, vartotojas privalo turėti priejimą prie GRID teikiamų paslaugų, taip pat privalo turėti internetą bei kompiuterį, iš kurio jungsis prie GRID. Darbo vieta mobili. Tyrimams skirtą programinę įrangą bei gražintus rezultatus vartotojas gali pasiekti iš nutolusio kompiuterio.

4.4.5. Reikalavimai sistemos priežiūrai

Metodika – tai ne sistema ir ji priežiūros nereikalauja, paprasčiausiai atliekant tyrimus ji gali būti naudojama, ja remiamasi arba ne.

4.4.6. Reikalavimai saugumui

Metodikai bei programoms, kuriomis naudojantis atliekami tyrimai saugumo reikalavimo nėra.

4.5. Projekto išeiga

4.5.1. Atviri klausimai

Metodikos taikymas naudojantis mūsų pasirašytais formulėmis gali būti atliekamas tik skaičiuojamiesiems GRID ir labiau tinkantis tirti klasterius norint išsiaiškinti, kurie iš jų labiau tinka spręsti mažesnės apimties uždaviniams, kur aktualus greitas atsakymas. Tačiau itin didelės apimties uždaviniams ar kitokio tipo GRID tirti, nėra programų pritaikytų nuimti reikiamus naujo tipo parametrus. Bet formules ir metodiką, papildomų parametrų pridėjimui prie gautinės skaičiavimo formulės, galima lengvai pritaikyti visiems GRID parametrų tipams, kur aiškus parametro teigiamas ar neigiamas dydis.

4.5.2. Naujos problemos

Galimos GRID saugumo sukeltos problemos

Atliekant paketinio vėlinimo, paketinės vėlinimo fliuktuacijos ir pasiekiamumo tyrimus gali kilti problemos, jei klasterių administratoriai uždraudžia PING paketus ar jų ugniasienė nepraleidžia.

4.5.3. Vartotojo dokumentacija ir apmokymas

Magistro tezes galima laikyti šios metodikos dokumentacija ir vartotojo vedliu, kaip ją pritaikyti, atliekant GRID servisų kokybės tyrimus.

5. SERVISŲ KOKYBĖS TYRIMAS

QoS (paslaugų kokybė) – tai interneto ir kitų tinklų sąvoka, nusakanti, kaip išmatuoti ir padidinti perdavimo greičius, sumažinti klaidų santykį. QoS yra suvokiamas kaip nenutrūkstamas paslaugos veikimo užtikrinimas (ypatingai video ar multimedia informacijai). Paslaugų kokybės užtikrinimas ypač sudėtingas įprastiniuose IP tinkluose naudojančiuose taip vadinamus „geriausių pastangų“ (*best effort*) protokolus, nes tada kai jie atsirado nebuvo didelio poreikio paslaugos kokybės užtikrinimui [32].

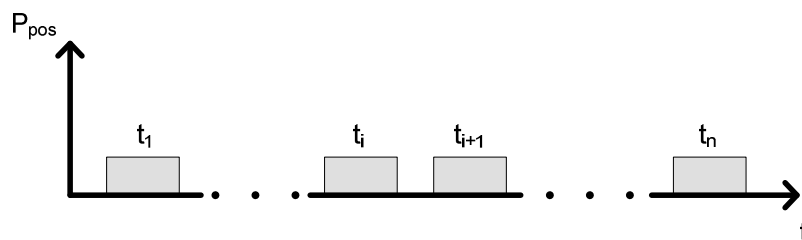
QoS apibūdinamas daugeliu parametru iš kurių pagrindiniai yra: pasiekiamumas, prieinamumas, patikimumas, saugumas, vėlinimas, vėlinimo fliktuacijos, pralaidumas, panaudojimas, paketų praradimas [33].

5.1. Pasiekiamumas

Apibrėžimas: Pasiekiamumas suprantamas kaip parametras, parodantis, ar sistema pasirengusi teikti paslaugą vartotojui užklausus. Pasiekiamumas parodo tikimybę, ar sistema bus pasiekiamas tam tikru laiko momentu. Jeigu šis parametras K_A artėja prie vieneto, tai reiškia, kad paslauga bus pasiekiamas visada.

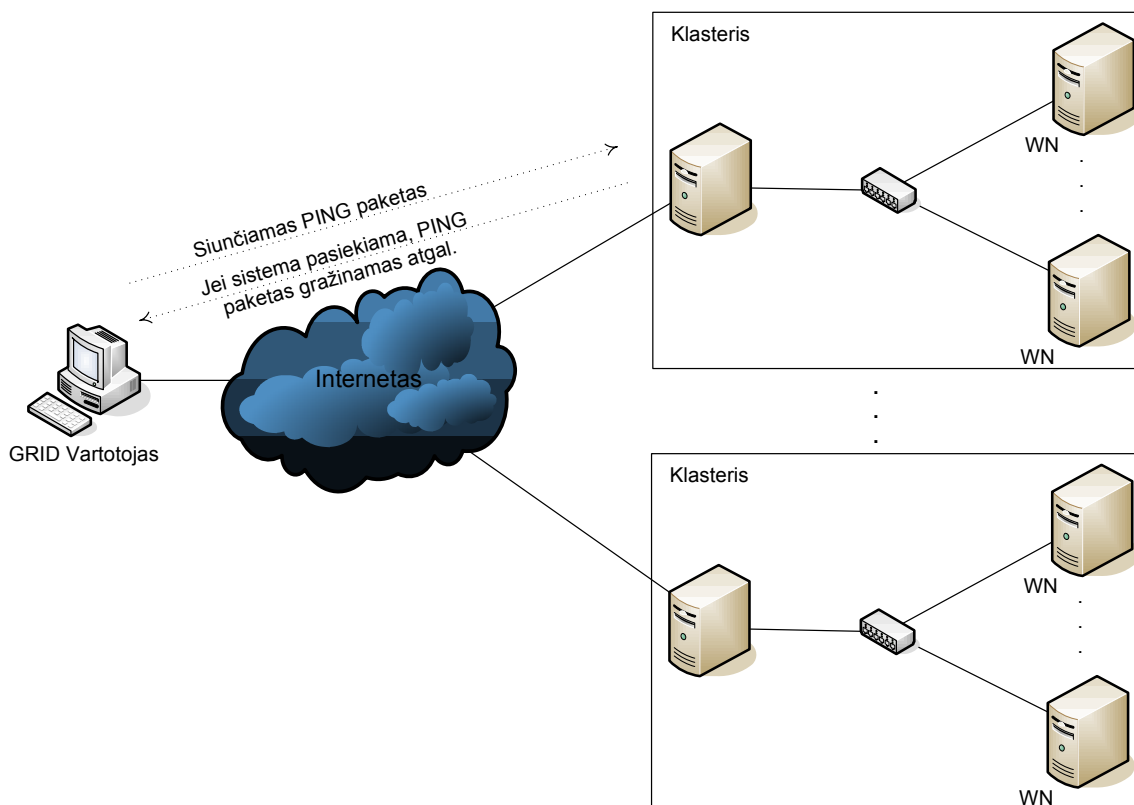
$$K_A = \frac{\sum_{i=1}^n t_i}{t}, \quad (13)$$

čia: t_i – laiko tarpai, kada pasiekiamas yra paslauga; n – pasiekiamų t_i laiko tarpų skaičius; t – visas periodas, kurio metu atliekami skaičiavimai. Sistemose paslaugos pasiekiamumo dinamika gali atrodyti kaip 6 paveikslėlyje, kur P_{pos} – paslaugos pasiekiamumas, o patį pasiekiamumą galima paskaičiuoti pagal 13 formulę.



6 pav. Paslaugos pasiekiamumo dinamika

Siūloma: Patikrint, ar sistema (klasteris) šiuo metu pasiekiamas, galima siųsti PING paketus. Jei sistemos administratorius nebus uždraudęs šių paketų, tuomet sistema būdama pasiekiamas, šiuos paketus gražins atgal, priešingu atveju, visi paketai bus prarasti, ir tai reikš, kad sistema nepasiekiamas. Siūlomas pasiekiamumo skaičiavimo būdas grafiškai pavaizduotas (7 pav.).



7 pav. Siūlomas pasiekiamumo skaičiavimo būdas

Metodika: Parašytoje programoje (*script*) surašyti klasterių adresai, kurių pasiekiamumą skaičiuojame. Joje taip pat nurodomas laiko intervalas, kas kiek laiko PING paketas turėtų būti leidžiamas. Jei šią programą planuojama paleisti ilgam laikui, tai paleisti geriausia foniniam režime. Gražinami duomenys nukreipiami į norimą failą. Gautą rezultatų failą apdoroja PHP programa, kuri atrenka reikiamus duomenis iš failo bei apskaičiuoja pasiekiamumą ir kelis kitiems skaičiavimams reikalingus parametrus.

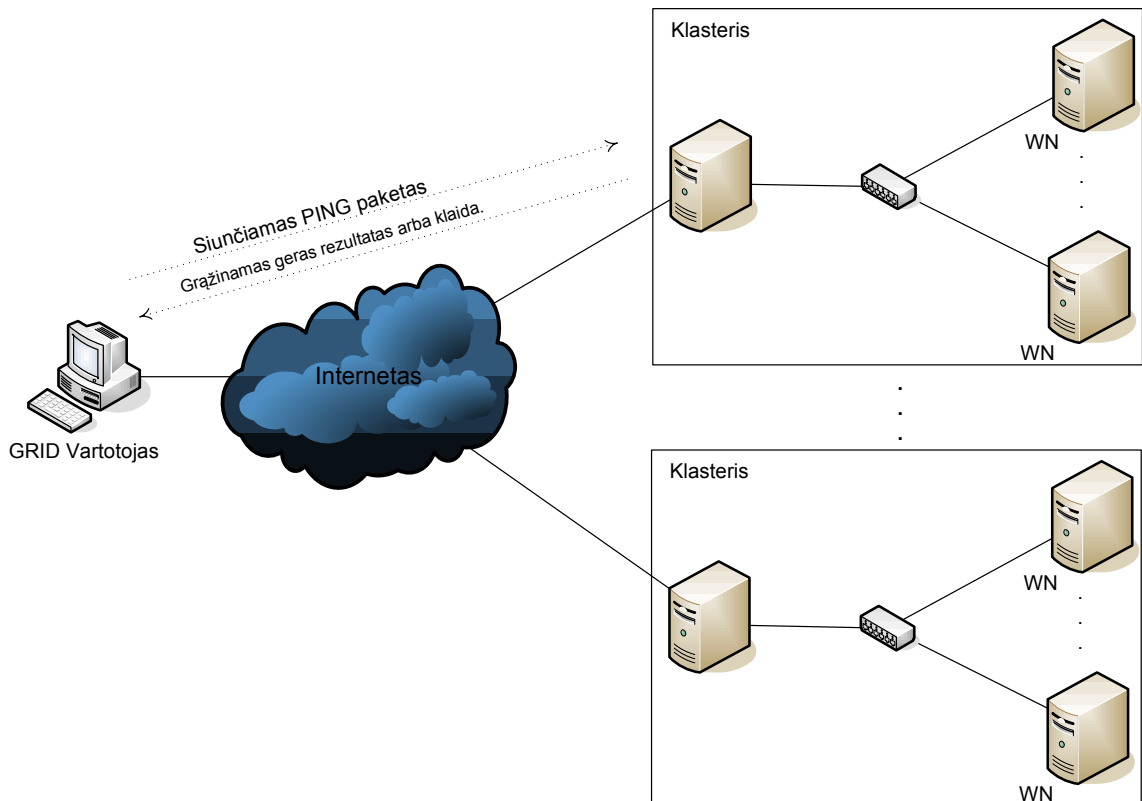
Rezultatai: Testas buvo atliekamas apie 24 dienas (11 lentelė), tikrinant pasiekiamumą kas valandą, į klasterį paleidus po 2 PING paketus.

5.2. Prieinamumas

Apibrėžimas: Prieinamumas - tai laiko procentas, kai sistema yra funkcionali galiniam vartotojui. Sistema gali būti pasiekiamas, bet neprieinamas. Prieinamumas yra

dvejtainis matavimo vienetas – servisas prieinamas arba neprieinamas. Galiniam vartotojui faktas, kad tam tikri serviso elementai yra neprieinami nekelia rūpesčių jei šis neprieinamumas yra paslepiamas naudojant sistemos elementų dubliavimą.

Siūloma: Tam, kad patikrinti, ar sistema funkcionali esamu laiko momentu, galima nusiųsti trumpą uždavinį į klasterį ir sulaukti rezultato (8 pav.). Jei uždavinys bus išspręstas – sistema funkcionali, priešingu atveju nefunkcionala. Atlikus eilę tokių patikrinimų, iš gautų rezultatų galima apskaičiuoti procentinį arba tikimybinį prieinamumo parametą.



8 pav. Siūlomas prieinamumo skaičiavimo būdas

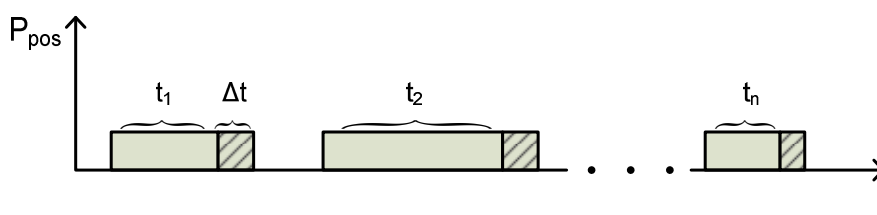
Metodika: Buvo tyrinėjamas realus BalticGrid tinklas, kurio oficialiame puslapyje pateikta statistika apie prieinamumą skaičiavimo ir duomenų saugyklos paslaugas teikiančius klasterius, priklausančius BaltiGrid virtualiai organizacijai, kuriame pateikiama paskutinių 4 dienų statistika [27]. Taigi skaičiavimai ir buvo atlikti remiantis BalticGrid pateiktais stebėjimo duomenimis.

Rezultatai: Klasterių statistika buvo renkama savaitę. Ir sukaupti rezultatai sugrupuoti bei susisteminti. Microsoft Office Excel 2003 pagalba apskaičiuotas prieinamumo parametras kiekvienam klasteriui (12 lentelė ir 13 lentelė).

5.3. Operatyvios parengties koeficientas

Apibrėžimas: Šis parametras nurodo sugebėjimo išlaikyti paslaugos kokybės laipsnį. Gedimų skaičius per mėnesį ar metus parodo GRID paslaugų patikimumą. Kitaip tariant patikimumas laikomas sėkmingai gautų ir išsiųstų paketų ar įvykdytų užduočių kiekiu.

Apibrėžimas: Šis parametras nurodo sugebėjimo išlaikyti paslaugos kokybės laipsnį. Jei norint išspręsti uždavinį klasteris pasiekiamas, tuomet aktualu, kad būtų prieinama paslauga reikalinga uždaviniui spręsti. Kiekvieno uždavinio sprendimas užtrunka tam tikrą laiko tarpą Δt (9 pav.). Klasteris teikia paslaugą t_i laiko intervalais. Taigi, jei uždavinys bus pradėtas spręsti likus mažiau nei Δt laiko iki paslaugos teikimo pabaigos, tuomet uždavinys nebus išspręstas ir bus prarasti jo duomenys. Jei uždavinys pradedamas spręsti anksčiau, tuomet tikimybė, jog uždavinys bus išspręstas žymiai išauga.



9 pav. Operatyvios parengties koeficiento vaizdinis modelis

Siūloma: kadangi vienam resursui gali būti siunčiami vieno ar kelių tipų uždaviniai, taigi skaičiuojant operatyvios parengties koeficientą reikėtų į tai atsižvelgti.

Metodika: Formulė (14) skirta apskaičiuoti operatyvios parengties koeficientą K_0 vienam resursui:

$$K_0 = \frac{\sum_{i=1}^I (t_i - \Delta t)}{t_{\Sigma}}, \quad (14)$$

čia K_0 – operatyvios parengties koeficientas; I – laiko intervalų skaičius; t_i – i -tasis laiko tarpas kai galima pasirinkti resursus; Δt – laiko tarpas kurį užtrunka sistema atlikdama užduotą darbą; t_{Σ} – visas matavimo laikas.

Formulė (15) skirta apskaičiuoti operatyvios parengties koeficientą N tipų uždaviniams vienam resursui:

$$K_0 = \frac{\sum_{n=1}^N \sum_{i=1}^I (t_i - \Delta t_n)}{N \cdot t_\Sigma}, \quad (15)$$

čia N – paslaugų tipai; Δt_n - laiko tarpas reikalingas atlikti n -tojo tipo uždaviočiai

5.4. Saugumas

Apibrėžimas: Tai WEB paslaugų kokybės parametras, kuris nusako kaip yra užtikrinamas ta paslauga besinaudojančių asmenų konfidencialumas. Saugumas pasidarė labai aktualus šiomis dienomis, kadangi dauguma WEB paslaugų prieinamos viešai. Paslaugos tiekėjas gali taikyti skirtingus saugumo lygius skirtingiems paslaugos vartotojams.

GRID saugumas ypač aktualus parametras, kadangi pati GRID specifika tinklą suskirsto į daugelį virtualių tinklų, kurių saugumas privalo būti užtikrinamas. Todėl čia paslaugos kokybės užtikrinimui yra naudojami saugūs protokolai SSL, X.509. Naudojamas Kerberos šifravimo algoritmas.

GRID saugumo lygio nustatymas atliekamas stebint sekančius GRID serverių parametrus:

- **Centrinis SYSLOG serveris**

Jei sistema buvo sukompromituota, tai labai tikėtina, kad atakuotojas ištrins LOG failus, kurie buvo sukurti įsibrovimo metu. Sistemos LOG failai visuomet yra vertingi po incidentinės analizės metu, ir LOG failai sukompromituotoje sistemoje negali būti patikimi, nes atakuotojas galėjo juos pakeisti [28].

Ši problema gali būti (bent dalinai) išspręsta naudojant centrinį SYSLOG serverį. Toks serveris gali rinkti sistemos LOG failus iš įvairių tinklo mazgų. Kaip pasekmė, tai atakuotojas gali pakeisti LOG failus sukompromituotoje sistemoje, tačiau itin sudėtinga būtų pakeisti šiuos failus centriniame SYSLOG serveryje.

Taigi šio SYSLOG serverio naudojimas GRID tinkle būtų itin vertingas. Tai gali dažnai paspartinti serviso atstatymą po incidento, ir užkirsti galimybę tokiai pat atakai įvykti ateityje.

- **Sistemos metrikos**

Svarbu stebėti sistemos metrikas, nes tai padeda užtikrinant serviso prieinamumą, tolydumą ir kartais atskleisti įsibrovimus[28].

Pavyzdžiui, gali būti vertingi staigūs pranešimai apie neveikiantį vieną iš servisų.

- **Sistemos veikimas, charakteristika**

Stebint sistemos veikimą ir apžvelgiant jos evoliuciją įprastoje aplinkoje gali būti labai naudinga. Tai leidžia sistemoms šeiminiams stebėti buvusius ir dabartinį serviso naudojimą ir taip pat numatyti ateities veikimą daugeliu atveju.

Taip pat yra didelis saugumo susidomėjimas sistemos veikimo stebėjime. Dauguma atakų tikrai paveikia sistemos charakteristikas, veikimą, tokius kaip kietojo disko ar tinklo išnaudojimą. Netgi jei ilgas sistemos stebėjimo laikas bus nepakankamas, kad aptiktume sistemos sutrikimą, tai dažnai duoda naudingos informacijos, kad suprasti kaip ir kada sistemos sukompromitavimas įvyko.

Yra keli įrankiai sukurti sistemos veikimo stebėjimui. „Ganglia“ yra vienas iš žinomiausių įrankių, ypač sukurtų klasteriams ir kompiuterių grupėms. Tai leidžia vartotojui analizuoti daugelio faktorių evoliuciją, kad aptikti veikimo sutrikimus ar silpnas vietas WEB puslapio pagalba [28].

Šis stebėjimas naudingas ne tik saugumo ar stabilumo atžvilgiu, tačiau ir klasterių apkrautumo, išnaudojimo statistinių duomenų rinkimui, kurie gali būti naudingi vertinant klasterio efektyvumą.

- **Sistemos pataisymų būseną**

„Yumit“ teikia stebėjimą ir pranešimų mechanizmą, kad patikrintų taisymo (patching) būseną Linux sistemose.

Saugumo pataisymai yra paprastai galimi iš Yum ar apt-get saugyklų Red Hat Linux Legacy produktams, Fedora Core ir Scientific Linux, ir iš Red Hat tinklų per up2date įrankius Red Hat Enterprise Linux.

Kartą įdiegtas kliento hoste, Yumit kiekvieną naktį patikrins ar galimi nauji pataisymai ir praneša apie tai Yumit serveriui. Automatiškai neįdiegia pataisymų.

Taip Yumit WEB puslapis pateikia sąrašą registruotų sistemų ir sąrašą paruoštų jiems pataisymų. Tai padeda sistemos administratoriui palaikyti daug mašinų su naujausiais pataisymais ir išvengti retai naudojamų mašinų užmiršimo atnaujinti [28].

Pataisymų atnaujinimas ypač naudingas saugume, nes pataisymai ir skirti dažniausiai tam, kad užlopyti saugumo skylės sistemoje.

- **Blokuoti BATCH darbus nuo SSH “back doors” kūrimo**

Męs norėdami sustabdyti BATCH darbus, keičiame jų pagrindinę aplinką į tokią, kad vartotojas galėtų naudoti SSH prisijungimui prie BATCH darbininko vėlesniam laikotarpiui? Tam tikras SEs, CEs, WNs ir RBs visi suteikia vartotojui tam tikrą priėjimo lygį prie BATCH darbų ar su GRIDFTP [28].

Tai reiškia, kad negalima leisti naudotis vartotojui visomis BACH komandomis, o tik leisti tam tikrų komandų grupę. Tai tikrai geras būdas saugoti sistemą, nuo jau prisijungimą prie sistemos turinčių vartotojų piktų kėslių.

- **Sistemos ugniasienės konfigūravimas**

Netgi jei puslapio ugniasienė siūlo svarbią apsaugą, asmeninė ugniasienė pridės papildomą apsaugos sluoksnį.

Tai leidžia sistemai būti apsaugotai prieš:

- Kitas sistemas iš lokalaus tinklo. Tai labai tikėtina, kai atakuotojas puolantis sistemą pabandys atakuoti kitas mašinas tame pačiame tinkle.

- Nekorektišką puslapio ugniasienės konfigūraciją.

- Nekorektišką sistemos konfigūraciją, kurioje gali būti paleisti nereikalingi tinklo servaisai.

- Nuo piktavališkų vartotojų leidžiančių neteisėtus servisus.

Gaunami prisijungimai Linux sistemoje gali būti filtruojami įvairiais metodais, tokiais kaip IPTABLES [28].

Yra dviejų požiūrių ugniasienės konfigūracijos. Uždrausti visus priėjimus ir leisti tik tam tikrus, reikalingus vartotojams, arba leisti visus ir uždrausti tik žinomus priėjimus, kurie netinkami ar gali būti piktavališkai naudojami. Pirmasis požiūris uždraudžiant viską yra saugesnis, tačiau apriboja vartotojų sprendžiamų uždavinių pasirinkimo laisvę.

- **SSH raktų naudojimas**

Rekomenduojama naudoti SSH raktus autentifikacijai vietoj slaptažodžio autentifikacijos jungiantis prie nuotolinių SSH serverių.

Iš tikrųjų, kartą SSH viešąjį raktą išsaugojus nuotoliniame serveryje, yra galima autentifikuotis į jį naudojant tiesiogiai susijusį SSH privatų raktą, kuri yra apsaugota slapta fraze.

Tai pagerina saugumą abiejų – vartotojo ir SSH serverio:

Išvengiant realaus slaptažodžio nuskaitymo per tinklą ir išgavimo. SSH privataus rakto slapta frazė yra bevertė be SSH privataus rakto [28].

Tai itin padidina saugumą, tačiau tai paskatina prisirišti vartotoją prie vieno kompiuterio iš kurio jis jungiasi prie GRID.

- **Uždraudimas prisijungti prie ROOT su slaptažodžiu**

Sistemos administratorius turėtų naudoti SSH rakto autentifikaciją vietoj slaptažodžio autentifikacijos ypač ROOT prisijungimui.

Karta šį procesą įvykdžius galima atjungti ROOT prisijungimą su slaptažodžiu. Tai reiškia, kad jei ROOT slaptažodis nenustatytas ar žinomas, vistiek negalimas nuotolinis ROOT prisijungimas [28].

ROOT vartotoją apsaugoti vienas svarbiausių saugumo punktų.

- **Saugumo pataisymų pritaikymas**

Nepavykus ar užmiršus įdiegti saugumo pataisymus yra viena iš pagrindinių saugumo incidentų priežasčių.

Svarbu nuolat tikrinti ir įdiegti saugumo pataisymus į sistemą:

- Gali būti sunku pritaikyti atnaujintus paketus jei sąrašas yra per ilgas.

- Tai gali sutrukdyti blogas žinojimas kaip sistemos silpnybės gali būti išnaudotos blogiems tikslams.

Per paskutiniuosius kelis metus dauguma pikto kodo sumažinta, kai buvo išleista saugumo patarimų ir pataisymų. Šiuo metu vis daugiau pikto kodo programų išleidžiama tą pačią dieną kaip ir patarimai ar pataisymai kaip nuo jų apsisaugoti.

GRID dalyviai turėtų itin stipriai sekti atsirandančius naujausius saugumo pataisymus ir juos diegti į savo sistemas [28].

- **Sistemos kopijų saugojimas (Performing system backups)**

Norint išvengti sunkių sistemos pažeidimo pasekmių ir verslo darbo nutraukimo, sistemos kopijų saugojimas turi itin svarbią reikšmę saugumo incidentų metu. Šios kopijos leidžia tyrinėtoją surasti kada ir kuri dalis failų sistemoje buvo pakeista ir išsaugoti informaciją, kuri buvo prarasta incidento metu.

Patirtis rodo, jog taip pat naudinga ištestuoti ar kopijų sugražinimo procedūros gerai veikia [28].

Taip pat ir vartotojas siūsdamas užduotis į GRID, šių užduočių kopijas galėtų išsisaugoti ir pas save, tai taip pat padėtų pačiam vartotojui išsaugoti savo duomenis, jei klasteryje ar GRID įvyktų incidentas.

- **Nenaudingų servisų uždraudimas ar pašalinimas**

Jei sistema buvo sukompromituota, tai labiausiai tikėtina, kad atakuotojas panaudojo tinklo servisu. Kiekvienas sistemoje veikiantis tinklo servisas yra potencialus taikiny atakuotojui.

Todėl rekomenduojama sumažinti tokių potencialių atakuotojo taikinių skaičių juos uždraudžiant ar išvis pašalinant nereikalingus servisis.

Svarbu pašalinti šiuos nenaudingus servisis ir įsitikinti, kad jie nebus pakartotinai aktyvuoti įvykus klaidai ar kito proceso vykdymo metu [28].

Šis saugumo parametras labai panašus į ugniasienės konfigūracijos parametą.

- **Kopijos (*Backups*)**

Kopijos atlieka svarbų vaidmenį nelaimės padarinių atstatyme, verslo nenutrūkstumui, ir saugumo incidento tyrimui. Taip pat rekomenduojama patikrinti kopijų atstatymo procedūras, ar jos tikrai gali atstatyti sistema į prieš tai buvusią būseną.

Tai leidžia servisų šeimininką būti tikru, kad servisas tikrai gali būti atstatytas bet kuriuo metu [28].

- **Tinklo servिसai**

Netgi jei sistema buvo atidžiai sukonfigūruota, visuomet yra gerai patikrinti kokie servisis klausosi tinklo. Idealiausias būdas būtų atlikti šį testą iš nutolusio kompiuterio, tam, kad pamatyti, kurie servisis prieinami nuotoliniu būdu (remotely) [28].

Šis parametras taip pat tampriai susijęs su ugniasienės konfigūracija.

- **Nuotolinės silpnos vietos (*Remote vulnerabilities*)**

Testuoti sistemos silpnas vietas plėtros ir gamybos fazėse yra itin naudingas dalykas.

Keli silpnų vietų skanuotojai, tokie kaip <http://www.nessus.org/> yra gan lengvai naudojami. Nessus atlieka silpnų vietų testą prieš nutolusias sistemas ir platų servisų diapazoną.

Yra galimybė išjungti tokiu pavojingus testus, kaip Serviso Atjungimas (DoS). Nessus reikia paleisti paskutinius išleistus testus ant galutinai paruoštos įprastam darbui sistemos, tai būtų veiksmingiausias patikrinimas.

Kartais gali būti labai naudinga paleisti Nessus prieš tinklo router'ius [28].

Tokie tikrinimai labai sumažina galimybę įsilaužti į sistemą.

- **Stebėti incidento reakcijos taisykles (*Follow the incident response policies*)**

Kiekvienas neautorizuotas priėjimas, panaudojimas ar pakeitimas sistemoje, ar sistemos naudojimas pažeidžiant egzistuojančias taisykles yra saugumo incidentas.

GRID dalyviai turi pranešti apie saugumo incidentus susijusius su incidentų reakcijos politika.

Nedelsiant reaguojant į saugumo incidentus aprūpinant tinkama ir tikslią informaciją, gali žymiai sumažinti keliamą problemos poveikį servisui [28].

- **Verslo nenutrūkstumumo planas**

Svarbi sistemos dokumentacijos sritis yra verslo nenutrūkstamumo planas.

Šis dokumentas yra dalis sistemos rizikos analizės atsitikus svarbesnėms nelaimėms.

Šio dokumento tikslas yra paruošti detalias procedūras, kurios bus reikalingos naudojant servisus kuomet atsitinka nelaimė. Pavyzdžiui, mašina su šviežiomis kopijomis (backup), kopijų įrašai gali pagelbėti atkuriant servisus kai reikalaujama.

Tobuliausias verslo nenutrūkstamumo planas turi pakankamai informacijos, kad būtų įvykdytas ir nelabai patyrusio personalo.

Tai tikrai nelengva gana gerai aprašyti verslo nenutrūkstamumo planą. Kiekvienas plano aspektas turi būti kruopščiai aprašytas tam, kad galėtų būti panaudotas bet kuriuo metu.

Taigi turėtų būti aprašytas ir atitikti aukštus reikalavimus, kad būtų pilnai efektyvus [28].

Tačiau tokį dokumentą sudaryti itin sudėtinga, kintant GRID paslaugoms.

- **Sistemos dokumentacija**

Rašymas ir tęsimas sistemos dokumentacijos su atnaujinimais yra užsitęsęs uždavinys, kuris leidžia sistemos šeimininkui turėti bendrą supratimą apie sistemą:

- sistemos paskirtis;
- sistemos konfigūracija;
- failų sistemos suplanavimas;
- tinklo konfigūracija, pridedant ugniasienės taisykles;
- įvairios grupės ir vartojai;
- atstatymo procedūros;
- verslo nenutrūkstamumo planas;
- sistemos kopijos (*backup*);
- sąrašas žmonių turinčių privilegijuotą priejimą;
- LOG konfigūracija;
- keitimas valdymo procedūrų;
- sistemos stebėjimas;
- sistemos pataisymai;
- failų sąžiningumo tikrinimas;
- priklausomybė nuo kitų sistemų.

Jei saugumo incidentas įvyksta, turint atnaujintą sistemos dokumentaciją galima stipriai sumažinti incidento pasekmes [28].

- **Host-lygio IDS**

Host lygio Įsibrovimo Aptikimo Sistema (IDS), tokios kaip tripwire dažniausiai drastiškai pagerina saugumo incidento tyrimus.

Iš tikrųjų tokie įrankiai dažniausiai sukuria sąrašą pagrindinių sistemos failų. Tai palengvina patikrinant kas pasikeitė sistemoje tarp dviejų sąrašų.

Pasitikėjimui (saugumui) iširti, Tripwire duomenų bazė turėtų būti išsaugota į tik nuskaitymui skirtą laikmeną, tokią kaip diskelį ar USB laikmeną su fiziniu įrašymo uždraudimu [28].

- **ROOTKIT tikrintojai**

Leidžiant įrankius kurie suranda įdiegtus ROOTKIT, gali padėti sistemos administratoriui atsekti įsibrovimus į jų sistemą.

ROOTKIT – tai rinkinys įrankių naudojamų įsibrovėlio po įsilaužimo į kompiuterinę sistemą. Šie įrankiai padeda atakuotojui paslėpti ir palaikyti savo priėjimą prie sistemos ir panaudoti tai piktavališkiems tikslams.

Chkrootkit yra paplitusi skirta UNIX sistemoms programa, kuri patikrina sistemą ar nėra joje žinomų ROOTKIT. Ji dirba naudodama kelis mechanizmus, įskaitant failų aprašymų palyginimą su žinomais ROOTKIT, ir tikrinant įtariamas veiklas [28].

Tačiau kintant ROOTKIT programoms, tikrintojai nespėja taip greitai atsinaujinti, ir vien šiais tikrintojais negalima visiškai pasikliauti.

- **Tinklo lygio IDS**

Stebint tinklą galima pastebėti pavojaus signalus ir aptikti atakos pėdsakus.

Tinklo lygio IDS analizuoja tinklo apkrovimą realiu laiku tam, kad aptikti įtarius veiksmus. Tai gali sukelti parengtį, kuri leis imtis sistemos savininkui atitinkamų veiksmų. Kaip rezultatas, tokie įrankiai padeda aptinkant ir išvengiant saugumo incidentų.

Tinklo lygio IDS turėtų būti įdiegta kuo arčiau pagrindinio tinklo mazgo, kur apkrovimo lygis aukštas. Taip pat svarbu palaikyti tinklo struktūrą atnaujintą.

Vienas iš populiariausių tinklo lygio IDS yra snort [28].

- **Centralizuotas IDS**

HOST paremta IDS tokia kaip Triplewire yra gana efektyvu kelioms pavienėms sistemoms, bet tai nebeįdomiai skirta ne labai didelio masto sprendimams.

Keli labai pažengę IDS jau yra prieinami, kurie skirti didelio masto kompiuterių grupėms. Šie IDS siūlo host paremtą IDS funkcionalumą paskirstytam host'ų rinkiniui, naudojant centralizuotą servisą, kuris atlieka failų sąžiningumo tikrinimą host paremtų įsibrovimų aptikimui.

Pavyzdžiui, Osiris ir Samhain yra gerai žinomos IDS, kurios gali būti naudojami dideliu skaičiumi sistemų [28].

Siūloma: Tačiau klasterių administratoriai nepublikuoja savo saugumo parametrų, taigi ir skaičiavimams jų išgauti neįmanoma. Tuo tikslu DELPHI programavimo kalba parašėme programą, kuri apskaičiuoja tariamą saugumo lygį atsižvelgdama į administratoriaus nuomonę apie šių parametrų svarbą.

Saugumo programoje apskaičiuoti saugumo lygiui S naudojama ši formulė:

$$S = \frac{\sum_{i=1}^k R_i}{\sum_{j=1}^n T_j}, \quad (16)$$

čia S – saugumo lygis; R_i - i -tasis įvykdytas saugumo kriterijus su atitinkamu svoriu; T_j - visi saugumo kriterijai su atitinkamai parinktais svoriais; k – įvykdytų ar parinktų saugumo kriterijų skaičius; n – visų saugumo kriterijų skaičius.

Saugumo lygiui gauti vartotojas savo nuomone priskiria atitinkamiems T_j saugumo kriterijams svorius nuo 1 iki 10 ir pažymi pas jį klasteryje įvykdytus kriterijus. Programa jo parinktus kriterijų svorius susumuoja ir padalina iš visų kriterijų parinktų svarbos svorių.

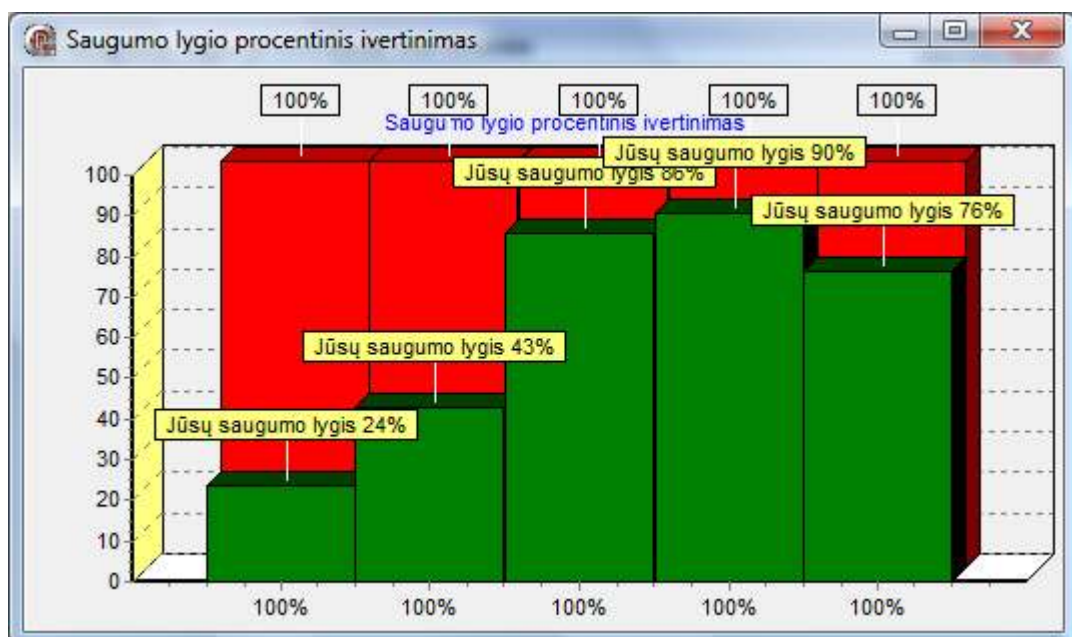
Rezultatai: Saugumo įvertinimą gali atlikti tik pats klasterio administratorius (10 pav.).

GRID saugumo ivertinimas

9	<input checked="" type="checkbox"/>	Centrinis syslog serveris
8	<input type="checkbox"/>	Sistemos metriko
10	<input type="checkbox"/>	Sistemos veikimas, charakteristika (Performance)
10	<input checked="" type="checkbox"/>	Sistemos pataisymu (patching) busena
10	<input checked="" type="checkbox"/>	Blokuoti batch darbus nuo ssh "back doors" kurimo
10	<input type="checkbox"/>	Sistemos ugniasienes konfigravimas
10	<input checked="" type="checkbox"/>	SSH raktu naudojimas
9	<input checked="" type="checkbox"/>	Uzdraudimas prisijungti prie root su slaptazodziu
10	<input checked="" type="checkbox"/>	Saugumo pataisymu pritaikymas
8	<input checked="" type="checkbox"/>	Sistemos kopiju saugojimas (Performing system backups)
7	<input checked="" type="checkbox"/>	Nenaudingu servisu uzdraudimas ar pašalinimas
8	<input checked="" type="checkbox"/>	Kopijos (Backups)
8	<input checked="" type="checkbox"/>	Tinklo servais
10	<input checked="" type="checkbox"/>	Nuotolines silpnos vietos (Remote vulnerabilities)
5	<input type="checkbox"/>	Sekti incidento reakcijos taisykles (Follow the incident response policies)
5	<input type="checkbox"/>	Verslo nenutrukstamumo planas
4	<input checked="" type="checkbox"/>	Sistemos dokumentacija
10	<input type="checkbox"/>	Host-lygio IDS
10	<input type="checkbox"/>	Rootkit tikrintoja
10	<input type="checkbox"/>	Tinklo lygio IDS
10	<input type="checkbox"/>	Centralizuotas IDS

Jūsų saugumo lygis 57%

10 pav. Saugumo įvertinimui sukurtos programos skaičiavimo pavyzdys



11 pav. Saugumo įvertinimui sukurtos programos rezultato pavyzdys

Toliau esantys stulpeliai rodo 100% saugumo lygį, o arčiau esantys stulpeliai - apskaičiuotus rezultatus (11 pav.).

5.5. Vėlinimas

Apibrėžimas: IP paslaugų paketų perdavimo vėlinimas apibrėžiamas kaip laiko skirtumas tarp laiko kai pirmas paketo bitas praeina įėjimo matavimo tašką, ir laiko kai paskutinis bitas praeina išėjimo matavimo tašką kitame gale. Vėlinimo dydis priklauso nuo paketo ilgio, vėlinimo pačiame tinkle, tarpinių interneto tiekėjų, kurių tinklais keliauja paketai perdavimo greičio, geografinio atstumo.

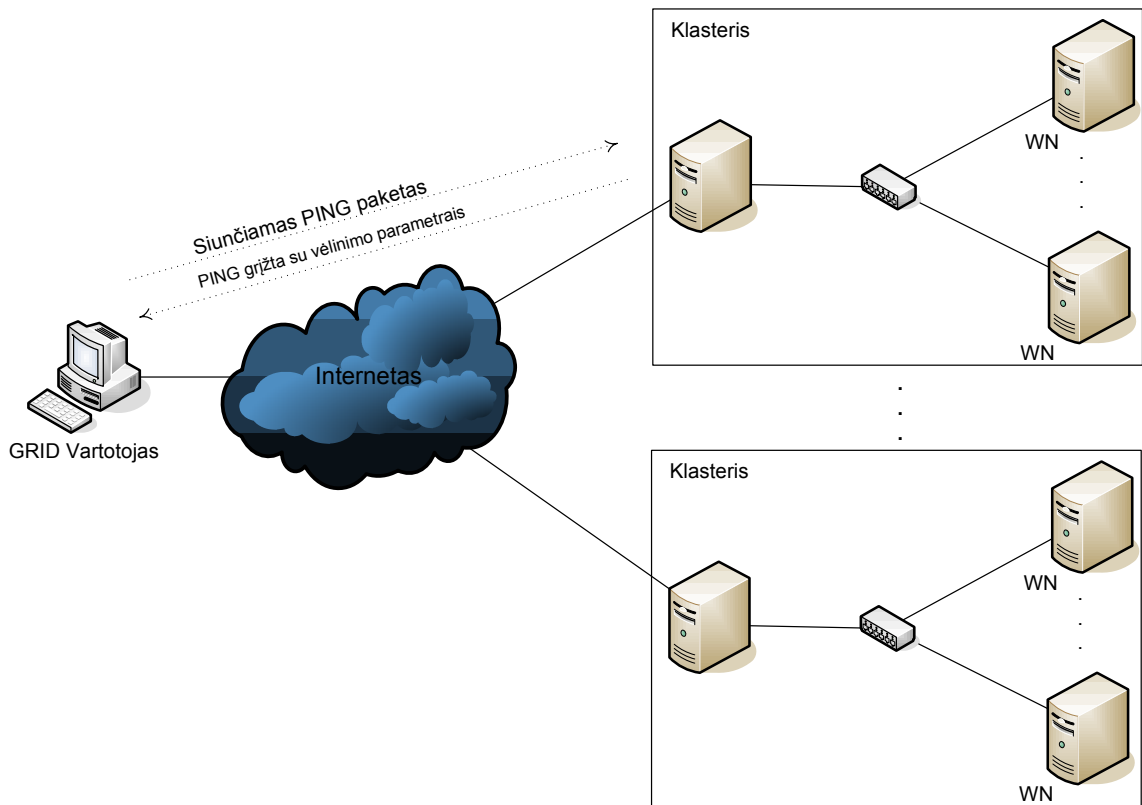
GRID tinkluose vėlinimas analizuojamas įvairiais aspektais: duomenų bazės užklausos laikas, skaičiavimo resursų suradimo ir priskirimo laikas, autentifikavimo laikas ir k.t. Norint gauti išsamius duomenis, rekomenduojama tirti ir jų kombinacijas.

Jautrumas paketų vėlavimui gali būti skirtingas. Vėlinimo laikas labai svarbus realaus laiko servisams. Pernelyg didelis vėlinimas greitai sumažina WEB puslapių ir interaktyvaus garso ir vaizdo paslaugų kokybę.

Siūloma: Vėlinimas gali būti analizuojamas kaip įprastai, tiriant kiek laiko paketas keliauja tinkle. Tačiau GRID tinkluose galima žvelgti į vėlinimą ir kaip į bendresnių užduočių greitį. Pavyzdžiui, duomenų bazės užklausos laikas, skaičiavimo resursų suradimo ir priskirimo laikas, autentifikavimo laikas ir t.t. Be to, galima tirti ir jų kombinacijas.

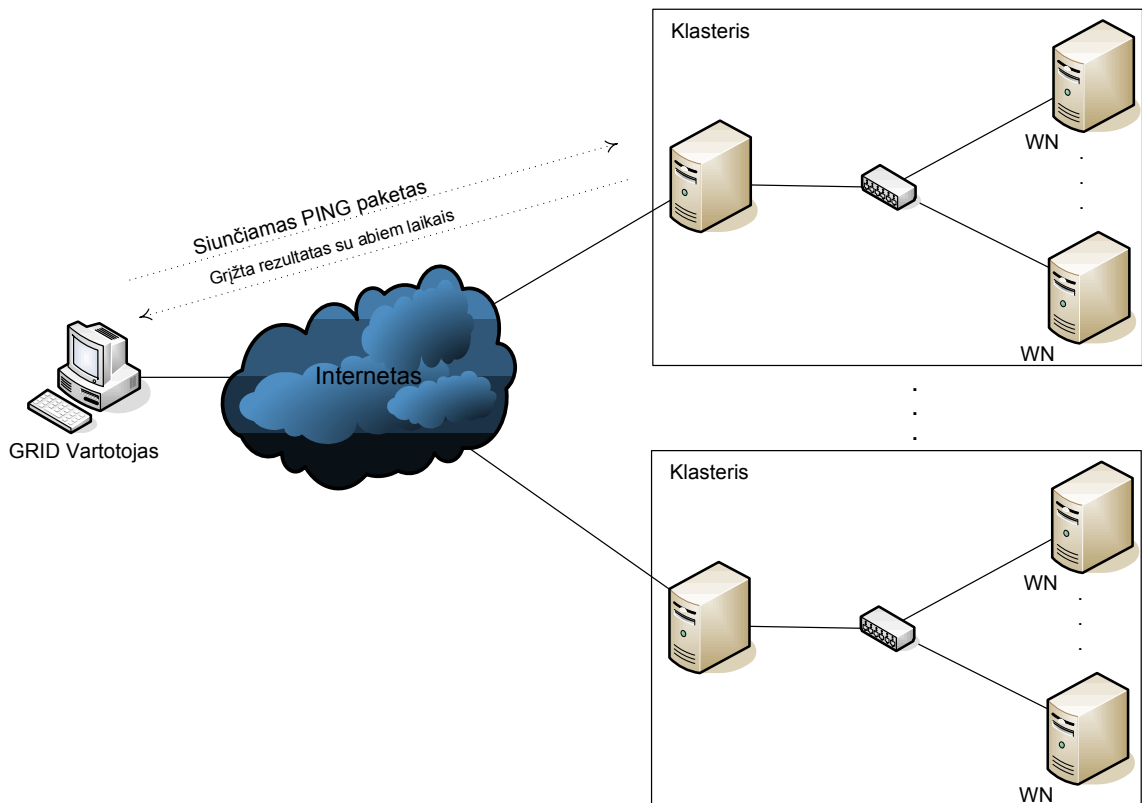
Siūloma tirti 2 tipų vėlinimą. Paketinį vėlinimą ir uždavinių išsprendimo vėlinimą.

Paketiniam vėlinimui siūsti PING paketus, jie grąžina paketo vėlinimo laiką (12 pav.).



12 pav. Siūlomas paketinio vėlinimo skaičiavimo būdas

Uždavinių vėlinimui nuo uždavinio išsiuntimo ir priėmimo į procesorius spręsti sužinoti siūloma siųsti tokį uždavinį, kuris nuskaitytų laiką išsiuntimo metu ir nuskaitytų laiką uždavinio sprendimo metu (13 pav.).



13 pav. Siūlomas uždavinio vėlinimo skaičiavimo būdas

Metodika: tiriama abiejų tipų vėlinimai.

Paketiniam vėlinimui apskaičiuoti panaudojami tie patys duomenys, kurie buvo surinkti pasiekiamumui apskaičiuoti, nes PING rezultatų faile pateikiama informacija ir apie paketinį vėlinimą. Duomenims nuskaityti rekomenduojama panaudoti taip pat tą pačią PHP programą, kuri atrenka vėlinimo duomenis iš failo bei apskaičiuoja atlikto bandymo metu nuskaitytų vėlinimų vidurkį.

Užduoties sprendimo vėlinimas – tam tikrais laiko intervalais siunčiamas uždavinys į GRID tinklą ir stebima, per kiek laiko bus išspręstas uždavinys kiekviename klasteryje. Tam tikslui parašyta programa BASH kalba, kuri patikrina naujausią klasterių sąrašą ir jiems išsiunčia uždavinį, kurio užduotis sužinoti išsiuntimo laiką ir uždavinio sprendimo pražios laiką. Gautiems rezultatams atrinkti ir susisteminti kuriama PHP programa (14 lentelė).

5.6. Vėlinimo fliktuacijos

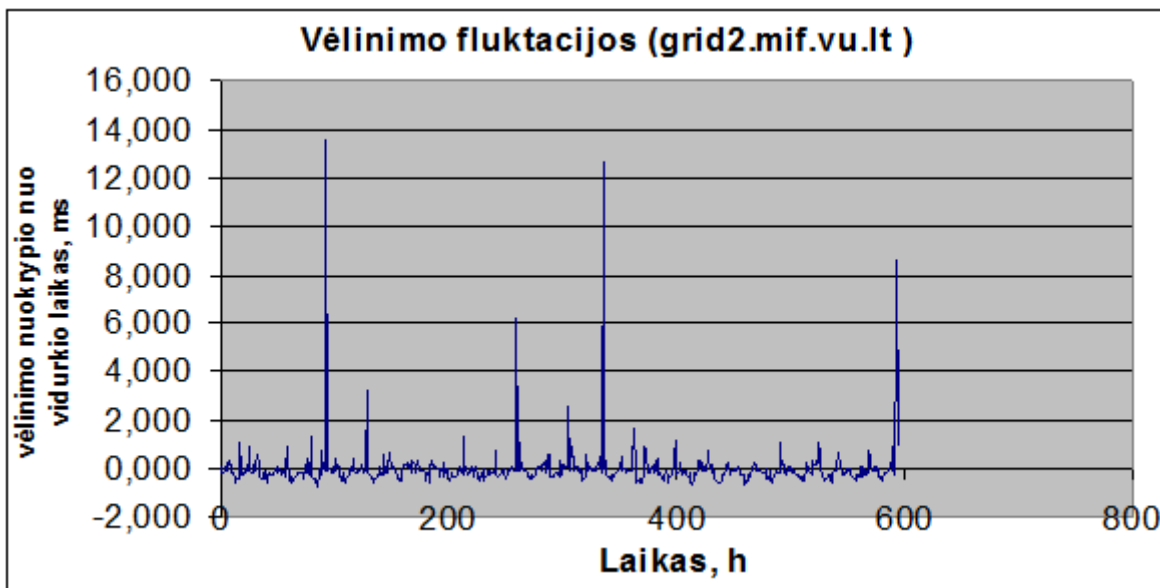
Apibrėžimas: Vėlinimo fliktuacijos yra duomenų atvykimo spartos nukrypimas nuo idealios (vėlinimo nukrypimas). Vėlinimo fliktuacijos visada atsiranda, kai sistemoje yra eilės ir buferizacija. Didžiausios fliktuacijos atsiranda permaršrutizuojant paketus dėl tinklo perpildymo ar gedimo. Fliktuacijų sumažinimui priimančiajame įrenginyje gali būti naudojamas buferizavimas. Tačiau toks buferizavimas sudaro sąlygas vėlinimui, kuris paprastai yra vieną ar du kartus didesnis negu pačios fliktuacijos, todėl tai nėra visais atvejais tinkamas sprendimas.

Siūloma: Fliktuacijų apskaičiavimui rekomenduojama stebėti paketų vėlinimą, kaip ir aprašyta punktu aukščiau, ir vėlinimo fliktuacijas apskaičiuoti iš gautų vėlinimo rezultatų. Iš maksimalaus vėlinimo laiko atėmus minimalų vėlinimo laiką gaunama vėlinimo fliktuacija.

Metodika: Fliktuacijų apskaičiavimui panaudojami tie patys duomenys, kurie buvo surinkti pasiekiamumui apskaičiuoti, nes PING rezultatų faile pateikiama informacija ir apie paketinį vėlinimą. Duomenims nuskaityti rekomenduojama panaudoti taip pat tą pačią PHP programą, kuri atrenka vėlinimo duomenis iš failo bei apskaičiuoja atlikto bandymo metu nuskaitytų vėlinimų fliktuacijas.

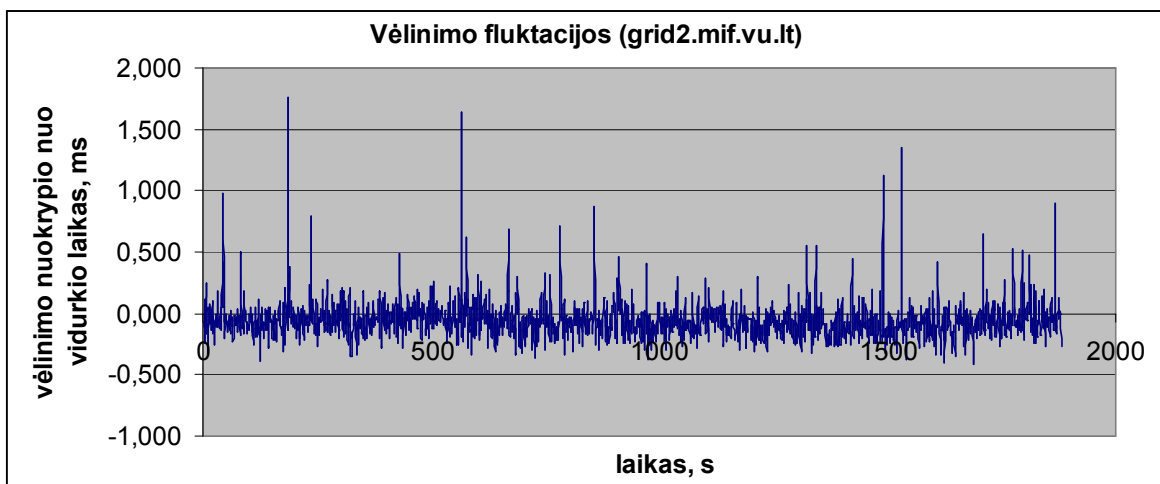
Rezultatai: Su pateikta PHP programa apskaičiuotos fliktuacijos, kurių matavimai buvo atliekami kas valandą (15 lentelė).

Laiko atžvilgiu tikrinant vėlinimą vienos valandos intervalu atsitiktinai pasirinkto klasterio „grid2.mif.vu.lt” fliktuacijos nuokrypis nuo vėlinimo vidurkio atrodo taip kaip pavaizduota grafike žemiau (14 pav.):



14 pav. Paketinio vėlinimo fliktuacijos išsibarstymas apie vėlinimo vidurkį laike, skaičiuojant 1 val. intervalu

To paties klasterio vėlinimą patikrinus 1 sekundės intervalu 1 valandos laike, fliktuacija gavosi 114.907 ms. O vizualiai fliktuacijos pasiskirstymas nuo vėlinimo vidurkio atrodo (15 pav.):



15 pav. Paketinio vėlinimo fliktuacijos išsibarstymas apie vėlinimo vidurkį laike, skaičiuojant 1 s. intervalu

5.7. Pralaidumas

Apibrėžimas: Pralaidumas parodo WEB paslaugos užklausų ar duomenų kiekį per laiko tarpą. Aptarnaujanti sistema turi turėti pakankamai pralaidumo, kad galėtų užtikrinti failų perdavimo servisą, o taip pat palaikyti reikiamą vienalaikių sujungimų skaičių.

Pralaidumas rodo paketų skaičių arba duomenų kiekį baitais per sekundę, kurį paslauga gali palaikyti. Skirtingų paslaugos klasių atveju, pralaidumas taikomas kiekvienai paslaugos klasei ir gali būti skirtingas skirtingoms klasėms. Parametras taip pat priklauso nuo šaltinio adreso ir/arba tikslo adreso (skirtumas tarp lokalaus srauto, didelių atstumų srauto ir pan.). Paslaugos pralaidumas gali priklausyti nuo dienos laiko/savaitės dienos ir pan. arba šio parametro reikšmė turi būti taikoma visada (įskaitant piko valandas), arba gali būti skirtingos reikšmės skirtingiems laikams.

Analogiškai vėlinimui ir pralaidumas gali būti analizuojamas dvejopai. GRID galima pralaidumą vertinti, kaip resursų suradimo skirtingose virtualiose organizacijose greitį, kaip apdorotų duomenų bazės užklausų kiekį per laiko vienetą arba kaip užduočių priskyrimo laisviems resursams greitį.

Siūloma: Siųsti į pasirinktus klasterius įvairaus dydžio failus ir stebėti duomenų perdavimo spartą. Siuntimą pakartoti kelis kartus ir apskaičiuoti vidurkį.

Metodika: Pralaidumui sužinoti siunčiami failai į klasterius, teikiančius duomenų GRID (*Storage Element*) paslaugą, ir stebimas pralaidumo dydis. Šis veiksmas vykdomas iš „pupa.elen.ktu.lt“. Tik skaičiavimo GRID (*Computing Element*) paslaugą teikiančio klasterio pralaidumui sužinoti yra atrinkti fiziškai arčiausiai jų esantys klasteriai ir jiems prilyginamas tas pats pralaidumas. Jei skaičiavimo elementas neturi artimų duomenų elementų, tuomet iki jų pralaidumo sužinoti nepavyko.

Rezultatai: Matavimai buvo atlikti vieną dieną. Padarius duomenų kopiją „pupa.elen.ktu.lt“ klasteryje ir siunčiant ją į visus kitus duomenų GRID paslaugą teikiančius klasterius (16 lentelė ir 17 lentelė). Duomenų failas buvo atsitiktinai sugeneruotų simbolių rinkinys maždaug 700 MB dydžio.

5.8. Laisvi CPU

Apibrėžimas: Šis parametras parodo, kiek klasteris savo kompiuterių (*Working Nodes*) skiria tiriamam GRID. Kiekvienas klasteris gali turėti tam tikrą kiekį kompiuterių, tačiau vienai ar kitai virtualiai organizacijai paskirti tik dalį savo turimų resursų. Tai šie paskirti resursai tiriamos VO (*Virtualios Organizacijos*) GRID ir vadinami laisvais CPU.

Metodika: Yra specialios GRID komandos, kurias įvykdžius sužinoma klasterio būseną, jo dydis, uždavinių skaičius eilėje.

Rezultatai: Aš tiriu Baltic GRID, ir jame įvykdžius komandą „lcg-infosites --vo balticgrid ce“ gaunami rezultatai matomi (18 lentelė).

6. EFEKTYVUMO ĮVERTINIMAS KLASTERIUI GRID KONTEKSTE

GRID klasterių efektyvumą galima vertinti subjektyviais ir objektyviais būdais. Pirmasis pagrįstas tuo, kad atskaitos taškas yra vartotojo pasitenkinimas konkrečiu teikiamu servisu. Tačiau šis būdas dažniausiai nepasiteisina. Čia pateikiamas efektyvumo įvertinimas antruoju būdu, kai vartotojas nusako, kokie parametrai turi būti tenkinami GRID klasteriui, kad galima būtų užtikrinti reikiamą darbų veikimą.

Pastaruoju metodu norint sužinoti norimo GRID ar jo klasterio QoS būtina tuos parametrus įvertinti. Skaitinę išraišką galima gauti pagal sekančią formulę:

$$Q = \frac{\sum_{i=1}^I (K_i \cdot A_i)}{\sum_{i=1}^I A_i}; \quad (17)$$

čia K_i - ankstesniuose paragrafuose aprašytų i -tojo parametro reikšmė. A_i - i -tojo parametro vartotojo užduotas svoris intervale $[1; 10]$, pagal numatomo uždavinio reikalavimus. Kiekvieno parametro reikšmę aprašo sekančios formulės (18-20).

Užsiduodame tiriamam klasteriui minimalius reikalavimus: uždavinio sprendimui pageidaujamas maksimalus vėlinimas V_{max} ir uždavinio sprendimui pageidaujamas maksimalus vėlinimo nuokrypis (fliuktuacija) F_{max} . Jei tiriamo klasterio vėlinimas $V > V_{max}$, tada vėlinimo koeficientas $K_V = 0$. Jei to paties klasterio vėlinimo fliuktuacija $F > F_{max}$, tada fliuktuacijos koeficientas $K_F = 0$.

Pralaidumo iki tiriamojo GRID klasterio koeficientas skaičiuojamas pagal sekančią formulę:

$$K_P = \frac{P}{P_{max}}; \quad (18)$$

čia: P - tiriamo klasterio pralaidumas; P_{max} - maksimalus pralaidumas tiriamame GRID tinkle.

Laisvų CPU koeficientas skaičiuojamas pagal sekančią formulę:

$$K_C = \frac{C}{C_{\max}} \quad (19)$$

čia: C - tiriama klasterio laisvų CPU skaičius jis gali būti tik mažiau ar balygus C_{\max} ; C_{\max} - maksimalus CPU skaičius tiriama GRID tinkle.

Vėlinimo koeficientas apskaičiuojamas sekančiai:

$$K_V = \frac{V_{\max} - V}{V_{\max} - V_{\min}}; \quad (20)$$

čia: V_{\min} - minimalus vėlinimas tiriama GRID; V_{\max} - maksimalus vėlinimas tiriama GRID; V - tiriama klasterio vėlinimas, kuris $V_{\min} \leq V \leq V_{\max}$.

Vėlinimo fliktuacijoms naudojama formulė (21):

$$K_F = \frac{F_{\max} - F}{F_{\max} - F_{\min}}; \quad (21)$$

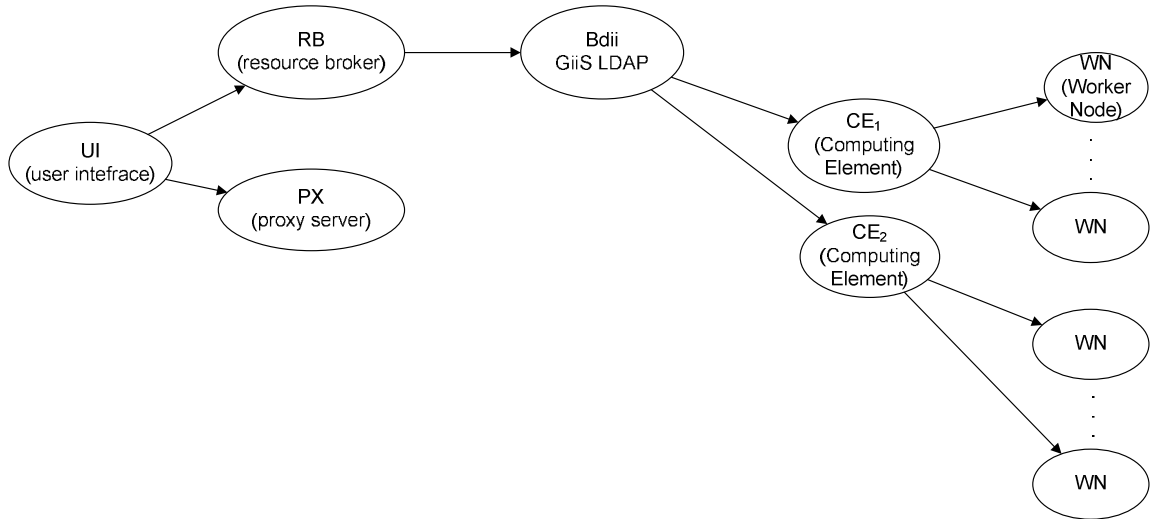
čia: F_{\min} - minimali fliktuacija tiriama GRID; F_{\max} - maksimali vėlinimo fliktuacija tiriama GRID; F - tiriama klasterio vėlinimo fliktuacija, kur $F_{\min} \leq F \leq F_{\max}$.

Uždavinių vėlinimas bei jų vėlinimo fliktuacijos apskaičiuojamas analogiškai pagal formules (20-21).

Viso GRID QoS galima apskaičiuoti įstačius į formulę (17) gautus formulėmis (18-21) rezultatus.

7. BalticGrid VEIKIMO PRINCIPAS

Tyrimai buvo atliekami BalticGRID tinkle, kurio veikimo principas pavaizduotas 16 paveikslėlyje.

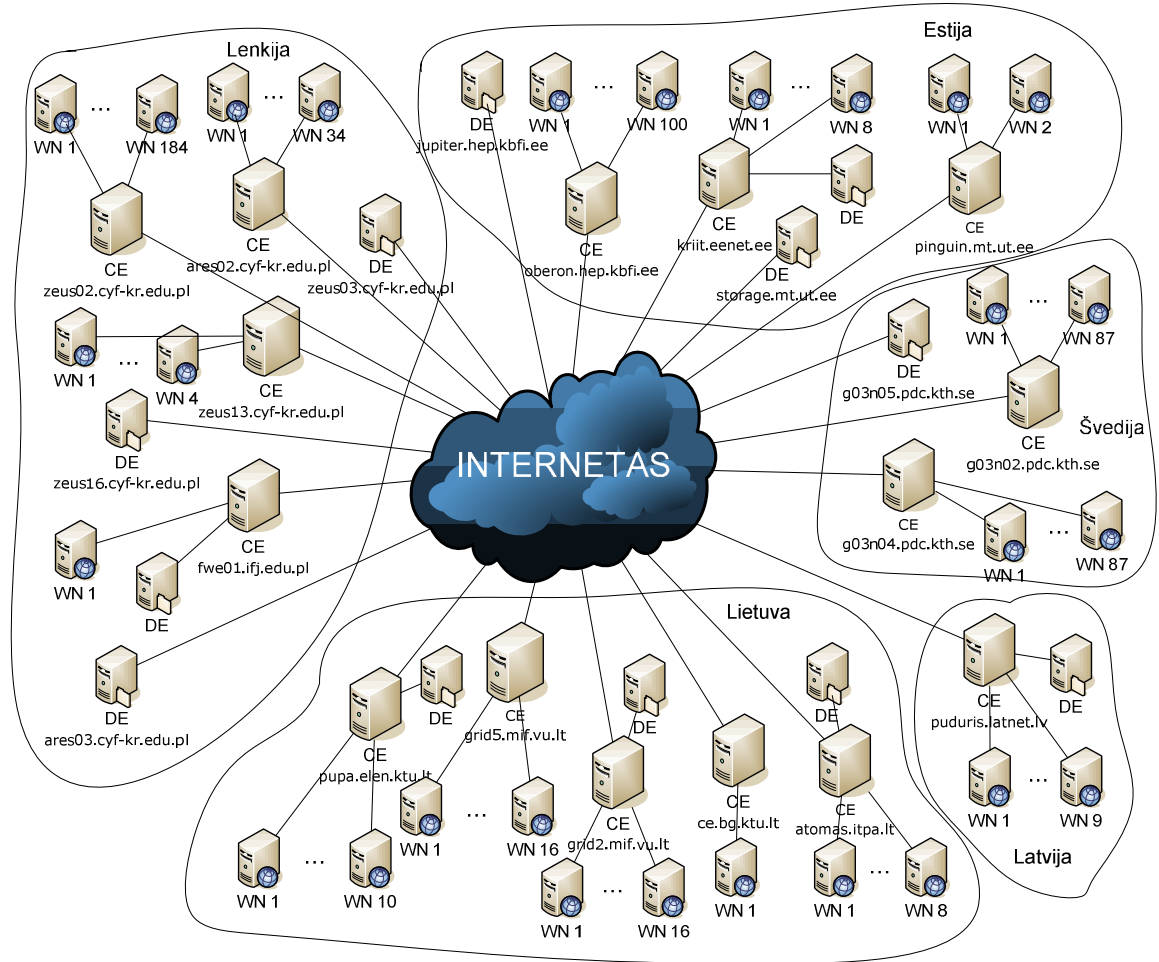


16 pav. BalticGRID middleware veikimo principas

Vartotojas per UI (vartotojo sąsaja) jungiasi prie GRID resursų (16 pav.). UI jungiasi prie RB (resursų brokeris), kad sužinotų, kokių resursų, iš kurio klasterio paimti taip resursų brokeris nukreipia UI į jam tinkamus, ir vartotojo sąsaja nukreipiama į Bdii (*Berkeley Database Information Index*), Bdii patikrina bei identifikuoja vartotoją bei parenka atitinkamus CE (Skaičiavimo elementus), kurie paskirsto skaičiavimus po WN (*Worker Node*). CE yra klasteriai, o WN – tai klasteriui priklausantys skaičiavimams, ar duomenų saugojimui naudojami kompiuteriai.

8. ATLIEKAMŲ TYRIMŲ REZULTATAI

Tyrimams atlikti pasirinktas BalticGrid projektui naudojamas tinklas, kuriame dalyvauja Lietuvos, Lenkijos, Latvijos, Estijos ir Švedijos klasteriai (17 pav.).



17 pav. BalticGrid tinklo topologija

17 paveiksle CE – skaičiavimo elementas (*Computing Element*); DE – duomenų elementas (*Data Element arba Storage Element*); WN – darbinis mazgas (*Working Node*) arba kitaip vadinamas kompiuteris pajungtas į klasterį.

Skaičiavimo elemento galingumą iš dalies galime įvertinti pagal į jį pajungtų kompiuterių (WN) skaičių (6 lentelė).

6 lentelė. Skaičiavimo elementų palyginimas pagal darbinių mazgų skaičių

Skaičiavimo elementas	Skirtu darbo mazgų
g03n04.pdc.kth.se:2119/blah-pbs-balticgrid	0
kriit.eenet.ee:2119/jobmanager-pbs-balticgrid	8
penguin.mt.ut.ee:2119/jobmanager-pbs-balticgrid	16
g03n02.pdc.kth.se:2119/jobmanager-lcgpbs-balticgrid	0
atomas.itpa.lt:2119/jobmanager-lcgpbs-balticgrid	3
puduris.latnet.lv:2119/jobmanager-lcgpbs-balticgrid	7
ce.bg.ktu.lt:2119/jobmanager-lcgpbs-balticgrid	41
fwe01.ifj.edu.pl:2119/jobmanager-lcgpbs-balticgrid	6
grid2.mif.vu.lt:2119/jobmanager-lcgpbs-balticgrid	18
oberon.hep.kbfi.ee:2119/jobmanager-lcgpbs-balticgrid	4
pupa.elen.ktu.lt:2119/jobmanager-lcgpbs-balticgrid	8
ares02.cyf-kr.edu.pl:2119/jobmanager-lcgpbs-balticgrid	22
zeus02.cyf-kr.edu.pl:2119/jobmanager-lcgpbs-balticgrid	5
zeus13.cyf-kr.edu.pl:2119/jobmanager-lcgpbs-balticgrid	-
grid5.mif.vu.lt:2119/blah-pbs-balticgrid	18
fangorn.man.poznan.pl:2119/jobmanager-lcgpbs-balticgrid	-
birzs.latnet.lv:2119/jobmanager-pbs-balticgrid	16
vdupdc.vdu.lt:2119/jobmanager-lcgpbs-balticgrid	21
grid.vtu.lt:2119/jobmanager-lcgpbs-balticgrid	18
ce.egee.man.poznan.pl:2119/jobmanager-lcgpbs-balticgrid	153

Duomenų elemente vienas iš dominančių svarbių parametru yra klasterio talpa (7 lentelė).

7 lentelė. Duomenų elementų palyginimas pagal saugyklai skirtos vietos dydį

Duomenų elementas	Galima vieta (Kb)
kriit.eenet.ee	2031293960
g03n05.pdc.kth.se	68769256
storage.mt.ut.ee	2756477548
atomas.itpa.lt	19681488
puduris.latnet.lv	1094913036
fwe01.ifj.edu.pl	69727816
grid2.mif.vu.lt	155764092
jupiter.hep.kbfi.ee	284000
pupa.elen.ktu.lt	156987392
ares03.cyf-kr.edu.pl	22142028
zeus03.cyf-kr.edu.pl	285707368
zeus16.cyf-kr.edu.pl	13892660
dpm.cyf-kr.edu.pl	16900000000
birzs.latnet.lv	135051252
dpm01.grid.etf.rtu.lv	1015610000
io.hep.kbfi.ee	1
vdupdc.vdu.lt	22516248
se1.egee.man.poznan.pl	35700000

8 lentelė. Skaičiavimo elementų apskaičiuotų pasiekiamumo, prieinamumo, vėlinimo, vėlinimo fliktuacijos ir pralaidumo parametrų lentelė

Skaičiavimo elementas	pasiekiamumas	prieinamumas	vėlinimas, ms	Fliktuacija, ms	Pralaidumas, MB/s
g03n04.pdc.kth.se	0.998324958124	-	37.1246107383	19.916	-
kriit.eenet.ee	0.996649916248	1	16.694	14.585	3
penguin.mt.ut.ee	0.79229480737	0,040323	17.5647674419	21.369	-
g03n02.pdc.kth.se	0.998324958124	0,661157	37.1866895973	19.973	-
atomas.itpa.lt	0.998324958124	0,965909	2.22531375839	8.62	3.5
puduris.latnet.lv	0.996649916248	0,981132	8.76115462185	64.272	2.6
ce.bg.ktu.lt	0.949748743719	0,904762	0.859816578483	13.401	-
fwe01.ifj.edu.pl	0.998324958124	0,395161	23.7873120805	50.25	0.6
grid2.mif.vu.lt	0.996649916248	0,865079	2.84904537815	14.296	3.2
oberon.hep.kbfi.ee	0.998324958124	0,163793	225.915290268s	8996.931	-
Pupa.elen.ktu.lt	1	0,927928	0.306185929648	2.945	8
ares02.cyf-kr.edu.pl	0.996649916248	0,95122	22.5905294118	12.624	2.4
zeus02.cyf-kr.edu.pl	0.996649916248	0,843373	22.7089260504	31.832	2.5
zeus13.cyf-kr.edu.pl	0.996649916248	0	22.6607478992	24.813	-
Grid2.mif.vu.lt	0.996649916248	0,865079	3.46887394958	39.509	-
fangorn.man.poznan.pl	0.998324958124	-	14.5501392617	17.28	3.1
Birzs.latnet.lv	0.996649916248	0,97619	8.40012773109	30.735	2.3
vdupdc.vdu.lt	0.998324958124	0,746032	1.31095973154	8.643	-
grid.vtu.lt	- (uždrausta)	1	- (uždrausta)	- (uždrausta)	-
ce.egee.man.poznan.pl	- (uždrausta)	0,722689	- (uždrausta)	- (uždrausta)	-

9 lentelė. Duomenų elementų apskaičiuotų pasiekiamumo, prieinamumo, vėlinimo, vėlinimo, vėlinimo fliktuacijos ir pralaidumo parametrų lentelė

Duomenų elementas	pasiekiamumas	prieinamumas	vėlinimas, ms	Fliktuacija, ms	pralaidumas
kriit.eenet.ee	0.996649916248	0,648438	37.1246107383	14.585	3 MB/s
g03n05.pdc.kth.se	-	-	-	-	1.6 Mb/s
storage.mt.ut.ee	-	0,0625	-	-	-
atomas.itpa.lt	0.998324958124	0,640625	2.22531375839	8.62	3.5 Mb/s
puduris.latnet.lv	0.996649916248	0,671875	8.76115462185	64.272	2.6 MB/s
fwe01.ifj.edu.pl	0.998324958124	0,632813	23.7873120805	50.25	0.6 MB/s
grid2.mif.vu.lt	0.996649916248	0,757813	2.84904537815	14.296	3.2 MB/s
Jupiter.hep.kbfi.ee	-	-	-	-	-
Pupa.elen.ktu.lt	1	0,59375	0.306185929648	2.945	8MB/s
ares03.cyf-kr.edu.pl	0.996649916248	0,95122	22.5905294118	12.624	2.4 MB/s
zeus03.cyf-kr.edu.pl	0.996649916248	0,726563	22.7089260504	31.832	2.5 MB/s
zeus16.cyf-kr.edu.pl	-	-	-	-	Isjungtas
dpm.cyf-kr.edu.pl	-	0	-	-	-
Birzs.latnet.lv	0.996649916248	0,826923	8.40012773109	30.735	2.3 MB/s
dpm01.grid.etf.rtu.lv	-	0,546875	-	-	2.6 MB/s
io.hep.kbfi.ee	-	0,585938	-	-	-
vdupdc.vdu.lt	0.998324958124	0,59375	1.31095973154	8.643	-
se1.egee.man.poznan.pl	-	0,65625	-	-	3.1 MB/s

GRID tinklo topologija kintanti laike. Taigi topologijos nupiešimo ir pradinių duomenų surinkimo metu topologija gali skirtis nuo tinklo topologijos, kuomet buvo vykdomi parametrų skaičiavimai, todėl parametrų rezultatuose gali būti neapskaičiuotų kai kurių klasterių nei vieno parametro. Tai gali reikšti, kad tyrimo metu klasteris galėjo būti

išjungtas. Taip pat po šių tyrimų galėjo būti prijungta naujų klasterių, kurių parametrai negalėjo būti apskaičiuoti.

10 lentelė. Klasterių QoS skaičiavimas pagal kelis užsidočius parametrus

	Pasiekiamumas	Prieinamumas	Paketinis vėlinimas	Paketinio vėlinimo fliktuacijos	Laisvi CPU	QoS
Svorių koef.	5	5	5	5	5	
kriit.eenet.ee	0,998	0,98	16,694	14,585	1	0,62
atomas.itpa.lt	0,998	0,96	2,222	8,62	7	0,83
puduris.latnet.lv	0,996	0,98	8,761	64,2	5	0,57
fwe01.ifj.edu.pl	0,998	0,39	23,7	50,25	0	0,32
grid2.mif.vu.lt	0,996	0,86	2,84	14,2	0	0,71
pupa.elen.ktu.lt	1	0,92	0,3	2,9	6	0,85
ares02.cyf-kr.edu.pl	0,996	0,92	22,5	12,6	5	0,61
zeus02.cyf-kr.edu.pl	0,996	0,95	22,7	31,8	18	0,70
birzs.latnet.lv	0,996	0,97	8,4	30,7	0	0,63

Klasterių palyginimas pagal QoS pavaizduotas 10 lentelėje. Formulėje parametų svorių koeficientai parinkti visų vienodi, po 5. Kaip matome pagal QoS geriausias klasteris būtų pupa.elen.ktu.lt, nes jo QoS auksčiausias, net 0.85, artimiausias jam pagal QoS klasteris būtų grid2.mif.vu.lt (0.71). pupa.elen.ktu.lt parametrai geriausi dėlto, kad jis randasi tame pačiame potinklyje kaip ir kompiuteris iš kurio buvo atliekami skaičiavimai. Tarkim kaip skaičiavimo uždaviniams kuriems aktualus CPU skaičius visiškai netiktų fwe01.ifj.edu.pl, grid2.mif.vu.lt, birzs.latnet.lv, todėl, kad tyrimo metu BalticGrid tinklui šie klasteriai nebuvo išskyrę nei vieno CPU. Taigi skaičiavimų atlikti ant jų negalima.

IŠVADOS

Šiame darbe buvo apžvelgti kitų GRID tyrinetojų siūlomi metodai vertinti servisų kokybės efektyvumą. Pagrindinis jų trūkumas, atsižvelgiama į tinklo ir serverių kokybės parametrus remiamantis tik vartotojo apklausos empiriniais duomenimis.

Sukurta metodiką, kurios pagalba įmanoma analizuoti GRID servisų kokybę bei atlikti tiek QoS atskirų parametrų skaičiavimus, tiek apskaičiuoti GRID servisų kokybės parametrus ir įvertinti efektyvumą.

Vienas iš esminių sukurtos GRID efektyvumo metodikos privalumų, jog galima atlikti išsamią analizę norimais QoS aspektais, paverčiant analizuojamus GRID parametrus bedimensiniais dydžiais.

Sukurta programa, kurios pagalba galima nustatyti ne tik viso analizuojamo GRID, bet ir atskirų jo klasterių QoS.

Atlikto eksperimento metu pagal empirinius duomenis nustatyta, kad atsitiktinai pasirenkant klasterį (ką daro standartiškai resursų brokeriai), skaičiavimai užtrunka ilgiau apie 27% nei parenkant geriausią klasterį sukurtos metodikos pagalba.

Darbo metu sukaupta medžiaga buvo aprobuota dviejose mokslinėse konferencijose IT 2006 ir IT 2007. Atspausdintas straipsnis konferencinėje medžiagoje IT 2006 bei priimtas kitas straipsnis į tarptautinę konferenciją ITI 2007 Kroatijoje. Ši medžiaga taip pat buvo panaudota ESF projekte (Virtualios mokslo ir studijų organizacijos kūrimas, vystymas ir diegimas GRID technologijų pagrindu. Nr. BPD-182/F450-BPD-182) pravedant mokymus pirmai tislinei grupei. Buvo dėstoma paskaita apie GRID QoS efektyvumo skaičiavimus, bei atliktas laboratorinis ta pačia tema, remiantis šia metodika.

Šią sukurta metodiką numatyta plėtoti ateityje, papildant ją kitais reikalingais aspektais.

LITERATŪRA

- [1] The EuroGrid Project - <http://www.eurogrid.org/>
- [2] P. Bala, J. Pytlinski, M. Nazaruk, BioGRID-An European grid for molecular biology, Proceedings of 11th IEEE International Symposium on High Performance Distributed Computing (HPDC-11), Edinburgh, UK, 2002.
- [3] R. Menday, P. Wieder, GRIP: The Evolution of UNICORE towards a Service-Oriented Grid, Cracow Grid Workshop, October 27 - 29, 2003, Cracow Poland.
- [4] NAREGI Project - <http://www.naregi.org/>
- [5] F. Berman and R. Wolski. The AppLeS Project: A Status Report. 8th NEC Research Symposium, Berlin, Germany, May 1997.
- [6] D. Abramson, J. Giddy, and L. Kotler, High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid?, Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS 2000), May 1-5, 2000, Cancun, Mexico, IEEE CS Press, USA, 2000.
- [7] M.Sato, H. Nakada, S. Sekiguchi, S. Matsuoka, U. Nagashima and H. Takagi, Ninf: A Network Based Information Library for Global World-Wide Computing Infrastructure, Proceedings of the International Conference on High Performance Computing and Networking Europe (HPCN Europe), Vienna, Austria, April 28-30, 1997.
- [8] J. Frey, T. Tannenbaum, I. Foster, M. Livny, and S. Tuecke, Condor-G: A Computation Management Agent for Multi-Institutional Grids, Proceedings of the Tenth IEEE Symposium on High Performance Distributed Computing (HPDC10) San Francisco, California, August 7-9, 2001.
- [9] S. Venugopal, R. Buyya and L. Winton, A Grid Service Broker for Scheduling Distributed Data-Oriented Applications on Global Grids, Technical Report, GRIDS-TR-2004-1, Grid Computing and Distributed Systems Laboratory, University of Melbourne, Australia, February 2004.
- [10] T. Hey and A. E. Trefethen, The UK e-Science Core Programme and the Grid, Future Generation Computer Systems, Volume 18, Issue 8, October 2002, Pages 1017-1031.
- [11] W. Hoschek , J. Jaen-Martinez, A. Samar , H. Stockinger , K. Stockinger, Data Management in an International Data Grid Project, Proceedings of the first IEEE/ACM International Workshop on Grid Computing, (Springer Verlag Press, Germany), India, 2000.

- [12] Grid Physics Network (GriPhyN).<http://www.griphyn.org/> (Accessed Jun 2004)
- [13] W. Johnston, D. Gannon, and B. Nitzberg. Grids as production computing environments: The engineering aspects of NASA's Information Power Grid. In Proc. Eighth IEEE International Symposium on High Performance Distributed Computing, Redondo Beach, CA, USA, 1999
- [14] A. Natrajan, M. Crowley, N. Wilkins-Diehr, M. Humphrey, A. D. Fox, A. Grimshaw, C. L. Brooks III, Studying Protein Folding on the Grid: Experiences using CHARMM on NPACI Resources under Legion, Proceedings of the 10th International Symposium on High Performance Distributed Computing (HPDC-10), August 7-9, 2001. 19
- [15] N. Beekwilder, A. Grimshaw, Parallelization of an Axially Symmetric Steady Flow Program, CS Technical Report CS-98-10, University of Virginia, USA, May 29, 1998.
- [16] NC-BioGrid - <http://www.ncbiogrid.org/>
- [17] D. Abramson, Nimrod on Legion Middleware, <http://www.dstc.edu.au/Research/activesheets-ov.html>
- [18] A. Grimshaw, A Worldwide Virtual Computer for an Advancing Legion of Applications, NPACI, <http://www.npaci.edu/enVision/v15.2/legion.html>
- [19] J. Rahman and G. Davis, TIME and DIME: Supporting Natural Resource Modelling with .NET and Alchemi, CSIRO, <http://alchemi.net/projects.html>
- [20] A. Warmenhoven, Amsterdam PrivateGrid, The Netherlands, <http://www.private-grid.nl/>
- [21] R. Buyya, S. Date, Y. Mizuno-Matsumoto, S. Venugopal, and D. Abramson, Composition of Distributed Brain Activity Analysis and its On-Demand Deployment on Global Grids, New Frontiers in High-Performance Computing: Proceedings of the 10th International Conference on High Performance Computing (HiPC 2003) Workshops (Dec. 17, 2003, Hyderabad, India), ISBN: 81-88901-05-9, Elite Publishing House, New Delhi, India.
- [22] BADG Project, Belle Analysis Data Grid, <http://epp.ph.unimelb.edu.au/epp/grid/badg/>
- [23] B. Beeson, S. Melnikoff, S. Venugopal, and D. Barnes, A Portal for Grid-enabled Physics, Australian Virtual Observatory, <http://www.aus-vo.org/>.

- [24] B. Hughes, S. Venugopal, R. Buyya, Grid-based Indexing of a Newswire Corpus, Technical Report, GRIDSTR- 2004-4, Grid Computing and Distributed Systems Laboratory, University of Melbourne, Australia, June/July 2004.
- [25] Ian Foster, Carl Kesselman and Steven Tuecke, “The Anatomy of the Grid: enabling scalable virtual organizations” IJS, 2001. [ZJ. Ian Foster, “What is The Grid? A Three Point [http://www-fn.nics.anl.gov/~foster](http://www.fn.nics.anl.gov/~foster)
- [26] Vler Sander, William A. Adamson, Ian Foster, “End-to-End Provision of Policy Information for Network QoS”, Proceedings of the Tenth IEEE Symposium on HPDC, August 2001
- [27] BalticGrid. Prieiga internetu: <https://lxb6112.cern.ch/sam/sam.py> .[žiūrėta 2006.10.15].
- [28] GRID operations centre. Prieiga internetu: http://goc.grid-support.ac.uk/gridsite/operations/security_info.php . [žiūrėta 2006.11.15].
- [29] Oracle 10g Grid Computing. Prieiga internetu: http://www.remote-dba.cc/grid_13.htm. [žiūrėta 2007.01.20].
- [30] M. F. Maldonado, „Grid computing in higher education: Trends, values and offerings. education on demand“, December 2004. http://www-1.ibm.com/linux/grid/pdf/grid_computing_in_higher_ed.pdf
- [31] A. Konstantinov „Pagrindiniai Grid principai ir NorduGrid ARC pristatymas“, http://www.nordugrid.org/tutorial/Vilnius_2005_03_17/Tutorial_lt.pdf
- [32] R. Al-Ali, G. von Laszewski, K. Amin, M.I Hategan, „QoS Support for High-Performance Scientific Grid Applications“, <http://www-unix.mcs.anl.gov/~laszewsk/papers/vonLaszewski-qos-ccgrid04.pdf>
- [33] A. Mani, A. Nagarajan, „Understanding quality of service for Web services“, 01 Jan 2002, <http://www-128.ibm.com/developerworks/webservices/library/ws-quality.html>
- [34] D. A. Menascé, E. Casalicchio, “QoS in GRID Computing” <http://www.computer.org/internet/>

SUTRUMPINIMAI

GRID – (*Global Resource Information Database*) globali resursų informacijos duomenų bazė;

QoS – (*Quality of Service*) servisų kokybė;

VO – (*Virtual Organization*) virtuali organizacija;

IT – informacinės technologijos;

IPT – internetinių paslaugų tiekėjas;

CPU – (*Central Processing Unit*) Centrinio procesoriaus vienetas;

SLA – (*Service Level Agreement*) servisų lygio sutartis;

WSDL – (*Web Services Description Language*) WEB swervisų aprašymo kalba;

QoP – (*multimedia QoS*) multimedia servisų kokybė;

API – (*Application Programming Interface*) programinės įrangos programavimo sąsaja;

IP – (*Internet Protocol*) interneto protokolas;

SE – (*Storage Element*) duomenų elementas;

CE – (*Computing Element*) skaičiavimo elementas;

WN – (*Working Node*) darbinis mazgas;

SSH – (*Secure Shell*) saugus prisijungimas prie shell;

IDS – įsibrovimo aptikimo sistema,

UI – (*User Interface*) vartotojo sąsaja;

RB – (*Resource Broker*) resursų brokeris;

Bdii – (*Berkeley Database Information Index*) indeksuota informacijos duomenų bazė;

PRIEDAI

11 lentelė. Programos apskaičiuotų pasiekiamumo parametro lentelė

Skaičiavimo elementas	pasiekiamumas
g03n04.pdc.kth.se	0.998324958124
kriit.eenet.ee	0.996649916248
pinguin.mt.ut.ee	0.79229480737
g03n02.pdc.kth.se	0.998324958124
atomas.itpa.lt	0.998324958124
puduris.latnet.lv	0.996649916248
ce.bg.ktu.lt	0.949748743719
fwe01.ifj.edu.pl	0.998324958124
grid2.mif.vu.lt	0.996649916248
oberon.hep.kbfi.ee	0.998324958124
pupa.elen.ktu.lt	1
ares02.cyf-kr.edu.pl	0.996649916248
zeus02.cyf-kr.edu.pl	0.996649916248
zeus13.cyf-kr.edu.pl	0.996649916248
Grid2.mif.vu.lt	0.996649916248
fangorn.man.poznan.pl	0.998324958124
birzs.latnet.lv	0.996649916248
vdupdc.vdu.lt	0.998324958124
grid.vtu.lt	- (uždrausta)
ce.egee.man.poznan.pl	- (uždrausta)

12 lentelė. Skaičiavimo elementų apskaičiuoto prieinamumo parametro lentelė

Skaičiavimo elementas	prieinamumas
g03n04.pdc.kth.se	-
kriit.eenet.ee	1
pinguin.mt.ut.ee	0,040323
g03n02.pdc.kth.se	0,661157
atomas.itpa.lt	0,965909
puduris.latnet.lv	0,981132
ce.bg.ktu.lt	0,904762
fwe01.ifj.edu.pl	0,395161
grid2.mif.vu.lt	0,865079
oberon.hep.kbfi.ee	0,163793
pupa.elen.ktu.lt	0,927928
ares02.cyf-kr.edu.pl	0,95122
zeus02.cyf-kr.edu.pl	0,843373
zeus13.cyf-kr.edu.pl	0
Grid2.mif.vu.lt	0,865079
fangorn.man.poznan.pl	-
birzs.latnet.lv	0,97619
vdupdc.vdu.lt	0,746032
grid.vtu.lt	1
ce.egee.man.poznan.pl	0,722689

13 lentelė. Duomenų elementų apskaičiuoto prieinamumo parametro lentelė

Duomenų elementas	prieinamumas
kriit.eenet.ee	0,648438
g03n05.pdc.kth.se	-
storage.mt.ut.ee	0,0625
atomas.itpa.lt	0,640625
puduris.latnet.lv	0,671875
fwe01.ifj.edu.pl	0,632813
grid2.mif.vu.lt	0,757813
Jupiter.hep.kbfi.ee	-
pupa.elen.ktu.lt	0,59375
ares03.cyf-kr.edu.pl	0,95122
zeus03.cyf-kr.edu.pl	0,726563
zeus16.cyf-kr.edu.pl	-
dpm.cyf-kr.edu.pl	0
birzs.latnet.lv	0,826923
dpm01.grid.etf.rtu.lv	0,546875
io.hep.kbfi.ee	0,585938
vdupdc.vdu.lt	0,59375
se1.egee.man.poznan.pl	0,65625

14 lentelė. Klasterių apskaičiuoto vėlinimo parametro lentelė

Skaičiavimo elementas	Pakietinis vėlinimas, ms	Uždavinių vėlinimas, s
g03n04.pdc.kth.se	37.1246107383	
kriit.eenet.ee	16.694	30
penguin.mt.ut.ee	17.5647674419	
g03n02.pdc.kth.se	37.1866895973	6346
atomas.itpa.lt	2.22531375839	75
puduris.latnet.lv	8.76115462185	
ce.bg.ktu.lt	0.859816578483	63
fwe01.ifj.edu.pl	23.7873120805	
grid2.mif.vu.lt	2.84904537815	
oberon.hep.kbfi.ee	225.915290268s	13045
pupa.elen.ktu.lt	0.306185929648	103
ares02.cyf-kr.edu.pl	22.5905294118	3513
zeus02.cyf-kr.edu.pl	22.7089260504	3449
zeus13.cyf-kr.edu.pl	22.6607478992	
Grid2.mif.vu.lt	3.46887394958	121
fangorn.man.poznan.pl	14.5501392617	
birzs.latnet.lv	8.40012773109	
vdupdc.vdu.lt	1.31095973154	47
grid.vtu.lt	- (uždrausta)	96
ce.egee.man.poznan.pl	- (uždrausta)	

15 lentelė. Klasterių apskaičiuoto vėlinimo fliktuacijų parametro lentelė

Klasteris	Paketinė fliktuacija, ms	Uždavinių fliktuacija, ms
g03n04.pdc.kth.se	19.916	
kriit.eenet.ee	14.585	7
penguin.mt.ut.ee	21.369	
g03n02.pdc.kth.se	19.973	8329
atomas.itpa.lt	8.62	30
puduris.latnet.lv	64.272	
ce.bg.ktu.lt	13.401	88
fwe01.ifj.edu.pl	50.25	
grid2.mif.vu.lt	14.296	118
oberon.hep.kbfi.ee	8996.931	11690
pupa.elen.ktu.lt	2.945	31
ares02.cyf-kr.edu.pl	12.624	24
zeus02.cyf-kr.edu.pl	31.832	194
zeus13.cyf-kr.edu.pl	24.813	
grid2.mif.vu.lt	39.509	
fangorn.man.poznan.pl	17.28	
birzs.latnet.lv	30.735	
vdupdc.vdu.lt	8.643	8
grid.vtu.lt	- (uždrausta)	81
ce.egee.man.poznan.pl	- (uždrausta)	

16 lentelė. Skaičiavimo elementų pralaidumo parametro lentelė

Skaičiavimo elementas	pralaidumas
g03n04.pdc.kth.se	-
kriit.eenet.ee	3 MB/s
penguin.mt.ut.ee	-
g03n02.pdc.kth.se	-
atomas.itpa.lt	3.5 Mb/s
puduris.latnet.lv	2.6 MB/s
ce.bg.ktu.lt	-
fwe01.ifj.edu.pl	0.6 MB/s
grid2.mif.vu.lt	3.2 MB/s
oberon.hep.kbfi.ee	-
pupa.elen.ktu.lt	8MB/s
ares02.cyf-kr.edu.pl	2.4 MB/s
zeus02.cyf-kr.edu.pl	2.5 MB/s
zeus13.cyf-kr.edu.pl	-
Grid2.mif.vu.lt	-
fangorn.man.poznan.pl	3.1 MB/s
birzs.latnet.lv	2.3 MB/s
vdupdc.vdu.lt	-
grid.vtu.lt	-
ce.egee.man.poznan.pl	-

17 lentelė. Duomenų elementų pralaidumo parametro lentelė

Duomenų elementas	pralaidumas
kriit.eenet.ee	3 MB/s
g03n05.pdc.kth.se	1.6 Mb/s
storage.mt.ut.ee	-
atomas.itpa.lt	3.5 Mb/s
puduris.latnet.lv	2.6 MB/s
fwe01.ifj.edu.pl	0.6 MB/s
grid2.mif.vu.lt	3.2 MB/s
Jupiter.hep.kbfi.ee	-
pupa.elen.ktu.lt	8MB/s
ares03.cyf-kr.edu.pl	2.4 MB/s
zeus03.cyf-kr.edu.pl	2.5 MB/s
zeus16.cyf-kr.edu.pl	Išjungtas
dpm.cyf-kr.edu.pl	-
birzs.latnet.lv	2.3 MB/s
dpm01.grid.etf.rtu.lv	2.6 MB/s
io.hep.kbfi.ee	-
vdupdc.vdu.lt	-
se1.egee.man.poznan.pl	3.1 MB/s

18 lentelė. Laisvi CPU

Klasterio pavadinimas	CPU	Laisvi CPU
ares02.cyf-kr.edu.pl	44	5
zeus02.cyf-kr.edu.pl	274	18
kriit.eenet.ee	6	1
grid.fi.lt	2	2
fwe01.ifj.edu.pl	0	0
birzs.latnet.lv	0	0
puduris.latnet.lv	9	5
atomas.itpa.lt	8	7
grid.pri.kmu.lt	12	12
ce.bg.ktu.lt	44	43
pupa.elen.ktu.lt	10	6
dane.ku.lt	2	2
grid.mii.lt	6	6
g03n02.pdc.kth.se	84	73
egee-ce01.pdc.kth.se	84	73
g03n04.pdc.kth.se	84	73
ce01.grid.etf.rtu.lv	20	18
oberon.hep.kbfi.ee	100	16
penguin.mt.ut.ee	16	16
arvutaja.physic.ut.ee	2	2
vdupdc.vdu.lt	18	18
grid.vtu.lt	19	15
grid6.mif.vu.lt	108	105
grid2.mif.vu.lt	21	0
grid5.mif.vu.lt	21	0
ce.egee.man.poznan.pl	0	0

Uždavinių vėlinimo rezultatų pavyzdys:

Failas **rezultatai.txt**

Acceptable sites:

```
"ares02.cyf-kr.edu.pl:2119/jobmanager-lcgpbs-balticgrid"  
"arvutaja.physic.ut.ee:2119/jobmanager-pbs-balticgrid"  
"ce.bg.ktu.lt:2119/jobmanager-lcgpbs-balticgrid"  
"ce01.grid.etf.rtu.lv:2119/jobmanager-pbs-balticgrid"  
"dane.ku.lt:2119/jobmanager-lcgpbs-balticgrid"  
"fangorn.man.poznan.pl:2119/jobmanager-lcgpbs-balticgrid"  
"grid.fi.lt:2119/jobmanager-lcgpbs-balticgrid"  
"grid.mii.lt:2119/jobmanager-lcgpbs-balticgrid"  
"grid.pri.kmu.lt:2119/jobmanager-lcgpbs-balticgrid"  
"grid.vtu.lt:2119/jobmanager-lcgpbs-balticgrid"  
"grid2.mif.vu.lt:2119/jobmanager-lcgpbs-balticgrid"  
"grid5.mif.vu.lt:2119/blah-pbs-balticgrid"  
"kriit.eenet.ee:2119/jobmanager-pbs-balticgrid"  
"puduris.latnet.lv:2119/jobmanager-lcgpbs-balticgrid"  
"vdupdc.vdu.lt:2119/jobmanager-lcgpbs-balticgrid"  
"zeus02.cyf-kr.edu.pl:2119/jobmanager-pbs-balticgrid"  
"oberon.hep.kbfi.ee:2119/jobmanager-lcgpbs-balticgrid"  
"pupa.elen.ktu.lt:2119/jobmanager-lcgpbs-balticgrid"  
"fwe01.ifj.edu.pl:2119/jobmanager-lcgpbs-balticgrid"  
"atomas.itpa.lt:2119/jobmanager-lcgpbs-balticgrid"  
"ce.egee.man.poznan.pl:2119/jobmanager-lcgpbs-balticgrid"  
"g03n02.pdc.kth.se:2119/jobmanager-pbs-balticgrid"  
"grid.su.lt:2119/jobmanager-lcgpbs-balticgrid"
```

Job submission...

```
Submit to ares02.cyf-kr.edu.pl:2119/jobmanager-lcgpbs-balticgrid ....  
Fri Mar 9 16:38:06 EET 2007
```

Selected Virtual Organisation name (from proxy certificate extension): balticgrid

Connecting to host grid4.mif.vu.lt, port 7772

Logging to host grid4.mif.vu.lt, port 9002

```
===== glite-job-submit Success  
=====
```

The job has been successfully submitted to the Network Server.

Use glite-job-status command to check job current status. Your job identifier is:

- <https://grid4.mif.vu.lt:9000/202gv1j5PICP6T89jxWLWA>

The job identifier has been saved in the following file:

/home/donce/public_html/PC1/uzd/check.jobs

```
=====
```

Submit to arvutaja.physic.ut.ee:2119/jobmanager-pbs-balticgrid
Fri Mar 9 16:38:10 EET 2007

Selected Virtual Organisation name (from proxy certificate extension): balticgrid
Connecting to host grid4.mif.vu.lt, port 7772
Logging to host grid4.mif.vu.lt, port 9002

=====
===== glite-job-submit Success
=====

The job has been successfully submitted to the Network Server.
Use glite-job-status command to check job current status. Your job identifier is:

- https://grid4.mif.vu.lt:9000/AuGU6_zabWh9OmXLFsWKRg

The job identifier has been saved in the following file:
/home/donce/public_html/PC1/uzd/check.jobs

=====
=====

Submit to ce.bg.ktu.lt:2119/jobmanager-lcgpbs-balticgrid
Fri Mar 9 16:38:14 EET 2007

Selected Virtual Organisation name (from proxy certificate extension): balticgrid
Connecting to host grid4.mif.vu.lt, port 7772
Logging to host grid4.mif.vu.lt, port 9002

=====
===== glite-job-submit Success
=====

The job has been successfully submitted to the Network Server.
Use glite-job-status command to check job current status. Your job identifier is:

- <https://grid4.mif.vu.lt:9000/w4FIL-IkSXfBTwc5dCRrpQ>

The job identifier has been saved in the following file:
/home/donce/public_html/PC1/uzd/check.jobs

=====
=====

Submit to ce01.grid.etf.rtu.lv:2119/jobmanager-pbs-balticgrid
Fri Mar 9 16:38:17 EET 2007

Selected Virtual Organisation name (from proxy certificate extension): balticgrid
Connecting to host grid4.mif.vu.lt, port 7772
Logging to host grid4.mif.vu.lt, port 9002

=====
===== glite-job-submit Success
=====

The job has been successfully submitted to the Network Server.
Use glite-job-status command to check job current status. Your job identifier is:

- https://grid4.mif.vu.lt:9000/NIC1Qr2kewcaqPorcki00w

The job identifier has been saved in the following file:

/home/donce/public_html/PC1/uzd/check.jobs

=====
=====

Submit to dane.ku.lt:2119/jobmanager-lcgpbs-balticgrid

Fri Mar 9 16:38:21 EET 2007

Selected Virtual Organisation name (from proxy certificate extension): balticgrid

Connecting to host grid4.mif.vu.lt, port 7772

Logging to host grid4.mif.vu.lt, port 9002

=====
===== glite-job-submit Success

=====
=====

The job has been successfully submitted to the Network Server.

Use glite-job-status command to check job current status. Your job identifier is:

- https://grid4.mif.vu.lt:9000/wZS5cHnX-lrw-y-YYUMGZQ

The job identifier has been saved in the following file:

/home/donce/public_html/PC1/uzd/check.jobs

=====
=====

Submit to fangorn.man.poznan.pl:2119/jobmanager-lcgpbs-balticgrid

Fri Mar 9 16:38:25 EET 2007

Selected Virtual Organisation name (from proxy certificate extension): balticgrid

Connecting to host grid4.mif.vu.lt, port 7772

Logging to host grid4.mif.vu.lt, port 9002

=====
===== glite-job-submit Success

=====
=====

The job has been successfully submitted to the Network Server.

Use glite-job-status command to check job current status. Your job identifier is:

- https://grid4.mif.vu.lt:9000/4wWJyzSG8ywgGOQkFD5rrw

The job identifier has been saved in the following file:

/home/donce/public_html/PC1/uzd/check.jobs

=====
=====

Submit to grid.fi.lt:2119/jobmanager-lcgpbs-balticgrid

Fri Mar 9 16:38:29 EET 2007

Selected Virtual Organisation name (from proxy certificate extension): balticgrid
Connecting to host grid4.mif.vu.lt, port 7772
Logging to host grid4.mif.vu.lt, port 9002

=====
=====
===== glite-job-submit Success
=====

The job has been successfully submitted to the Network Server.
Use glite-job-status command to check job current status. Your job identifier is:

- <https://grid4.mif.vu.lt:9000/iGcPcbSuVgxHdEDzYgvIMg>

The job identifier has been saved in the following file:
/home/donce/public_html/PC1/uzd/check.jobs

=====
=====

Submit to grid.mii.lt:2119/jobmanager-lcgpbs-balticgrid
Fri Mar 9 16:38:32 EET 2007

Selected Virtual Organisation name (from proxy certificate extension): balticgrid
Connecting to host grid4.mif.vu.lt, port 7772
Logging to host grid4.mif.vu.lt, port 9002

=====
=====
===== glite-job-submit Success
=====

The job has been successfully submitted to the Network Server.
Use glite-job-status command to check job current status. Your job identifier is:

- <https://grid4.mif.vu.lt:9000/EQjZrTH3cjrnxRCBRvA5rA>

The job identifier has been saved in the following file:
/home/donce/public_html/PC1/uzd/check.jobs

=====
=====

Submit to grid.pri.kmu.lt:2119/jobmanager-lcgpbs-balticgrid
Fri Mar 9 16:38:37 EET 2007

Selected Virtual Organisation name (from proxy certificate extension): balticgrid
Connecting to host grid4.mif.vu.lt, port 7772
Logging to host grid4.mif.vu.lt, port 9002

=====
=====
===== glite-job-submit Success
=====

The job has been successfully submitted to the Network Server.
Use glite-job-status command to check job current status. Your job identifier is:

- <https://grid4.mif.vu.lt:9000/6oocBrJC4V00I32AVooNPw>

The job identifier has been saved in the following file:

/home/donce/public_html/PC1/uzd/check.jobs

=====
=====

Submit to grid.vtu.lt:2119/jobmanager-lcgpbs-balticgrid

Fri Mar 9 16:38:40 EET 2007

Selected Virtual Organisation name (from proxy certificate extension): balticgrid

Connecting to host grid4.mif.vu.lt, port 7772

Logging to host grid4.mif.vu.lt, port 9002

=====
===== glite-job-submit Success

=====
=====

The job has been successfully submitted to the Network Server.

Use glite-job-status command to check job current status. Your job identifier is:

- <https://grid4.mif.vu.lt:9000/PLWxnpFMbDCOxyZGnSH1IA>

The job identifier has been saved in the following file:

/home/donce/public_html/PC1/uzd/check.jobs

=====
=====

Submit to grid2.mif.vu.lt:2119/jobmanager-lcgpbs-balticgrid

Fri Mar 9 16:38:44 EET 2007

Gryzusių uždavinių rezultatai:

Fri Mar 9 15:39:28 CET 2007

SITE: CYFRONET-IA64

GIIS: ares02.cyf-kr.edu.pl

LOG: ares02.cyf-kr.edu.pl

R marts 9 16:38:39 EET 2007

SITE: UT_PHYSIC

GIIS: arvutaja.physic.ut.ee

LOG: arvutaja.physic.ut.ee

Fri Mar 9 16:40:01 EET 2007

SITE: MII-LCG2

GIIS: grid.mii.lt

LOG: grid.mii.lt

Fri Mar 9 16:38:37 EET 2007

SITE: RTUETF

GIIS: ce01.grid.etf.rtu.lv

LOG: ce01.grid.etf.rtu.lv

Fri Mar 9 16:38:34 EET 2007

SITE: VGTU-gLite
 GIIS: grid.vtu.lt
 LOG: grid.vtu.lt

Fri Mar 9 16:38:50 EET 2007

SITE: KTU-BG-GLITE

GIIS: ce.bg.ktu.lt

LOG: ce.bg.ktu.lt

Paketinio vėlinimo, paketinių fliktuacijų bei pasiekiamumo apskaičiuoto rezultato pavyzdys:

Fri Mar 9 16:36:42 EET 2007	fangorn.man.poznan.pl	13.756
Esamas: 25	Viso: 25	Pasiekiamumas: 1
min: 13.638 ms, max: 14.122 ms	Fliktacija: 0.484 ms	velinimas: 13.7744 ms
Fri Mar 9 16:36:42 EET 2007	ares02.cyf-kr.edu.pl	21.874
Esamas: 25	Viso: 25	Pasiekiamumas: 1
min: 21.722 ms, max: 21.961 ms	Fliktacija: 0.239 ms	velinimas: 21.81812 ms
Fri Mar 9 16:36:42 EET 2007	zeus02.cyf-kr.edu.pl	21.997
Esamas: 25	Viso: 25	Pasiekiamumas: 1
min: 21.695 ms, max: 22.064 ms	Fliktacija: 0.369 ms	velinimas: 21.8826 ms
Fri Mar 9 16:36:42 EET 2007	kriit.eenet.ee	16.370
Esamas: 25	Viso: 25	Pasiekiamumas: 1
min: 16.191 ms, max: 16.621 ms	Fliktacija: 0.43 ms	velinimas: 16.3984 ms
Fri Mar 9 16:36:42 EET 2007	grid.fi.lt	2.063
Esamas: 25	Viso: 25	Pasiekiamumas: 1
min: 1.873 ms, max: 2.649 ms	Fliktacija: 0.776 ms	velinimas: 2.0242 ms
Fri Mar 9 16:36:42 EET 2007	fwe01.ifj.edu.pl	23.544
Esamas: 25	Viso: 25	Pasiekiamumas: 1
min: 22.058 ms, max: 25.658 ms	Fliktacija: 3.6 ms	velinimas: 22.95096 ms
Fri Mar 9 16:36:42 EET 2007	birzs.latnet.lv	7.779
Esamas: 25	Viso: 25	Pasiekiamumas: 1
min: 7.416 ms, max: 15.368 ms	Fliktacija: 7.952 ms	velinimas: 8.24112 ms
Fri Mar 9 16:36:42 EET 2007	puduris.latnet.lv	7.653
Esamas: 25	Viso: 25	Pasiekiamumas: 1
min: 7.357 ms, max: 12.219 ms	Fliktacija: 4.862 ms	velinimas: 8.12296 ms
Fri Mar 9 16:36:42 EET 2007	atomas.itpa.lt	1.588
Esamas: 25	Viso: 25	Pasiekiamumas: 1
min: 1.545 ms, max: 1.852 ms	Fliktacija: 0.307 ms	velinimas: 1.6382 ms
Fri Mar 9 16:36:42 EET 2007	grid.pri.kmu.lt	3.420
Esamas: 25	Viso: 25	Pasiekiamumas: 1

min: 3.332 ms, max: 5.367 ms	Fliuktacija: 2.035 ms	velinimas: 3.51296 ms
Fri Mar 9 16:36:42 EET 2007	ce.bg.ktu.lt	0.484
Esamas: 25	Viso: 25	Pasiekiamumas: 1
min: 0.402 ms, max: 0.616 ms	Fliuktacija: 0.214 ms	velinimas: 0.49004 ms
Fri Mar 9 16:36:42 EET 2007	pupa.elen.ktu.lt	0.266
Esamas: 25	Viso: 25	Pasiekiamumas: 1
min: 0.190 ms, max: 0.419 ms	Fliuktacija: 0.229 ms	velinimas: 0.27236 ms
Fri Mar 9 16:36:42 EET 2007	dane.ku.lt	2.927
Esamas: 25	Viso: 25	Pasiekiamumas: 1
min: 2.887 ms, max: 3.153 ms	Fliuktacija: 0.266 ms	velinimas: 2.99296 ms
Fri Mar 9 16:36:42 EET 2007	grid.mii.lt	1.736
Esamas: 25	Viso: 25	Pasiekiamumas: 1
min: 1.613 ms, max: 1.903 ms	Fliuktacija: 0.29 ms	velinimas: 1.7416 ms
Fri Mar 9 16:36:42 EET 2007	egee-ce01.pdc.kth.se	36.784
Esamas: 25	Viso: 25	Pasiekiamumas: 1
min: 36.637 ms, max: 39.876 ms	Fliuktacija: 3.239 ms	velinimas: 37.15184 ms
Fri Mar 9 16:36:42 EET 2007	g03n02.pdc.kth.se	36.973
Esamas: 25	Viso: 25	Pasiekiamumas: 1
min: 36.611 ms, max: 49.592 ms	Fliuktacija: 12.981 ms	velinimas: 37.55824 ms
Fri Mar 9 16:36:42 EET 2007	g03n04.pdc.kth.se	36.799
Esamas: 24	Viso: 25	Pasiekiamumas: 0.96
min: 36.657 ms, max: 38.494 ms	Fliuktacija: 1.837 ms	velinimas: 36.8598333333 ms
Fri Mar 9 16:36:42 EET 2007	ce01.grid.etf.rtu.lv	7.551
Esamas: 24	Viso: 25	Pasiekiamumas: 0.96
min: 7.516 ms, max: 8.042 ms	Fliuktacija: 0.526 ms	velinimas: 7.70416666667 ms
Fri Mar 9 16:36:42 EET 2007	grid.su.lt	5.621
Esamas: 24	Viso: 25	Pasiekiamumas: 0.96
min: 5.335 ms, max: 6.439 ms	Fliuktacija: 1.104 ms	velinimas: 5.624375 ms
Fri Mar 9 16:36:42 EET 2007	oberon.hep.kbfi.ee	12.892
Esamas: 24	Viso: 25	Pasiekiamumas: 0.96
min: 12.892 ms, max: 13.141 ms	Fliuktacija: 0.249 ms	velinimas: 12.981 ms
Fri Mar 9 16:36:42 EET 2007	penguin.mt.ut.ee	16.928
Esamas: 24	Viso: 25	Pasiekiamumas: 0.96
min: 16.594 ms, max: 20.757 ms	Fliuktacija: 4.163 ms	velinimas: 17.0655 ms
Fri Mar 9 16:36:42 EET 2007	arvutaja.physic.ut.ee	16.753
Esamas: 24	Viso: 25	Pasiekiamumas: 0.96
min: 16.576 ms, max: 18.304 ms	Fliuktacija: 1.728 ms	velinimas: 16.9003333333 ms

Fri Mar 9 16:36:42 EET 2007	vdupdc.vdu.lt	0.641
Esamas: 24	Viso: 25	Pasiekiamumas: 0.96
min: 0.575 ms, max: 0.890 ms	Fliuktacija: 0.315 ms	velinimas: 0.759166666667 ms
Fri Mar 9 16:36:42 EET 2007	grid.vtu.lt	-1
Esamas: 0	Viso: 25	Pasiekiamumas: 0
min: 100000 ms, max: 0 ms	Fliuktacija: -100000 ms	velinimas: 0 ms
Fri Mar 9 16:36:42 EET 2007	grid2.mif.vu.lt	2.211
Esamas: 24	Viso: 25	Pasiekiamumas: 0.96
min: 2.039 ms, max: 332.442 ms	Fliuktacija: 330.403 ms	velinimas: 15.9305833333 ms
Fri Mar 9 16:36:42 EET 2007	grid5.mif.vu.lt	2.215
Esamas: 24	Viso: 25	Pasiekiamumas: 0.96
min: 2.079 ms, max: 2.399 ms	Fliuktacija: 0.32 ms	velinimas: 2.25079166667 ms
Fri Mar 9 16:36:42 EET 2007	ce.egee.man.poznan.pl	-1
Esamas: 0	Viso: 25	Pasiekiamumas: 0
min: 100000 ms, max: 0 ms	Fliuktacija: -100000 ms	velinimas: 0 ms