

ŠIAULIŲ UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Andrius Sapiega
Informatikos (mok.) specialybės II kurso dieninio skyriaus studentas

**GRID TECHNOLOGIJA MOKYMO PROCESE: DUOMENŲ
VIZUALIZAVIMAS**

**GRID TECHNOLOGY FOR THE EDUCATIONAL PROCESS: DATA
VISUALIZATION**

MAGISTRO DARBAS

Darbo vadovas:
Lekt. V. Giedrimas

Recenzentas:
Doc. Dr. V. Sirius

Šiauliai, 2009

TURINYS

IVADAS	4
I. GRID TINKLAI IR JŲ NAUDOJIMO SRITYS	6
1.1. Grid technologijos analizė.....	6
1.2. <i>Grid</i> telkinio sandara.....	6
1.3. Grid tinklai Lietuvoje.....	8
1.4. Lygiagrečiai uždaviniai ir naudojimo sritys.....	10
1.5. Grid technologijos naudojimo mokymo procese būdai.....	14
II. DUOMENŲ VIZUALIZAVIMAS	15
2.1. Duomenų vizualizavimo samprata.....	15
2.2. <i>Grid</i> telkinių resursus vizualizavime naudojančios organizacijos.....	17
III. PROJEKTINĖ DARBO DALIS	26
3.1. Darbinės srities modelis.....	26
3.2. Įrankių ir priemonių pasirinkimo analizė.....	26
3.3. Projekto aprašymas.....	29
3.4. Aplinkos pasiruošimas darbui.....	31
3.5. Programos projektavimas.....	34
3.6. Problemų ir jų sprendimo aprašymai ir pagrindimai.....	38
IV. DARBO REZULTATŲ ANALIZĖ	40
PATARIMAI, PASTEBĖJIMAI, REKOMENDACIJOS	45
IŠVADOS	46
LITERATŪRA	47
SUMMARY	50
PRIEDAI	51

ILIUSTRACIJŲ SĄRAŠAS

1 pav. Šiaulių universiteto <i>Grid</i> telkinys	7
2 pav. LitGrid telkiniai Lietuvoje	8
3 pav. Uždavinio vykdymas 1 branduolio kompiuterio procesoriuje.....	10
4 pav. Lygiagretūs procesai pasaulyje	11
5 pav. Užduoties vykdymas 4 branduolių procesoriuje.....	11
6 pav. <i>Grid</i> naudojimas mokslinėje veikloje, vizualizavimas	12
7 pav. Uždavinio sprendimas ir duomenų vizualizavimas	15
8 pav. Ultragarinis skeneris.....	18
9 pav. Teršalų pasklidimo atmosferoje simuliacija	18
10 pav. Kraujo tėkmės kraujagyslėje vizualizavimas.....	19
11 pav. Oro prognozės modelio vizualizavimas.....	20
12 pav. Apsemiamų teritorijų modelio vizualizavimas.....	20
13 pav. Šiluminės elektrinės poveikio aplinkai spindulio vizualizavimas	20
14 pav. Duomenų vizualizavimas VizNet	21
15 pav. Gvid sandara	23
16 pav. Vizualizavimo laboratorija „Bifi“ organizacijoje	24
17 pav. Vizualaus kraujagyslių gydymo proceso etapų iliustracija	24
18 pav. ParaView programos langas	25
19 pav. Projekto darbinės srities būsenų diagrama.....	26
20 pav. Užduoties siuntimo ir vykdymo <i>Grid</i> telkinyje būsenų diagrama.....	26
21 pav. Programavimo aplinkos	27
22 pav. tightVNC programos pradinis prisijungimo langas	31
23 pav. „ui.grid.su.lt“ aplinka.....	32
24 pav. UIPnP paleidimas. Gaunami sertifikatai.....	33
25 pav. Pasibaigęs sertifikatas	33
26 pav. Automatinis UIPnP skripto įgyvendinimas	34
27 pav. Programos projekto klasių diagrama	35
28 pav. Resursų kontrolės modulis vykdomas grid telkinyje, atminties paskirstymas sukurtiems procesams.....	40
29 pav. Duomenų vizualizavimas naudojant grafiką intervale [-10,10] ir atsitiktinius skaičius generuojanti funkcija.	42
30 pav. Funkcijos $x = y$ kubu, kai $y \in [1;1000]$ reikšmių grafikas.....	42
31 pav. Spalvų gamos ir atsitiktinių duomenų grafikai.....	43

ĮVADAS

Mokslinėje veikloje, versle ir kt. nuolat susiduriama su vienokiais ar kitokiais uždaviniais, kuriems spręsti naudojamas kompiuteris. Kuo sudėtingesnis uždavinys tuo daugiau reikia skaičiavimo pajėgumų, todėl ypatingai sudėtingiems uždaviniams spręsti pasitelkiami superkompiuteriai.

Grid technologija leidžia kompiuterius, bei serverius apjungti į vieną sistemą, kurios skaičiavimo pajėgumai prilyginami superkompiuteriams.

Terminas *Grid* buvo pradėtas naudoti apie 90-tųjų metų vidurį, kuris apibrėžė išplėstinę mokslo ir inžinerijos infrastruktūrą. Sąvoka *Grid* yra angliškas žodis, kuris lietuviškai reiškia „*tinklas*“, arba kitaip „*tinklynas*“ [24]. *Grid* tinklas – tai lygiagrečiųjų ir paskirstytų skaičiavimų tinklas, kurį sudaro geografiškai nutolę kompiuteriniai skaičiavimų telkiniai, klasteriai (*angl. cluster*). Visame pasaulyje ši technologija jau senai naudojama, tačiau Baltijos šalyse ši technologija pradėta naudoti neseniai, kai buvo įkurta BalticGrid virtuali organizacija.

Grid tinklai naudojami mokslinėje veikloje, taikomuosiuose uždaviniuose, bei komercinėje veikloje.

Šis darbas orientuotas į *Grid* technologijos pritaikymą mokymo procese, specializuojantis į duomenų vizualizavimą. Lietuvoje tokio pobūdžio darbų nėra daug, todėl darbas turi didelę praktinę reikšmę. Jame atskleidžiamas statinis duomenų vizualizavimas *Grid* tinkluose.

Vizualizavimas – tai grafinis informacijos pateikimas, naudojamas tada, kai reikia įvertinti labai didelį duomenų kiekį.

Šio darbo pagrindinis probleminis klausimas: kokiose mokymo proceso srityse galima panaudoti *Grid* technologiją?

Tikslas - išanalizuoti *Grid* technologijos panaudojimą mokymo procese perspektyvas.

Uždaviniai:

1. Išsiaiškinti *Grid* technologijos sąvoką;
2. Aprašyti *Grid* technologijos taikymo prielaidas;
3. Išanalizuoti *Grid* technologijos panaudojimo mokymo procese metodus;
4. Išanalizuoti *Grid* technologijos panaudojimo mokymo procese vizualizavimo būdus;
5. Naudojant *Grid* technologiją sukurti priemonę duomenų vizualizavimui.

Tyrimo objektas – *Grid* technologijos naudojimo galimybės mokymo procese.

Tyrimo metodai – literatūros analizė, eksperimentas.

Darbo struktūra. Magistro baigiamąjį darbą sudaro, įvadas, 4 skyriai, išvados, rekomendacijos, panaudotos literatūros sąrašas ir priedai. Darbo apimtis 51 puslapių, 42 iliustracijos.

- Pirmame skyriuje pateikta *Grid* tinklų sąvoka ir jų naudojimo sritys;
- Antrame skyriuje analizuojama vizualizavimo samprata, aprašytos vizualizavimą naudojančios organizacijos bei vizualizavimo taikymas;
- Trečiame skyriuje nagrinėjama darbo projektinė dalis – darbinės srities modelis, įrankių ir priemonių pasirinkimo analizė, projekto aprašymas ir kt.
- Ketvirtame skyriuje pateikti ir aprašyti praktinio darbo rezultatai.

I. GRID TINKLAI IR JŲ NAUDOJIMO SRITYS

1.1. Grid technologijos analizė

Grid technologija – vieni nuo kitų geografiškai nutolę kompiuterių telkiniai, apjungti į tinklą. Šie kompiuteriai apjungti spręsti uždaviniams, kurie reikalauja didelių lygiagrečių skaičiavimų resursų. Žinoma tam yra superkompiuteriai, tačiau jie yra labai brangūs (per brangūs mažesnį biudžetą turinčioms mokslo institucijoms). Todėl *Grid* skaičiavimų tinklai¹ sukurti tam, kad pakeistų brangius didelių skaičiavimų įrenginius - superkompiuterius.

Panašus projektas buvo SETI (*angl. Search for Extraterrestrial Intelligence*) nežemiško intelekto paieška, tam kiekvienas turintis namie kompiuterį laisvu metu gali prisidėti prie duomenų gaunamų iš kosmoso analizės. Tam reikia parsisiųsti specialią programą, kuri leidžia namų vartotojų kompiuterius apjungti į superkompiuterį.

Grid tinklams reikalinga speciali programinė įranga, kuri sprendžiamą uždavinį išdalintų ne daugiau kaip dviems tūkstančiams *Grid* tinkle apjungtų kompiuterių. *Grid* tinklą galime sukurti kaip viešąjį telkinį arba kaip uždaros korporacijos telkinį.

Pasak John Patrick IBM korporacijos Interneto strategijų viceprezidento „Kitas didelis objektas bus *Grid* skaičiavimai“.[41]

Grid tinklą sudaro telkiniai *angl. cluster*. Telkinį gali sudaryti mums įprasti namų kompiuteriai su įprasta įranga (motininė plokštė, procesorius, atmintis, kietasis diskas, tinklo ar kita ryšio plokštė) arba specializuoti serveriai, tik svarbu, kad jie turėtų tinklo įrangą, pavyzdžiui Ethernet tinklo plokštę.

Viena telkinį gali sudaryti 10, 20 ir daugiau kompiuterių, kompiuterių, serverių skaičius neribotas. O visą *Grid* tinklą sudaro keletas ar kelios dešimtys tarpusavyje sujungtų telkinių.

Grid skaičiavimų technologija pagrįsta bendradarbiavimu, kiekviena organizacijoje dalyvaujanti institucija savo išteklius pateikia kitiems organizacijos nariams.

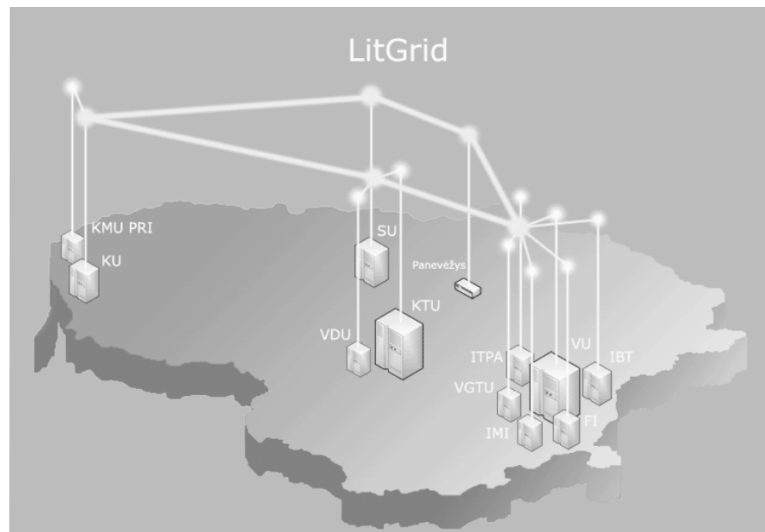
1.2. *Grid* telkinio sandara

Grid tinklų bendruomenės, kuriose bendraminčiai gali dirbti atlikdami tuos pačius tyrimus, naudodami tam tikrą programinę įrangą yra vadinamos *virtualiomis organizacijomis* (VO).

¹ Lokaliųjų skaičiavimų tinklų junginiai (klasteriai arba telkiniai *angl. clusters*)

1.3. Grid tinklai Lietuvoje

LitGrid – tai Lietuvos akademinė institucijų lygiagrečių ir paskirstytų skaičiavimų tinklas [1]. **Lygiagrečių ir paskirstytų skaičiavimų tinklai** [1] – tai nauja informacinių technologijų evoliucijos pakopa. Šio tinklo technologijų pagalba yra sujungti geografiškai nutolę klasteriai, galintys spręsti didelės skaičiavimo apimties reikalaujančius uždavinius Lietuvoje. *LitGrid* tinklo resursai išsibarstę visoje Lietuvoje (žr. 2 pav.), todėl jų valdymas, administravimas ir stebėjimas atliekamas specialių programų pagalba.



2 pav. LitGrid telkiniai Lietuvoje

Šaltinis: A. Juozapavičius (2008). Grid technologijų diegimas. Vilniaus universitetas.

LitGrid tinklo resursai – kompiuterių klasteriai ir saugyklos, sujungti naudojantis Lietuvos mokslo ir studijų kompiuterių tinklo LitNet infrastruktūra. LitGrid tinklas yra glaudžiai susijęs su kuriama Baltijos šalių lygiagrečių ir paskirstytų skaičiavimų tinklu BalticGrid, bei Europos *Grid* tinklu - EGEE. Numatoma, kad LitGrid kaip nacionalinė *Grid* infrastruktūra, įsijungs į Europinės *Grid* iniciatyvos veiklą.

LitGrid veiklą remia ir finansuoja Lietuvos Respublikos Švietimo ir mokslo ministerija.

Lietuvos švietimo ir mokslo ministras Algirdas Monkevičius su delegacija 2008 m. dalyvavo didžiojo hadronų greitintuvo atidarymo ceremonijoje. Lietuvai yra sudaryta galimybė dalyvauti viename iš keturių pagrindinių eksperimentų – CMS. Šiuo metu CERN veikloje aktyviai dalyvauja Vilniaus universiteto Fizikos fakulteto Puslaidininkių fizikos katedros prof. Juozo V. Vaitkaus vadovaujama grupė, kuri kuria naujo tipo įrenginius detektuojančius dalelių savybes, bei Lietuvos informatikai (*LitGrid* nariai), vadovaujami prof.

A. Juozapavičiaus, kurie glaudžiai įsitraukę į CERN skaičiavimus ir jų lygiagrečių ir paskirstytų skaičiavimų tinklą [2].

Lygiagrečiųjų ir paskirstytų skaičiavimų tinklas LitGrid orientuotas į Lietuvos akademinės ir mokslo visuomenės poreikius. Tinklo infrastruktūra ir programinė įranga naudojama fundamentiniams ir taikomiesiems moksliniams tyrimams atlikti. LitGrid tinklas projektuojamas kaip atviras Lietuvai aktualiems moksliniams tyrimams atlikti tinklas. Ateityje plečiantis tinklo infrastruktūrai, numatoma plėsti ir LitGrid teikiamų paslaugų spektrą.

LitGrid tikslai:

- Sukurti lygiagrečių ir paskirstytų skaičiavimų tinklo infrastruktūrą Lietuvoje bei integruoti ją į Europos Sąjungos šalių kuriamą EGI infrastruktūrą, kitus lygiagrečių ir paskirstytų skaičiavimų tinklus - Baltijos šalių BalticGrid ir didžiausią Europos tinklą EGEE;
- Suteikti Lietuvos mokslininkams ir specialistams skaičiavimo tinklų technologijų žinias ir galimybes naudotis tinklo resursais;
- Skatinti Lietuvos mokslininkus ir specialistus dalyvauti skaičiavimo tinklų technologijų vystyme, įtraukti juos į naujų tinklinių technologijų kūrimą;
- Kurti lygiagrečių ir paskirstytų skaičiavimo tinklų technologijomis pagrįstą taikomąją programinę įrangą bei adaptuoti esamas darbu tokiame tinkle.

LitGrid uždaviniai:

- Siekiant įgyvendinti programoje iškeltų tikslų vykdomi sekantys darbai:
- LitGRID infrastruktūros kūrimas, tobulinimas ir palaikymas;
- Matematinių ir kompiuterinių algoritmų, naudojančių lygiagrečius ir paskirstytus skaičiavimus, kūrimas;
- LitGRID taikomųjų uždavinių, taikymų ir paslaugų projektavimas, diegimas, palaikymas;
- LitGRID naudotojų apmokymo, techninės ir metodinės medžiagos rengimas;
- LitGRID valdymas ir administravimas.

LitGrid programa yra ilgalaikė, todėl sukūrus lanksčią LitGrid infrastruktūrą bus sudarytos sąlygos bei sukurti efektyvūs įrankiai didelės apimties uždavinių sprendimui. Tai sudarys galimybes Lietuvos mokslininkams būti pilnateisiais pasaulinio lygio tyrėjų partneriais.

Šiaulių universitetas LitGrid tinklą naudoja atlikti moksliniams tyrimams temomis:

- Sistemų skaitinis modeliavimas ir tyrimas;
- Automatizuotas programų sistemų kūrimas;

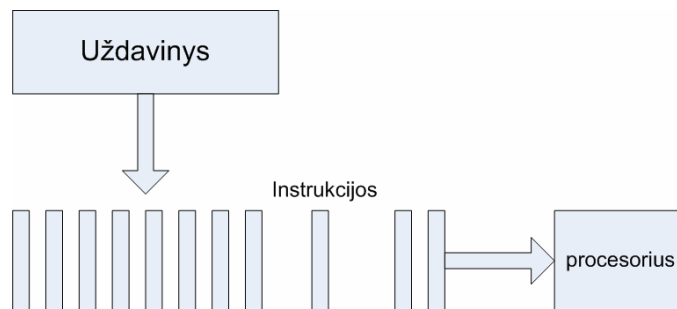
- *Grid* tinklų naudojimas mokymui.

Apibendrinant galima teigti, kad LitGrid panaudojimo sritys orientuotos daugiausiai į mokslinę-tiriamąją veiklą.

1.4. Lygiagrečiai uždaviniai ir naudojimo sritys

Daugelio profesijų atstovai kiekvieną dieną darbo metu naudoja kompiuterius. Yra žinoma, kad dažniausiai darbiniai kompiuteriai turi po vieną procesorių. Vienas procesorius vienu metu gali atlikti milijonus operacijų per sekundę, tačiau vienu metu gali vykdyti tik vieną užduotį.

Šiandieninėje procesorių rinkoje jau yra ir dviejų branduolių *angl. dualcore*, keturių branduolių *angl. quad core* procesoriai galintys vienu metu atlikti kelias ar keletą užduočių vienu metu 3 pav. pavaizduota kaip kompiuteris su vienu procesoriumi atlieka sprendžia uždavinį.



3 pav. Uždavinio vykdymas 1 branduolio kompiuterio procesoriuje

Šaltinis: Sudaryta darbo autoriaus, remiantis B. Barney (2009). Introduction to Parallel Computing.

Realiame pasaulyje yra daug lygiagrečių procesų 4 pav.

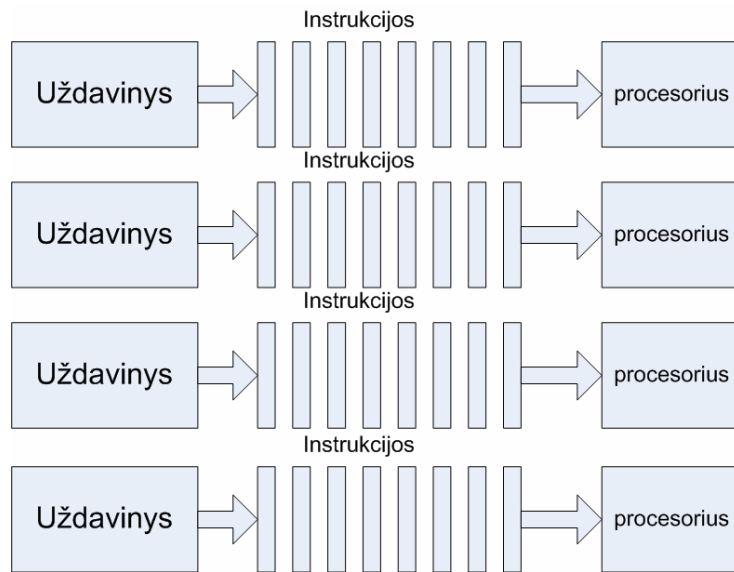
- Planetų judėjimas;
- Galaktikos formavimas;
- Automobilių surinkimas;
- Oro ir vandenynų vandens judėjimas;
- Tektoninių plokščių judėjimas;
- Automobilių judėjimas;
- Lėktuvų surinkimas;
- Įvairių cheminių, fizikos dalelių judėjimas;
- Net aptarnavimas restorane atliekamas lygiagrečiai.



4 pav. Lygiagretūs procesai pasaulyje

Šaltinis: B. Barney (2009). Introduction to Parallel Computing

Todėl, lygiagretieji skaičiavimai yra neišvengiami. Žemiau 5 pav. pavaizduotas uždavinio sprendimas lygiagrečiuoju atveju, užduotis skaidoma į smulkesnes užduotis ir šie uždaviniai vykdomi vienu metu, lygiagrečiai.



5 pav. Užduoties vykdymas 4 branduolių procesoriuje

Šaltinis: Sudaryta darbo autoriaus, remiantis B. Barney (2009). Introduction to Parallel Computing.

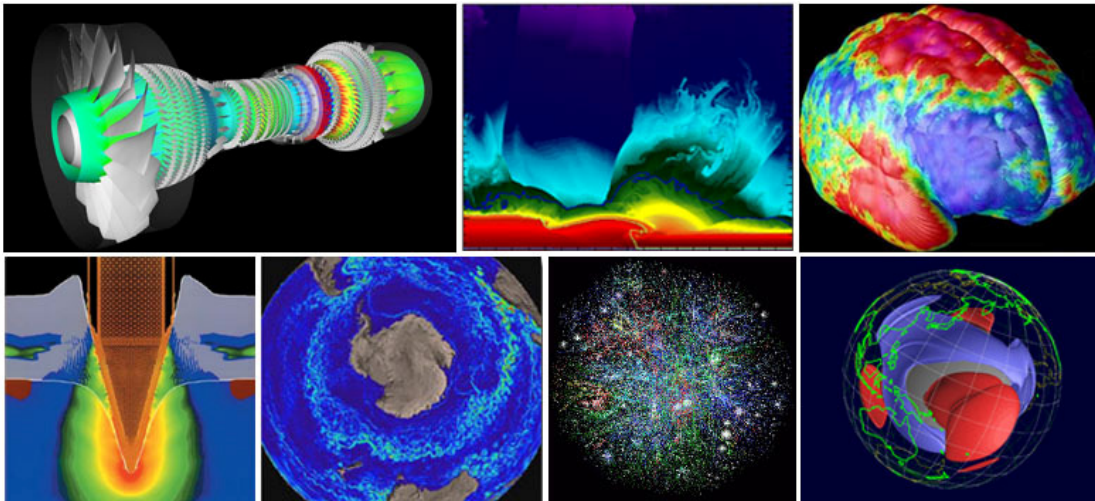
Istoriškai *Grid* buvo aukšto lygio kompiuterinės sistemos skirtos tyrinėti arba specifiniams moksliniams ir inžineriniams uždaviniams bei jų sprendimams.

Lygiagrečiųjų uždavinių pasaulyje daug, todėl *Grid* technologija taikoma labai plačiai.

Grid technologijos naudojimo mokslinėje veikloje sritys:

- Meteorologijoje – atmosferos, žemės, aplinkos tyrimams;

- Fizikos taikyme – branduolinėje fizikoje, dalelių, kondensuotų medžiagų, aukšto slėgio, sintezės ir t.t. Vienas iš pavyzdžių Europos branduolinių tyrimų organizacija CERN ir vienas iš didžiausių fizikos bandymų didžiausias pasaulyje hadronų greitintuvas, kuris startavo šiais 2008 metais. Juo gautiems duomenims įvertinti buvo numatyta naudoti *Grid* paskirstytų skaičiavimų technologijos resursus [2];
- Biologijoje, biotechnologijoje, genetikoje;
- Chemijoje, molekulinuose moksluose;
- Geologijoje, Seismologijoje;
- Mechaninėje inžinerijoje – nuo protezavimo iki erdvėlaivių kūrimo. Pavyzdžiui Ford Motors kompanija, Saab automobiliai naudoja *Grid* technologiją įvairiems automobilių gamybos uždaviniams spręsti;
- Elektrotechnikoje, schemų dizaine, mikroelektronikoje;
- Informatikoje, matematikoje – įvairiems skaičiavimams;
- Bioinformatikoje. Bioinformatika - tai mokslas apjungiantis informacines technologijas bioinžineriją ir mediciną. Vienas iš pavyzdžių kuriamos genomo, pogenominės (genomo kartografavimo projektas) disciplinos ir siekiama išgyti vėžį.



6 pav. *Grid* naudojimas mokslinėje veikloje, vizualizavimas

Šaltinis: B. Barney (2009). Introduction to Parallel Computing

Lygiagrečiai kompiuteriai naudojami ne vien tik mokslininkų, jie naudojami ir komercijoje - įvairios įmonės, kuriose uždavinių sprendimai, siekiai reikalauja didelių skaičiavimų ar duomenų apdorojimo:

- Didelių duomenų bazių, duomenų gavybos uždaviniuose;
- Naftos žvalgyboje, tiriant žemės sluoksnius;

- Medicinoje, vizualizavimo ir diagnostikos problemoms spręsti;
- Nacionalinėse organizacijose, valdyme;
- Finansiniai ir ekonominiai modeliavimai. Pvz. Wachovia naudoja *Grid* technologiją finansinių modelių tyrimams [42];
- Išplėstinė grafika ir virtualios realybės kūrimas. Žaidimų kūrimas - Butterfly.net yra pirmoji komercinė *Grid* sistema, kuri apdoroja internete virtualius vaizdo žaidimus per tinklą, todėl leidžia efektyviai panaudoti kompiuterio išteklius, kuriant ypač didelius trimačius žaidimų pasaulius;
- Tinklinėms vaizdo ir daugialypės terpės technologijoms;
- Vidiniams korporacijų tinklams - atsiradus pasauliniam masto tinklui (internetui) atsirado galimybė sujungti atskirus korporacijų intranetus tuo pačiu sukuriant *grid* telkinį. Dalinamasi resursais, siekiant išspręsti bendrus uždavinius;
- *Grid* naudojamas įvairiems tyrimams naudoja NASA, JAV gynybos departamentas ir kt.

Grid naudojimo priežastys:

- ✓ *Grid* tinklai kainuoja mažiau, nei superkompiuteriai, o skaičiuojamoji galia didelė, prilyginama superkompiuteriams. Ne visos mokslo organizacijos, gali įsigyti superkompiuterius, todėl šie tinklai yra gera išeitis;
- ✓ *Grid* tinklai leidžia atlikti sudėtingus uždavinius, o daugelis mokslinėje veikloje vykdomų uždavinių yra gana sudėtingi;
- ✓ *Grid* tinklai konkurencingi – leidžia bendradarbiauti apjungti darbus virtualioje erdvėje. Pvz. „accessgrid“ jis leidžia skirtingu šalių vartotojams bendradarbiauti sprendžiant uždavinius. (www.accessgrid.org);
- ✓ Įgalina naudoti, nutolusius skaičiavimo resursus. Resursai gali būti apjungti į kitus tinklus ir taip padidinama skaičiavimo galia.

Taigi iš pateiktų pavyzdžių galima teigti, kad *Grid* technologija taikoma įvairiose srityse, panaudojant lygiagrečiųjų uždavinių skaičiavimus.

1.5. Grid technologijos naudojimo mokymo procese būdai

Jau išsiaiškinome, kad *Grid* technologija labai plačiai taikoma įvairiose mokslo srityse, tačiau šiame darbe aktualiausias *Grid* technologijos taikymas mokymo procese ir vizualizavime. Lietuvoje *Grid* technologijos taikymas mokymo procese retai sutinkamas, tačiau užsienyje galima rasti daug *Grid* technologijos taikymo pavyzdžių. Todėl remiantis šaltiniais darbo autorius išskyrė tokius *Grid* technologijos naudojimo mokymo procese būdus:

a) Virtuali observatorija, aplinka. Darbas su specializuota programine įranga aprūpinta *Grid* technologija paremta sistema. Pvz.: meteorologijoje, chemijoje, architektūroje, fizikoje ir t.t. kai sudaroma virtuali aplinka, sąlygos ir atliekami eksperimentai. Čia labai dažnai reikia didelių kompiuterio skaičiavimų. Darbas interaktyvus, duomenys apdorojami ir dinamiškai vaizduojami.

b) Laboratoriniai darbai. Vietoje darbo įprastoje kompiuterių klasėje naudojant konkretų kompiuterį galima naudoti *Grid* kompiuterių tinklą. Pavyzdžiui vyksta kriptografijos laboratoriniai darbai studentams užduodame užduoti iššifruoti laišką AES ar RSA algoritmu pasitelkiant *Grid* skaičiavimo resursus.

c) Praktiniai darbai. Kursinių, bakalauro, magistro, doktoranto diplominių darbų atlikime, taip pat savarankiškuose darbuose, naudojant kaip priemonę darbo atlikimui.

d) Mokomieji *Grid* tinklai. Sukuriama organizacija, kuri pasitelkiant „web“ servisus ir *Grid* technologiją sukuria tam tikra infrastruktūrą, kuri suteikia vartotojams tam tikras paslaugas. Pavyzdžiui bendrų uždavinių sprendimas, resursų dalinimasis, nuotolinio mokymosi aplinka ir t.t.. Lietuvoje Kauno technologijos universitete yra vykdomas panašus projektas EleGI projektas.

Projekto tikslas – sukurti virtualios organizacijos prototipą *GRID* technologijų pagrindu. Prototipas apima technologines priemones ir resursus, skirtus apibrėžtų vartotojų grupių mokymo paslaugoms kurti ir valdyti.

Pagal numatytas funkcijas apibrėžti atitinkami paslaugų tipai ir vartotojų prieigos per *GRID* platformą būdai naudojant Globus bei kitas su *GRID* technologija suderintas programines priemones. Sukurtas vartotojo sąsajos sluoksnis ir ištirti galimi duomenų apykaitos atvejai bei parinkti būdai ir priemonės, kaip turėtų būti užtikrintos šios funkcijos virtualios organizacijos kontekste. Virtualios organizacijos dalyvių mokymui sukurta mokymo medžiaga moduliui „Virtualiosios Organizacijos įgyvendinimas *GRID* technologijų pagrindu“. Komponavimo paslauga išbandyta mokymo kursuose naudojantis aiškinamųjų objektų saugyklos, aiškinamųjų objektų metaduomenų, komponavimo paslaugos bei vykdomųjų objektų metaduomenų papildomas paslaugomis.

II. DUOMENŲ VIZUALIZAVIMAS

2.1. Duomenų vizualizavimo samprata

Šiuolaikinis mokslas nagrinėja įvairias problemas, sprendžiami didesni ir mažesni uždaviniai. Didesnių uždavinių sprendimui ir jų rezultatų pateikimui dažnai naudojamas duomenų vizualizavimas.

Duomenų vizualizavimo apibrėžimas įvairiose moksliniuose šaltiniuose gana panašūs.

V. Šarikova vizualizavimo sąvoką apibrėžia taip: vizualizavimas tai vaizdus duomenų pateikimas. Vizualizavimo metodai ir technologijos skirtos vaizdžiai pateikti ir analizuoti duomenis, tam pasitelkiant kompiuterinės grafikos galimybes ir į šį procesą įjungiant biologinę žmogaus regos bei smegenų sistemą, turinčia galingas vaizdų analizavimo ir atpažinimo galimybes [43].

D. Čeikauskaitės diplominiame darbe vizualizavimas pateiktas kaip grafinis informacijos pateikimas. Darbe iškeliami pagrindinė vizualizavimo idėja – duomenis pateikti tokia forma, kuri leistų vartotojui suprasti tiriamus duomenis, daryti išvadas ir tiesiogiai juos įtakoti. Vizualizavimas leidžia geriau suvokti sudėtingas duomenų aibes, gali padėti nustatyti dominuojančius jos poaibius. Vizualią informaciją žmogus pajėgus suvokti daug greičiau negu tekstinę [44]. Kiek kitokią sąvoką pateikia S. K. Card. Anot šio autoriaus vizualizavimas tai - kompiuterio palaikomos interaktyvios, vaizdinės duomenų būsenos reprezentacijos.

Apibendrinant galima teigti, kad geriausiai duomenų vizualizavimą apibūdina sąvoka: vizualizavimas - grafinis informacijos pateikimas.

Uždaviniams įgyvendinti turime sudaryti jų kompiuterinį algoritmą, šį algoritmą paversti kompiuteriui suprantama kalba, jį įvykdyti kompiuteryje. Po algoritmo vykdymo gaunami rezultatai tekstinė, skaitinė informacija. Jei informacijos nedaug tada įvertinti rezultatus nesunku, tačiau kai kuriuose uždaviniuose informacijos labai daug jos įvertinimui sugaištama labai daug laiko arba informacijos suvokti paprastais metodais negalima, tada yra naudojamas duomenų vizualizavimas. Uždavinio sprendimo ir gautų duomenų vizualizavimo kompiuteriu supaprastinta schema pateikta 7 pav.



7 pav. Uždavinio sprendimas ir duomenų vizualizavimas

Šaltinis: Sudaryta darbo autoriaus

Duomenų vizualizavimo būdai priklauso nuo duomenų tipo. Duomenų tipai:

- *Statiniai* – tai duomenys kurie nekinta. Nekinta nei jų kiekis, nei reikšmės;
- *Dinaminiai* – kintantys duomenys. Gali kisti arba jų kiekis, arba reikšmės, arba ir duomenų kiekis, ir duomenų reikšmės.

Vizualizavimas paprastai kalbant tai duomenų ar rezultatų vaizdinis pateikimas.

Vizualizavimas – tai grafinis informacijos komunikavimas [3].

Remiantis šaltiniu [48] vizualizavimas skirstomas į tris dalis:

- *Statinis* – tai vizualizavimas, kurio metu pateikiami nekintantys vaizdiniai duomenys. Statinio vizualizavimo metu, mes galime įvertinti jau esamus statinius duomenis ir pateikti juos grafiškai. Pavyzdžiui matematinės funkcijos reikšmių pavaizdavimą grafike.
- *Interaktyvus* – tai vizualizavimas, kurio metu vartotojas gali interaktyviai valdyti duomenų vizualizavimą tai yra naršyti po tam tikra duomenų skiltį, keisti duomenų pateikimo masteli, padidinti, sumažinti ir pan. Šio tipo vizualizavimas *Grid* tinkluose naudojamas, tačiau reikalauja specialios infrastruktūros tai yra įdiegtos specialios programinės įrangos.
- *Animuotas* – tai vizualizavimas, kai gaunamas kintantis grafinis vaizdas, kai matomas duomenų vaizdinis pokytis. Pvz.: debesų garavimas arba oro srovių judėjimas.

Vizualizavimo naudojimo sritys:

- *Medicina;*
- *Fizika;*
- *Architektūra;*
- *Aviacija;*
- *Chemija;*
- *Matematika;*
- *Informatika;*
- *Ir kitos sritys.*

Duomenų vizualizavimui galima naudoti vieną kompiuterį, tačiau yra nemažai atvejų, kai naudojami *Grid* telkiniai: norima gauti kelis vizualizavimo atvejus arba vizualizavimo metodai reikalauja didelių skaičiavimų.

Fizinių mokslų srities kryptyse – fizikoje, chemijoje, medicinoje, informatikoje ir t.t. uždavinių sprendimas ir jų rezultatų vizualizavimas reikalauja didelių skaičiavimų.

Pavyzdžiui fizikoje atliekamas dalelių judėjimo vizualizavimas, chemijoje - cheminių procesų vizualizavimas, informatikoje - daugiamačių duomenų vizualizavimas ir kt.

Šiaulių universitete yra vykdoma nemažai fizinių mokslų kryptių studijų programų, todėl praktiškai kiekvienoje iš jų galima naudoti *Grid* technologiją: uždavinių sprendimui ir jų rezultatų pateikimui.

Kiekviena kryptis turi savo specifinius uždavinius ir specifinius rezultatus, todėl duomenų vizualizavimo metodai tinkantis matematikos duomenų vizualizavimui ne visada tinka fizikoje ir pan. Todėl tos mokslo krypties specialistai turėtų parengti savo mokslo šakai specializuotus uždavinių sprendimo būdus ir duomenų vizualizavimo metodus.

2.2. *Grid* telkinių resursus vizualizavime naudojančios organizacijos

Yra daug organizacijų naudojančių *Grid* technologiją, dažniausiai tai mokslo institucijos. Norint praktiškai atskleisti vizualizavimo naudojimo *Grid* technologijoje sritis, pateikiama keletas organizacijų veiklos aprašymų, bei vykdomi projektai.

INT.EU.GRID – Interaktyvus Europos *Grid* projektas, kurio siekis sukurti infrastruktūrą Europos tyrimų erdvėje, skirtą interaktyvių programų, skirtų moksliniams tyrimams vykdymui. Int.eu.grid projektas startavo 2006 metų kovo 1 dieną. Šį projektą vykdo 13 institucijų iš 7 valstybių. Pagrindinės projekto charakteristikos:

- ✓ Lygiagretusis interaktyvių užduočių vykdymas ir jų duomenų saugojimas „Tera“ lygiu;
- ✓ Prieigos prie draugiškos vartotojui interaktyvaus vizualizavimo aplinkos kūrimas;
- ✓ Virtualių organizacijų palaikymas visuose lygiuose: bendradarbiavimo aplinka, *Grid* programų stiprinimas, vykdymo ir stebėjimo įrankiai, diskusijos apie gautus rezultatus ir t.t. [33].

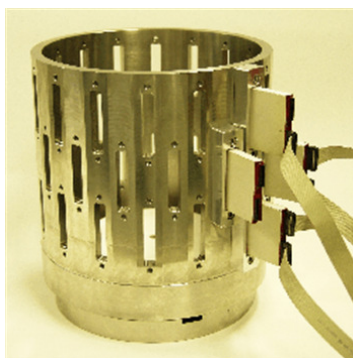
Interaktyvus Europos *Grid* tinklas apjungia kitus projektus tokius kaip EGEE arba CrossGrid.

INT.EU.GRID projekto svetainėje pateikti moksliniai tyrimai:

1. **Ultragarsu kompiuterinė tomografija** (*angl. Ultra Sound Computer Tomography*). Tai – vizualizavimo taikymo medicinoje pavyzdys. Medicinos atvaizdo vizualizavimo pagrindas yra skenuotų ultragarsinių duomenų vaizdinė rekonstrukcija (8 pav. pavaizduotas ultragarsinis skeneris).

Gauti skenerio duomenys (apie 20 Gb) yra persiunčiami analizavimui į *Grid* tinklą. Jei *Grid* telkinio skaičiavimo galia didelė, tai procesas užtrunka apie 30 minučių. Po to gauti rezultatai (apie 4 Gb informacijos) persiunčiami atgal į vartotojo kompiuterį. Vienu

kompiuteriu tokios užduoties sprendimas užtruktų labai ilgai, o ir atminties jis turėtų turėti labai daug. Šiuo atveju gelbėja *Grid* tinklų skaičiavimų galia ir dideli atminties resursai.



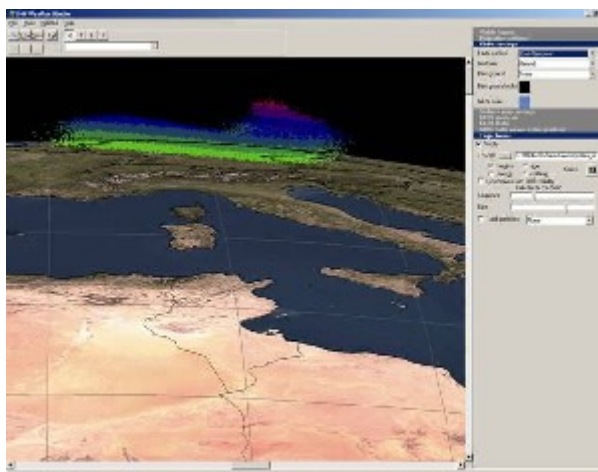
8 pav. Ultragarsinis skeneris

Šaltinis: Int.eu.grid svetainės duomenys [33]

2. Teršalų pasklidimo atmosferoje modeliavimas - simuliacija (*angl. Simulation of the Dispersion of Pollutant Agents in the Atmosphere*).

Interaktyvus Europos Grid tinklas nagrinėja daug praktinių uždavinių. Vienas iš jų teršalų pasklidimo atmosferoje modeliavimas.

Šis mokslinis tyrimas nagrinėja teršalų sklaidą atmosferoje: tiriamas nepriklausomų teršalų dalelių judėjimas (Lagranžo modelis). Terminas „dalelė“ reiškia oro teršalus arba konkrečias chemines medžiagas. 9 pav. vaizduojama šio modelio vizualizacija.



9 pav. Teršalų pasklidimo atmosferoje simuliacija

Šaltinis: Int.eu.grid svetainės duomenys [33]

Dalelių kelionės su vėjo, dalelių trajektorija ir jų sudėtis atspindima gamtos reiškiniais, tokiais kaip turbulencija, sausas nusėdimas, šlapias nusėdimas per lietų ir radioaktyviųjų dalelių irimas.

Analizuojamos dalelės yra paskirstomos tarp procesorių, kurie skaičiuoja individualias jų judėjimo trajektorijas. Kiekviena dalelių pozicija yra saugoma faile, kuris

vėliau yra naudojamas vizualizavimo tikslais. Be to, tūrio ir paviršiaus koncentracijos apskaičiuotos konkrečioms trimačio vaizdo tinklelio taškams.

3. **Atlas realaus laiko stebėsenos ir kalibravimo sistema** (angl. *ATLAS online monitoring and calibration system*).

Tai interaktyvaus QoS² (angl. *quality of service*) valdymo projektas:

- QoS tinklo valdymo programa;
- Atlas sistemos įvykių skirstymas;
- Vietinių svetainių tinklo efektyvumo analizavimas.

Sistemos pridėtinė vertė - tai programa specialiai pritaikyta *Grid* QoS testavimui. Programą gali būti potencialiai naudojama ir kitiems VO ne tik ATLAS VO. Šios programos vartotojai ATLAS eksperimento operatoriai.

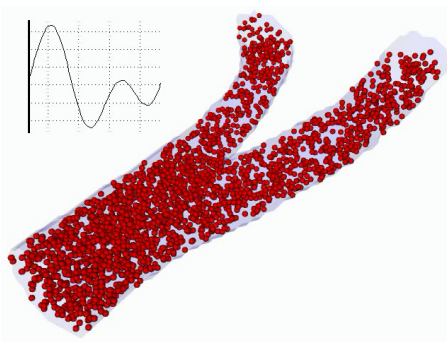
Projekte yra vykdoma begalė įvairiausių uždavinių ir mokslinių tyrimų, tačiau šiame darbe buvo išskirti trys pavyzdžiai.

Teršalų pasklidimo atmosferoje modeliavimas leidžia suprasti kaip sklinda teršalai, per kiek laiko jie nusėda ir t.t. Toks modeliavimas galėtų būti įtrauktas į gamtosaugininkų mokymo procesą.

Kitas pavyzdys vizualizavimo tyrimas, ultragarso kompiuterinė tomografija gali būti naudojamas ligoninėse gydytojų mokyme, jų kvalifikacijos kėlime ir pan.

CROSSGRID. Tai Europos projektas, kurio pagrindinis siekis išplėsti *Grid* technologijos naudojimo sritis, kuriant didelės praktinės reikšmės programas. Šio projekto uždaviniai skirstomi į sritis:

▪ **Biomedicinos.** Modeliuojama kraujagyslių rekonstrukcija – nepadarytos operacijos kraujo tėkmė, taip prognozuojama planuojamos kraujagyslių rekonstrukcijos operacijos sėkmė.



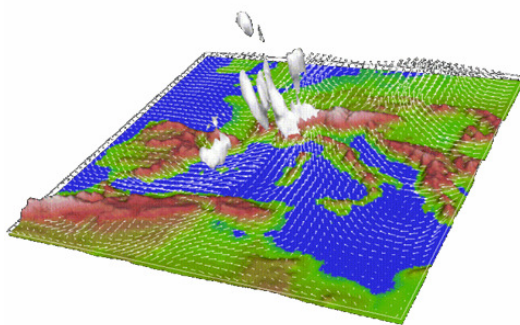
10 pav. Kraujo tėkmės kraujagyslėje vizualizavimas

Šaltinis: D. Albada. Biomedical Application (2002) CrossGrid

² Angl. Quality of Service (liet. Paslaugos kokybė) – sąvoka, žyminti tinklo paslaugos kokybės rodiklį.

▪ **Meteorologija.**

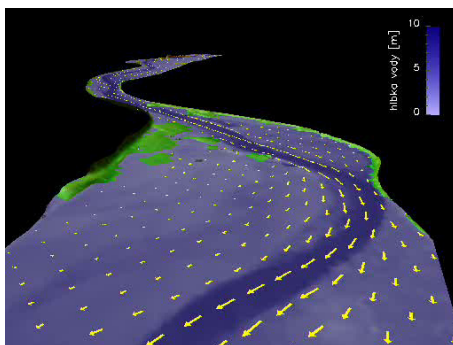
1. Orų prognozių modeliai



11 pav. Oro prognozės modelio vizualizavimas

Šaltinis: L. Hluchy. Flooding Crisis Team Support (2002) CrossGrid

2. Apsemtų teritorijų vizualizavimas

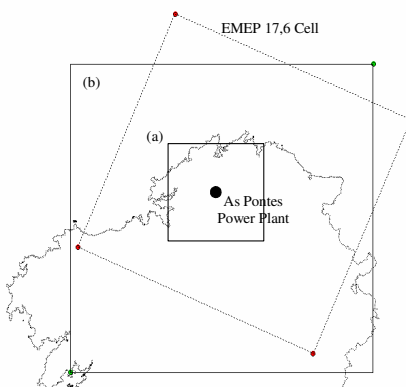


12 pav. Apsemtų teritorijų modelio vizualizavimas

Šaltinis: L. Hluchy. Flooding Crisis Team Support (2002) CrossGrid

3. Vandenyno bangų prognozavimas įvairaus sudėtingumo modeliais

4. Oro taršos modeliavimas pagal STEAM II modelį. Tai vienas iš pačių sudėtingiausių oro švaros modelių. Parašyta Fortran programavimo kalba. Pav. 13 Pavaizduota oro taršos modeliavimas aplink „AsPontes“ Ispanijoje šiluminę elektrinę.



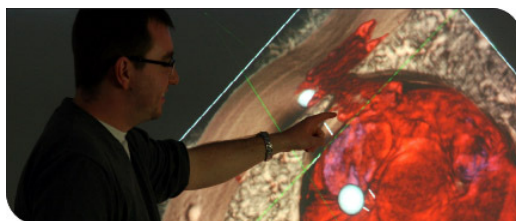
13 pav. Šiluminės elektrinės poveikio aplinkai spindulio vizualizavimas

Šaltinis: B. Jakubiak. Meteorological Applications (2002) CrossGrid

▪ **Praktinės fizikos.** Nagrinėjami ir kuriami praktinės fizikos uždaviniai pavyzdžiui vizualizuojamas įvairių dalelių judėjimas erdvėje.

Iš pateiktų CROSSGRID projekto pavyzdžių matome, kad *Grid* technologija naudojama įvairiose srityse. Šių sričių specialistai turėtų susipažinti su šiais modeliais ir jų taikymais praktikoje.

VIZNET – tai bendradarbiavimo tarp vizualizavimo centrų Jungtinėje Karalystėje (Uk) organizacija. Ji buvo sukurta tam, kad dalintis resursais, žiniomis mokant mokslininkus vizualizavimo srityje [33].



14 pav. Duomenų vizualizavimas VizNet

Šaltinis: [33]

Šioje organizacijoje yra atliekami įvairiausi tyrimai susiję su duomenų vizualizavimu.

BERKELEY LAB *angl. lawrence berkeley national laboratory* – tai nacionalinė vizualizavimo laboratorija esanti Jungtinėse Amerikos valstijose. Šios laboratorijos tikslas asistuoja mokslininkus siekiant mokslinių tikslų, sprendžiant pasaulio problemas per duomenų vizualizavimą ir analitiką [34].

Grafikos ir lygiagrečiųjų skaičiavimų institutas (*angl. Institute of Graphics and Parallel Processing*) – šios organizacijos pajėgos telkiasi ties moksliniais išradimais ir grafikos ir kompiuterių lygiagretaus skaičiavimo mokymu, tiria ir jų tarpusavio ryšius. Lygiagretiems skaičiavimams yra naudojamos multi procesorinės architektūros ir masinės lygiagrečios sistemos tokios kaip *Grid* skaičiavimų tinklai.

Atliekami tyrimai:

1. Kompiuterinė grafika

1. Okliuzijų vėsimas;
2. Aparatūrinės įrangos testavimas, egzaminavimas. Pvz.: „vertex“ šešėlių;
3. Realus laiko vaizdavimas. Pvz.: debesų, rūko;
4. Trimačių tinklų *angl. mesh* algoritmų optimizavimo tyrimai;
5. Įvairios realaus laiko praktinės sistemos;
6. Mokslinis vizualizavimas;
7. Stebėjimas ir derinimas lygiagrečiųjų skaičiavimų sistemų;

8. Saugus mokymas virtualiosiose terpėse ir t.t.
2. *Lygiagretieji skaičiavimai*
 1. Austrian *Grid* sukūrimo iniciatyva;
 2. Interaktyvus ir nuotolinis vizualizavimas naudojant „Grid“;
 3. Trimačių tinklų vaizdavimas naudojant multi procesorines mašinas;
 4. Duomenų išdalijimas naudojant HPF tipo kalbas. Pvz HPF, MPI, MPI2 ir t.t.;
 5. Programinės įrangos kūrimas lygiagrečiosioms sistemoms;
 6. Lygiagretieji algoritmai.

Taigi šioje organizacijoje atliekami įvairūs tyrimai ir mokymai, bet įgyvendinami projektai. Tačiau mums aktualiausi tik vizualizavimo naudojant *Grid* technologiją projektai, vienas iš tokių projektų „Gvid“ projektas.

Atviroji *Grid* sąsaja programoms vykdyti *Grid* tinkle yra labai ribota. Terminalinėms programoms pavyzdžiui „SSH“ įrankis „glogin“ yra vienas iš sprendimų, kuris leidžia interaktyviai naudoti *Grid* resursus. Tačiau pasak projekto šaltinio nėra tokios priemonės grafinėje terpėje „GUI“ *angl. graphical user interface*.

Galiausiai interaktyvioms programoms reikia specialių *Grid* vaizdo servisų, kurie įgalina transportuoti vaizdinę medžiagą paimant ją iš X11 programos naudojančios „OpenGL“ (atviroji grafinė biblioteka).

Pagrindiniai *Grid* technologijos tikslai visada buvo įgalinti didelio našumo programas, tokias kaip simuliacija ir jų duomenų vizualizacija.

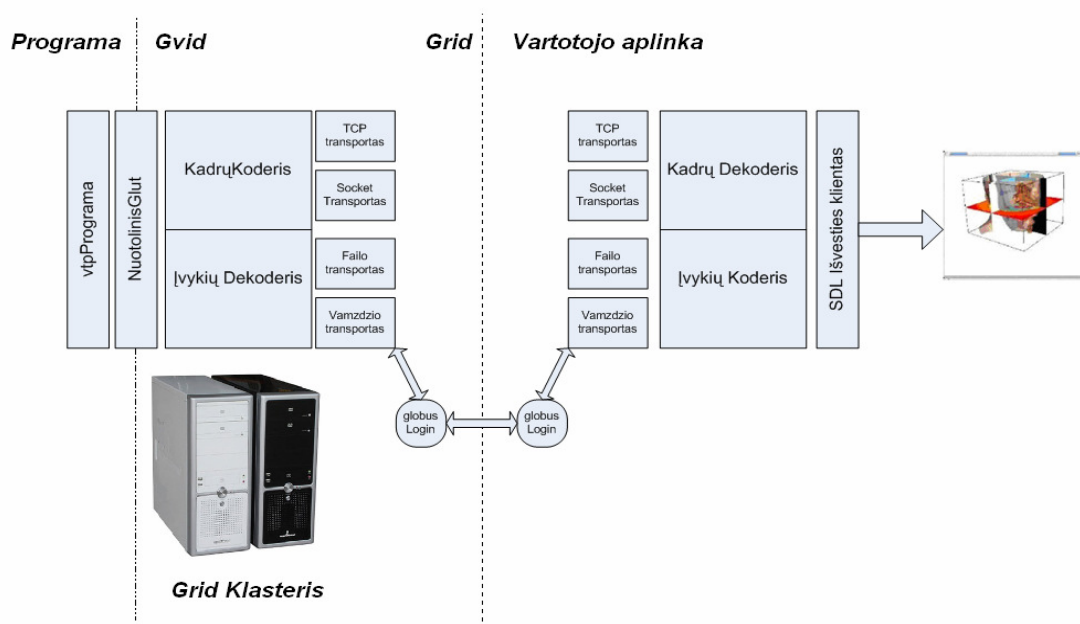
Skaičiavimo galia vizualizavime išauga tada, kai atsiranda galimybė vartotojui interaktyviai keisti simuliacijos ir vizualizavimo parametrus. Kai atliekamos tokios programos *Grid* tinkle, reikia apgalvoti kaip skirstyti vaizdavimą lygiagrečiai paskirstytiems skaičiavimams, kuriuos skiria tinklo atstumai ir todėl susidaro vėlavimo problemos.

Gvid projektas atlieka visas vizualizavimo užduotis išskyrus galutini vaizdavimą *Grid* tinkle. Tik efektyviai ir labai suspaustas vaizdo srautas yra perduodamas tinklu ir rodomas vaizdo klientuose.

Gvid struktūra. Skirstoma į tris dalis (žr. 15 pav.):

1. Įėjimo adapteriai *angl. input adapters*.
2. Transporto sluoksnį *angl. transport layer*. Transporto lygis rūpinasi, kad tarp įėjimo ir išėjimo adapterių būtų abipusis ryšys.
3. Išėjimo adapterius *angl. output adapters*. Tai yra galutinis interfeisas vartotojui rodoma dekoduoti vaizdinę medžiagą ir gaunama vartotojo interakcija, vaizdo valdymas, kuris grįžta atgal į programą.

Kompaktiniame diske priede rasite vaizdinę medžiagą, kurioje rodoma kaip naudojant „sony psp“ įrenginį yra dirbama su trimate grafika naudojant Gvid vizualizavimą.



15 pav. Gvid sandara

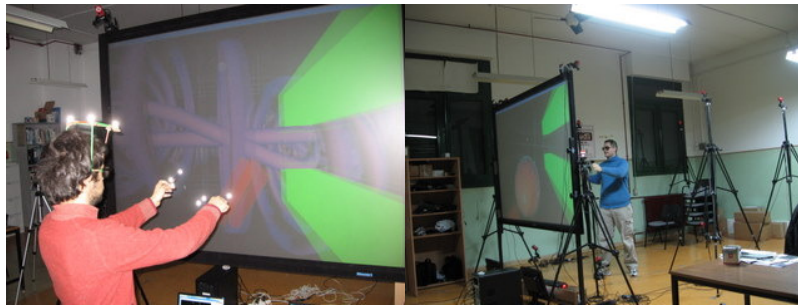
Šaltinis: [49]

Kiekvienai iš šių dalių yra priskiriama speciali programinė įranga, kuri turi būti įdiegta klasteryje. Įsidiegus ją galima nuotoliniu būdu prisijungus interaktyviai dirbti grafinėje terpėje priartinti vaizdą, patolinti, pasukti kampu, apversti ir t.t. Tai galinga vizualizavimo priemonė.

Sistemos trūkumas tas, kad specializuota programinė įranga turi būti įdiegta ne tik kliento pusėje, tačiau ir klasterio pusėje, bei realiuoju laiku perduodamų duomenų srautui reikalingas labai spartus duomenų perdavimas tinklu. Daugiau informacijos projekto svetainėje [49].

Bioskaičiavimu ir kompleksinės fizikos sistemų institutas (angl. *institute for biocomputation and physics of complex systems*) – „Bifi“ sukūrė savo 3D stereo sistemą, siekiant vizualizuoti įvairių rūšių modeliavimo rezultatus.

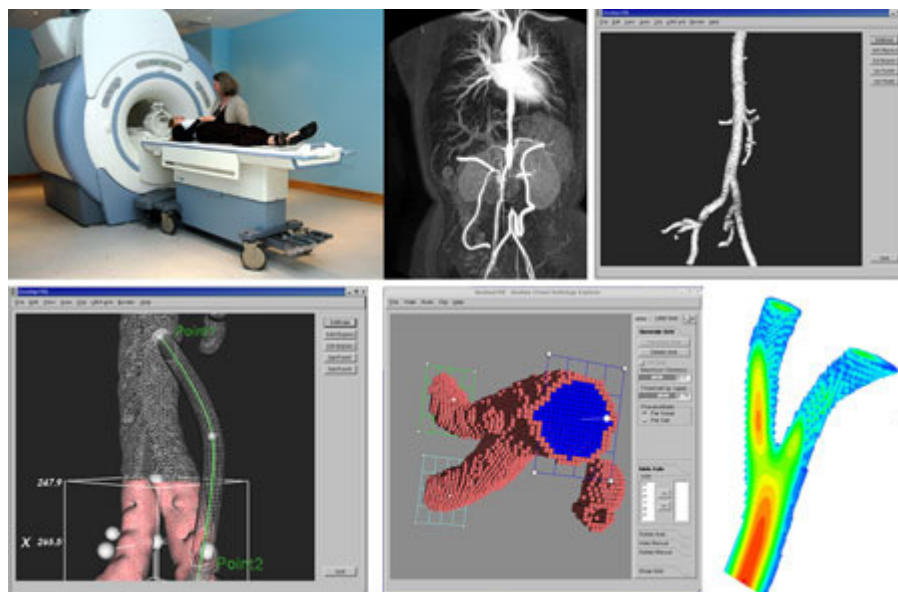
Diegimą apima du projektoriai ir polarizuotas ekranas. Taip pat būtini specialūs polarizuoti trimatės erdvės vizualizavimo akiniai. Kai naudojate retro projekciją rodomi stereo vaizdai. akiniai. Šiuo metu organizacijoje aktualiausi sintezės ir biocheminiai tyrimai.



16 pav. Vizualizavimo laboratorija „Bifi“ organizacijoje
Šaltinis:[36]

Kaip jau buvo minėta anksčiau vizualizavimą labai dažnai naudoja Universitetinės ligoninės įvairiems tyrimams atlikti. Vienas iš tokių pavyzdžių virtualus kraujagyslių tyrimas [36].

Virtualus kraujagyslių gydymas naudojant Grid - angl. *Virtual Vascular Surgery on the Grid*, kartais ligoninei neužtenka vien tik jos turimų skaičiavimo resursų. Amsterdamo Universitete demonstruojamas šis gydymas,naudojant „Grid“ servisus 17 pav.



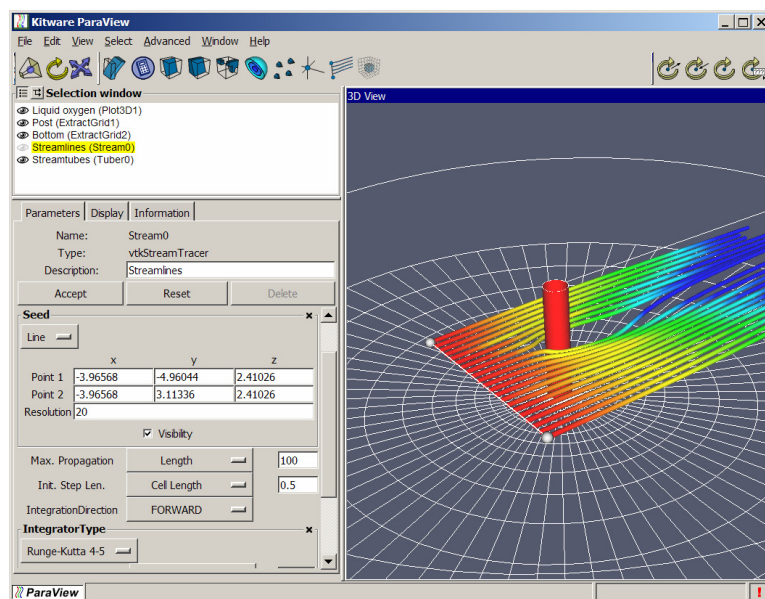
17 pav. Vizualaus kraujagyslių gydimo proceso etapų iliustracija

Aukščiau 17 pav. pateiktuose paveikslėliuose pavaizduota kraujo tėkmės simuliacija ir vizualizacija naudojant *Grid* resursus. Kairėje kampiniame paveikslėlyje pacientas yra skenuojamas Niderlanduose, po to gauta medžiaga siunčiama į saugyklą esančia Poznan, Latvijoje.

Ten medžiaga segmentuojama, filtruojama ir išskaidoma naudojant *Grid* servisus. Tada vaizduojamas kompiuterinis trimatis tinklas ir perduodamas lygiagrečios tėkmės analizatoriui, kuris veikia kompiuterių klasteryje esančiame Amsterdame, Ispanijoje.

Įvairių objektų vizualizavimui, galime naudoti viena kompiuterį, tačiau dideliems vizualizavimo uždaviniams atlikti neužtenka vieno kompiuterio naudojame lygiagrečius kompiuterius klasterius. Žinoma tam reikalinga ir specializuota programinė įranga. Vienas iš tokių projektų yra ParaView.

ParaView tai atvirojo kodo programa skirta lygiagrečiajam moksliniam vizualizavimui, sukurta organizacijos Kitware. ParaView yra sukurta vizualizacijos įrankio *angl. VTK* pagrindu, ir naudojanti lengvai naudojama Qt grafinį interfeisą [50].



18 pav. ParaView programos langas

Šaltinis:[50]

Ši programa palaiko keletą veikimo būdų:

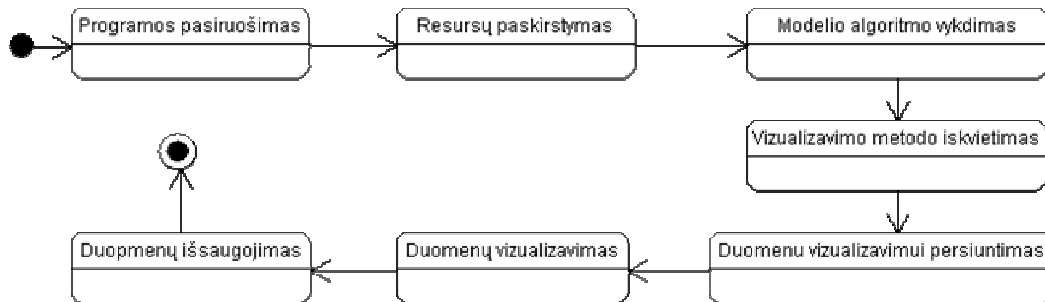
- *Lokalus klientas (angl. local client)*
- *Paprastas Serveris, Klientas – Serveris*
- *Vaizdavimo serveris (angl. render server), klientas – serveris.*
- *Grupinis vaizdavimas (angl. batch)*

Nesunku numanyti, kad serverio pusėje yra naudojamas specializuotą programinę įrangą aprūpintas telkinys ir žinoma užduotims išskaidyti MPI biblioteka. Duomenys vizualizavimui programoje gali būti surinkti iš eilės šaltinių ir vaizduojami įvairiais būdais (*angl. volume rendering, angl. isocontouring, angl. isosurfacing*).

III. PROJEKTINĖ DARBO DALIS

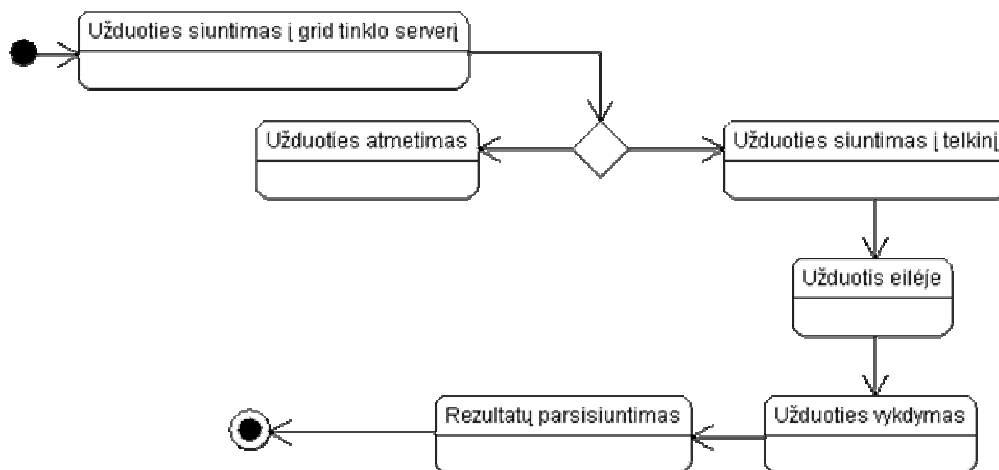
3.1. Darbinės srities modelis

Projekto darbinės srities būsenų diagrama 19 pav.



19 pav. Projekto darbinės srities būsenų diagrama

Šaltinis: sudaryta autoriaus



20 pav. Užduoties siuntimo ir vykdymo *Grid* telkinyje būsenų diagrama

Šaltinis: sudaryta autoriaus

3.2. Įrankių ir priemonių pasirinkimo analizė

Šiame darbe įrankius ir priemones pasirinkti nebuvo sunku. *Grid* klasteriai yra išnaudojami tada, kai norimą spręsti uždavinį skaidome į smulkesnius uždavinius ir juos efektyviai paskirstome naudojamam klasterio kompiuteriams.

Šiam darbui atlikti dažniausiai naudojamas MPI įrankis. MPI yra žinučių perdavimo interfeisas *angl. message passing interface*.

Pradedant naudoti *Grid* technologiją mokymo procese reikėtų nuo pat pradžių mokyti MPI bibliotekos naudojimo, nes tik šis įrankis suteikia galimybę efektyviai išnaudoti

klasterio resursus. Programos rašymas naudojant MPI biblioteka bei naudojant *Grid* resursus yra sudėtingas procesas.

Pirmiausiai programa projektuojama, po to atliekamas programos vertimas į programinį kodą, galiausiai programą kompiliuojama, vykdoma ir taisomos padarytos klaidos.

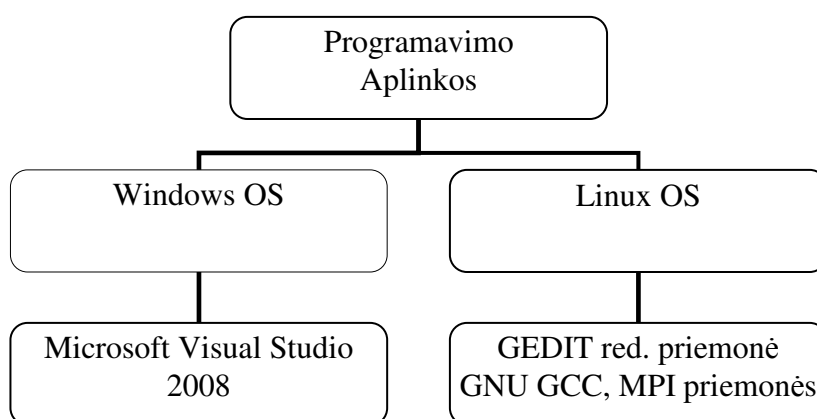
Praktiniam darbui atlikti buvo pasirinktos dvi programavimo aplinkos: Windows operacinės sistemos aplinka ir *Grid* vartotojo interfeiso *angl. user interface* (UI) aplinka.

Programuojant pasitaiko klaidų, o šių klaidų taisymas Windows aplinkoje spartus, tačiau *Grid* aplinkoje programavimo klaidų taisymas lėtas, nes programos išeitį tekstą reikia nusiųsti į užduočių vykdymo serverį, jame užduotis „pastatoma“ į eilę. Užduotis eilėje gali stovėti 1 minutę ar valandą, tačiau jei serveris labai apkrautas, tai eilė gali būti labai ilga.

Kol užduotis nėra įvykdyta, negalima pamatyti klaidų. Windows aplinkoje programos vykdymas atliekamas lokaliame kompiuteryje, eilės laukti nereikia, galime kompiliuoti programą ir klaidas gausime per keletą sekundžių, todėl programos funkcijų veikimui tikrinti galime naudoti windows operacinės sistemos aplinką 21 pav.

Programą galima rašyti keliomis programavimo kalbomis tai yra *Java*, *C*, *C++* ir *Fortan*.

Darbui atlikti buvo pasirinkta *C++* programavimo kalba.



21 pav. Programavimo aplinkos

Šaltinis: Sudaryta darbo autoriaus

Windows operacinės sistemos programavimo aplinkoje programų moduliams kompiliuoti, buvo pasirinktas Microsoft Visual Studio 2008 kompiliatorius, šis pasirinkimas nebuvo labai svarbus, nes mums reikėjo paprasčiausio patogaus kompiliatoriaus. Žinoma galima naudoti GNU GCC kompiliatorių, kuris įdiegtas Linux grafinėje aplinkoje, tačiau Visual Studio 2008 patogesnė priemonė, nes turi grafinę klaidų taisymo terpę.

Linux klasterio grafinėje aplinkoje, gali būti UNIX šeimos operacinės sistemos. *Grid* technologijos telkiniuose dažniausiai būna Scientific Linux ir tai priklauso nuo *Grid* klasterio administratoriaus arba organizacijos, kurioje šis klasteris yra.

Pirmoje Windows operacinės sistemos aplinkoje naudojame Microsoft Visual Studio 2008 ir Windows XP operacinę sistemą (šioje dirbame), o antroje grafinėje klasterio aplinkoje „Scientific Linux“ operacinę sistemą ir tekstinį redaktorių Gedit, kuris turi C++, C programavimo kalbos „spalvinimo“ funkciją.

Grafinėje klasterio terpėje, gali būti įdiegta tik Glite varotojo prieigos prie klasterio resursų programinė įranga ir to pakaktų užduoties siuntimui, tačiau „ui.grid.su.lt“ terpėje kurią mes naudojame, yra įdiegtos MPI biblioteka ir GNU GCC programos kompiliavimo priemonės. Aplinkoje pasigendama C++ su MPI biblioteka kompiliavimo priemonės, kuri leidžia kompiliuoti C++ programą naudojant MPI biblioteką. Paprastai C programavimo kalbos kompiliatoriaus su MPI biblioteka iškvietimas vykdomas „mpicc“ komanda, o „C++“ kompiliatoriaus mpiCC.

Linux aplinkai lokaliame kompiuteryje (UIPnP paketo diegimui) galime naudoti įvairius nemokamų operacinių sistemų paketus tai yra Scientific Linux, kuri veikia daugumoje klasterių, Slackware, Debian ir daug kitų operacinių sistemų. Tačiau norėdami į ją įdiegti UIPnP priemonę keliami specifiniai reikalavimai:

- Yra platinamos kelios UIPnP paketo versijos, tačiau darbo metu pasirinkta UIPnP 3.1 versija. [4].
- Remiantis literatūros šaltiniu [4] šis paketas dirba Scientific Linux, Suse, Debian operacinėse sistemose. Darbo autorius pastebi, kad išvardinti reikalavimai neatitinka tikrovės. UIPnP paketo diegimo metu, buvo nustatyti papildomi reikalavimai, pavyzdžiui reikalinga:
 1. Glibc 3 bibliotekos versija;
 2. bei kiti paketai pvz.: openssl paketas, kurio Ubuntu 5.01 operacinėje sistemoje nebuvo.

Remiantis šiais reikalavimais buvo pasirinkta Ubuntu operacinė sistema. Šiuo metu ji yra labai patraukli tarp vartotojų, nes ubuotu operacinėje sistemoje yra pagrindiniai ir reikalingiausia programų paketai, tačiau yra galimybė paprastai ir greitai įsidiegti įvairius kitus reikalingus paketus (synaptic packet manager).

Naujausioje Ubuntu versijoje glibc bibliotekos versija neatitinka reikalavimų (ji naujesnė), todėl buvo pasirinkta ir įdiegta senesnė Ubuntu 5.01 operacinė sistema ji turėjo reikalingas, senąsias bibliotekas.

UIPnP pakete yra viskas ir GliteUI ir Globus ir LCG priemonės, ir šio paketo diegimui pakanka paprasto vartotojo teisių Linux operacinėje sistemoje.

Taigi lentelėje pateikta trumpa naudojamų priemonių suvestinė.

Priemonių ir įrankių suvestinė

Operacinės sistemos	Windows XP	„ui.grid.su.lt“	Linux Ubuntu 5.01
<i>Redaktoriai</i>	<i>MS Visual Studio 2008</i>	<i>Gedit redaktorius, Scientific Linux</i>	<i>Gedit redaktorius</i>
<i>Kompiliatoriai</i>	<i>MS Visual Studio 2008</i>	<i>openMPI, GNU GCC ir kt.</i>	<i>GNU GCC</i>

Šaltinis: sudaryta darbo autoriaus

Reikia paminėti, kad kompiuteriams keliami reikalavimai nėra dideli:

- Windows XP ir Linux Ubuntu 5.01 aplinkoms su atsarga pakanka 1.6 Ghz spartos procesoriaus, 384 Mb operatyviosios atminties, 10 Gb vietos kietajame diske. Žinoma turi būtinai būti tinklo plokštė, bei prieiga prie interneto, tam kad kompiuteris galėtų palaikyti ryšį su *Grid* skaičiavimų telkiniais [5,6,7].
- Ui.grid.su.lt yra nuotolinė prieiga, todėl kompiuterio reikalavimai neaktualūs.

Anksčiau darbe buvo rašoma, kad reikia iš pradžių programą suprojektuoti, todėl tam reikalinga priemonė.

Šito darbo atlikimui pasirinkome laisvai platinamą, nemokamą programą *ArgoUML*. Tai UML diagramų kūrimo priemonė [8].

Prisijungimui prie ui.grid.su.lt taip pat reikalinga speciali programa palaikanti VNC protokolą, šiam darbui atlikti buvo pasirinkta nemokama *tightVNC* programa [9].

Duomenims ir programos išeities tekstams iš Windows XP aplinkos į ui.grid.su.lt aplinką persiuntimui, buvo pasirinkta winSCP programa, kuri jungiasi į „ui.grid.su.lt“ naudodama telnet failų persiuntimo – tftp protokolą, sudaromas abipusis ryšys [10].

Įrankių ir priemonių pasirinkime analizėje buvo, išskirtos trys *Grid* technologijos naudojimo aplinkos, todėl vartotojas gali pasirinkti jam tinkamiausią aplinką.

3.3. Projekto aprašymas

Kaip jau minėta duomenims vizualizuoti galime naudoti dinaminį vizualizavimą, animuotą vizualizavimą ir statinį vizualizavimą. Tenkinant užsibrėžtus uždavinius, kursime priemonę statiniam duomenų vizualizavimui. Primename statinis vizualizavimas, tai statinių duomenų vizualizavimas, kuomet gaunamas statinis vaizdas.

Remiantis šaltiniais *Grid* aplinkoje, be papildomų servisų įsikišimo, nei statinio, nei dinaminio vaizdo tiesiogiai, per grafinę aplinką gauti negalima.[11,12,13,14]

Naudojant įprastas Glite priemones uždavinių rezultatai ir informacija gaunama per failus, kurie po skaičiavimo parsiunčiami iš telkinio į vartotojo kompiuterį [1]. Dažniausiai naudojami tekstiniai failai rezultatams gauti, tačiau galime sukurti ir grafinius failus, todėl galime pateikti grafinę informaciją.

Šiame darbe nagrinėjamo **praktinio darbo idėja** ir buvo statinio vizualizavimo, panaudojant grafinį failą kūrimas.

Projekto motyvacija. Aukščiau buvo paminėta, kad duomenų vizualizavimui *Grid* aplinkoje yra naudojama specializuota programinė įranga.

LitGrid tinkle šiuo metu yra viena galimybė duomenims vizualizuoti naudojant MATH_R paketą. Paketų sąrašą galime pamatyti Balticgrid infosite svetainėje, pasirinkto telkinio programinės įrangos aprašo *angl. software environment* [15].

MATH_R tai statistikos matematinių skaičiavimų paketas, kuriame yra savita programavimo kalba. MATH_R programavimo kalbos sintaksė panaši į „MATHCAD“ daugeliui žinomo matematinio skaičiavimų paketo sintaksę. Vizualizavimas MATH_R pakete galimas tik dvimačio grafiko pavidalu, bei daugiausiai apsiriboja matematinių rezultatų pateikimu, kurie gaunami naudojant MATH_R programavimo kalbą.

Sudėtingų procesų ir algoritmų įgyvendinimui šios programavimo priemonės nepakanka, todėl yra naudojamos kitos priemonės. Viena iš jų C++ programavimo kalba. Ši programavimo kalba yra universali, ji leidžia įgyvendinti praktiškai bet kokią sugalvotą algoritmą ir naudojama mokymo procese. Todėl praktinio darbo įgyvendinimui, statinio vizualizavimo priemonės kūrimui, buvo naudojama C++ programavimo kalba. Šia kalba aprašomi vizualizavimo metodai bei uždavinių sprendimo algoritmai.

Tokio praktinio darbo mūsų žiniomis Lietuvoje niekas nekūrė, todėl šis darbas yra unikalus. Įvykdžius šį praktinį darbą gausime priemonę duomenų vizualizavimui, kurią galime panaudoti mokymo proceso metu.

Atliekant modeliavimą per lygiagrečius programavimo laboratorius darbus, duomenis galima būtų išvesti tiesiogiai grafinių failų pavidalu.

Kartais tenka tą patį vaizdavimo būdą kartoti keletą kartų, su skirtingais duomenimis, pasitelkus sukurtą priemonę būtų galima panaudoti *Grid* resursus ir vienu metu pavaizduoti keletą grafikų, tai yra išskaidant užduoti *Grid* telkinio kompiuteriams arba atliekant vieną didelių matmenų duomenų vizualizavimą pasitelkti keletą klasterio procesorių išskaidant sudėtingą užduotį į smulkesnes. Vykdam užduotis lygiagrečiai gaunama didesnė sparta.

Planuojami atlikti darbai:

1. Pasiruošimas darbui tai susipažinimas su ui.grid.su.lt aplinka;
2. „Glite“ diegimas lokaliame kompiuteryje;

3. Susipažinimas su MPI programavimu, elementarios programos bandymas;
4. Vizualizavimo programos projektavimas;
5. Modulių kūrimas Microsoft Visual Studio 2008 programavimo aplinkoje;
6. Programos galutinis įgyvendinimas *Grid* klasteryje, naudojant MPI biblioteką;
7. Rezultatų analizė.

3.4. Aplinkos pasiruošimas darbui

Šiam darbui atlikti reikalinga prieiga prie *Grid* resursų. Tam, kad prieiti prie *Grid* resursų reikia prisijungti prie *Grid* organizacijos ir gauti vartotojo prisijungimo sertifikatą. Šiam darbui atlikti darbo autoriui buvo suteikta galimybė naudoti LitGrid organizacijos teikiamus resursus.

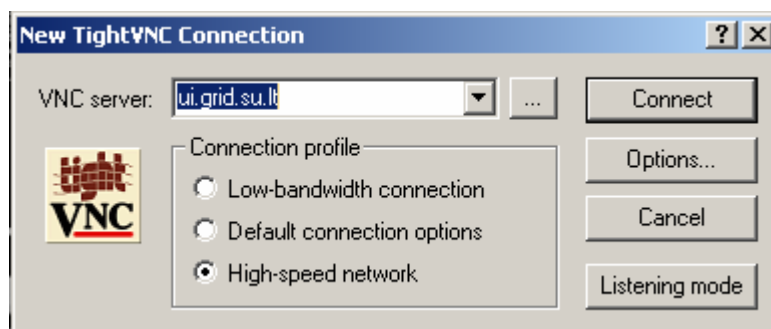
Turint sertifikatą galimi keli prieigos variantai:

1. Prisijungti per *Grid* telkinio grafines vartotojo aplinkos „UI“ pvz. „ui.grid.su.lt“;
2. Naudoti Migration Desktop ar kitą prieigos priemonę [40];
3. Įsidiegti Linux operacinę sistemą savo kompiuteryje ir įdiegti gliteUI vartotojo aplinką.

Šiame darbe buvo įgyvendintos dvi galimos prieigos pirmoji per „ui.grid.su.lt“ aplinka ir trečioji įsidiegiant glite vartotojo aplinką stacionariame ir virtualiame kompiuteryje.

Šiaulių Universiteto klasteryje yra įdiegta bendro naudojimosi grafinė aplinka „UI“ tad paprasčiausia išbandyti *Grid* aplinka buvo būtent pasinaudojant šiuo grafiniu interfeisu 23 pav.

Prisijungimui prie Šiaulių universiteto *Grid* telkinio grafines vartotojo aplinkos 22 pav. kaip darbe jau buvo minėta naudojama tightVNC programa. Prisijungimo prie ui.grid.su.lt naudojant tightVNC programos langas pateiktas 22 pav.



22 pav. TightVNC programos pradinis prisijungimo langas

Šaltinis: sudaryta darbo autoriaus

```

[ansa@ui hello]$ dir
ex2.jdl  ex2.jdl~  ex2.sh  rezult  source.c  status
[ansa@ui hello]$ glite-wms-job-submit --vo litgrid -a -o job.list ex2.jdl
Warning - --vo option ignored
Connecting to the service https://lxb026.mif.vu.lt:7443/glite_wms_wmproxy_server
Warning - Following Warning(s) found while parsing JDL:
NodeNumber attribute is going to be deprecated very soon. Use CpuNumber instead
===== glite-wms-job-submit Success =====
The job has been successfully submitted to the WMPoxy
Your job identifier is:
https://lxb026.mif.vu.lt:9000/mIChmhoSVpoZsv08v8ZT5g
The job identifier has been saved in the following file:
/home/ansa/hello/job.list
=====
MPI_Status stat;
*CEId*
- grid9.mif.vu.lt:2119/jobmanager-lcgpbs-long
- grid9.mif.vu.lt:2119/jobmanager-lcgpbs-short
- grid9.mif.vu.lt:2119/jobmanager-lcgpbs-verylong
- pupa.eLEN.ktu.lt:2119/jobmanager-pbs-litgrid
- ce.grid.vgtu.lt:2119/jobmanager-lcgpbs-litgrid
- dane.ku.lt:2119/jobmanager-pbs-litgrid
- grid6.mif.vu.lt:2119/jobmanager-lcgpbs-long
- grid6.mif.vu.lt:2119/jobmanager-lcgpbs-short
- grid6.mif.vu.lt:2119/jobmanager-lcgpbs-verylong
=====
LIST
irements have been found:

```

23 pav. „ui.grid.su.lt“ aplinka

Šaltinis: sudaryta darbo autoriaus

UIPnP diegimas

UIPnP tai pernešama Glite,Globus programinės įrangos versija. Ši aplinka veikia Linux operacinėse sistemose:SLC3, Fedora 3, Debitan, Suse, Ubuntu.

Paketo diegimas skaidomas į kelis etapus:

1. Paketo parsisiuntimas iškleidimas

Paketo parsisiuntimui terminale vykdoma komanda:

```
< wget http://voms-web.balticgrid.org/UIPnP-BG/UIPnP-balticgrid.glite.tar.gz>
```

Programos išpakavimas vykdomas komanda:

```
< tar -xvzf UIPnP-3.1-0.tgz>
```

2. Sistemos konfigūravimas

Pagal nutylėjimą, šis paketas turėtų veikti BalticGrid Grid tinkle, o Litgrid tinklas priklauso BalticGrid tinklui, tačiau praktika parodė, kad reikalingas papildomas paketo konfigūravimas.

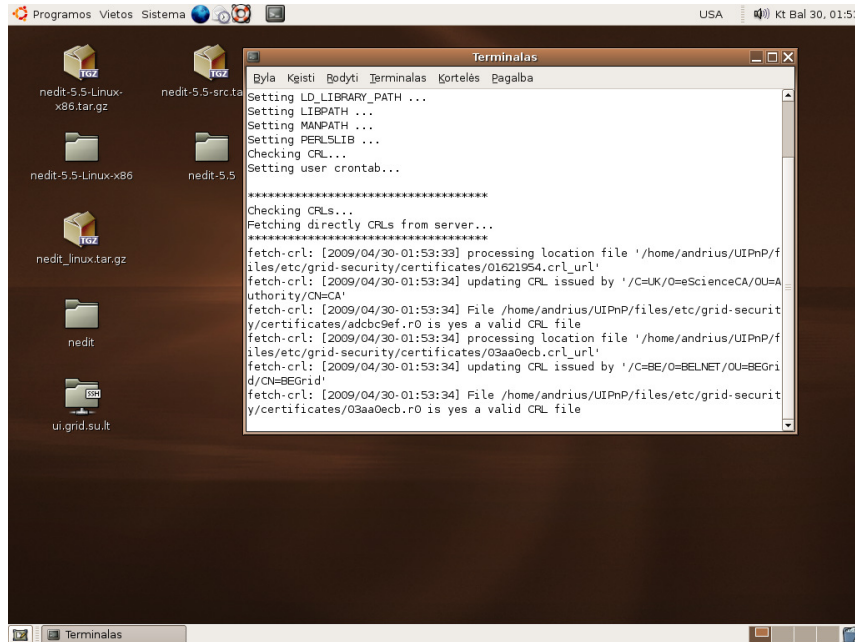
3. Paketo UIPnP paleidimas

Šio UIPnP paketo paleidimui teoriškai tereikia, įvykdyti diegimo skriptą UIPnP.sh.

Skripto paleidimui komandiniame lange įrašoma tokia komanda:

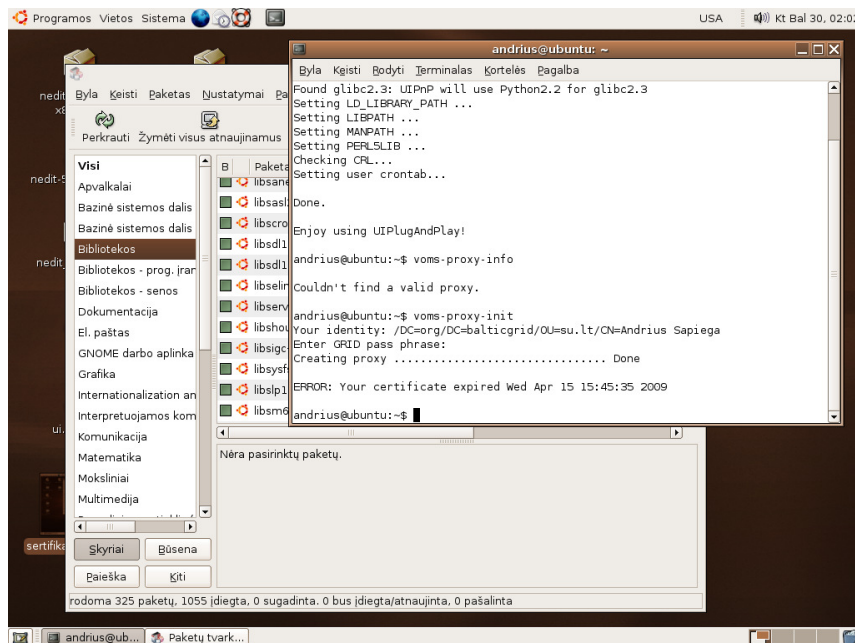
< source \$UIPnP/UIPnP.sh >

UIPnP priemonė buvo įdiegta Ubuntu 5.01 operacinėje sistemoje, 24 pav. matomas UIPnP paleidimo skripto vykdymas, lango apačioje matomi serverio sertifikatų sinchronizavimo pranešimai.



24 pav. UIPnP paleidimas. Gaunami sertifikatai.

Šaltinis: sudaryta darbo autoriaus

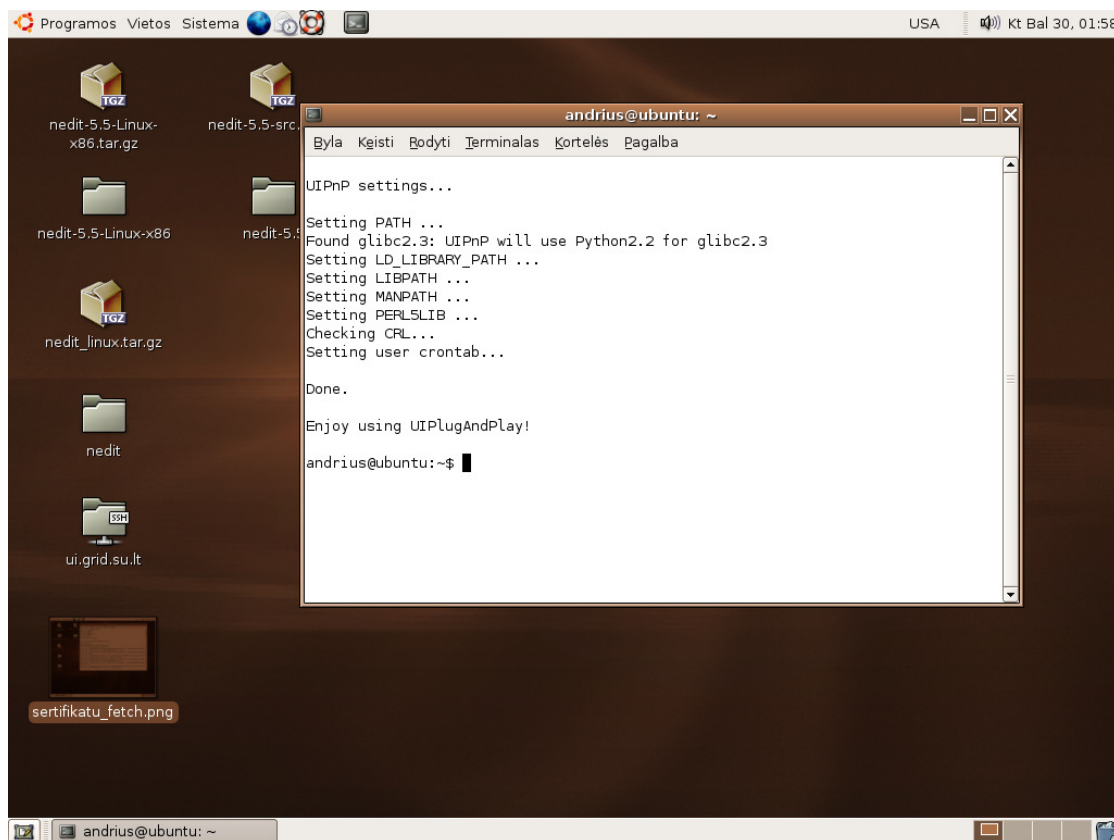


25 pav. Pasibaigęs sertifikatas

Šaltinis: sudaryta darbo autoriaus

Vartotojo sukurtas sertifikatas galioja metus, po metų jį reikia prasižesti, 25 pav. pavaizduotas UIPnP priemonės naudojimas ir gautas pranešimas apie pasibaigusį vartotojo identifikavimo sertifikatą.

UIPnP priemonės serverio sertifikatų sinchronizavimas vykdomas tik vieną kartą, todėl antrą kartą paleidus tą patį paketo paleidimo skriptą, jis įvykdomas greičiau ir praleidžiant sertifikatų sinchronizavimo procedūrą 26 pav.



26 pav. Automatinis UIPnP skripto įgyvendinimas

Šaltinis: sudaryta darbo autoriaus

Diegiant UIPnP buvo atlikti papildomi darbai:

- Esamos konfigūracijos perkėlimas iš `ui.grid.su.lt` į UIPnP paketo konfigūraciją;
- Iš Litgrid serverio sertifikatų pasiuntimas ir patalpinimas UIPnP priemonėje;
- Specialių bibliotekų instaliavimas Ubuntu Linux operacinėje sistemoje;
- Automatinis skripto paleidimo įkėlimas Ubuntu OS.

Automatinis UIPnP įkėlimui `/home/Andrius/.bashrc` failo gale buvo patalpintas paleidimo skriptas:

```
export UIPnP=/home/andrius/UIPnP
source $UIPnP/UIPnP.sh
cd $HOME
```

Kita kartą atsidarius terminalo langą, skriptas įvykdomas automatiškai 26 pav.

3.5. Programos projektavimas

Vizualizavimo skyriuje buvo žinoma, kad uždaviniams įgyvendinti turime sudaryti jų kompiuterinį algoritmą, šį algoritmą paversti kompiuteriui suprantama kalba, jį įvykdyti

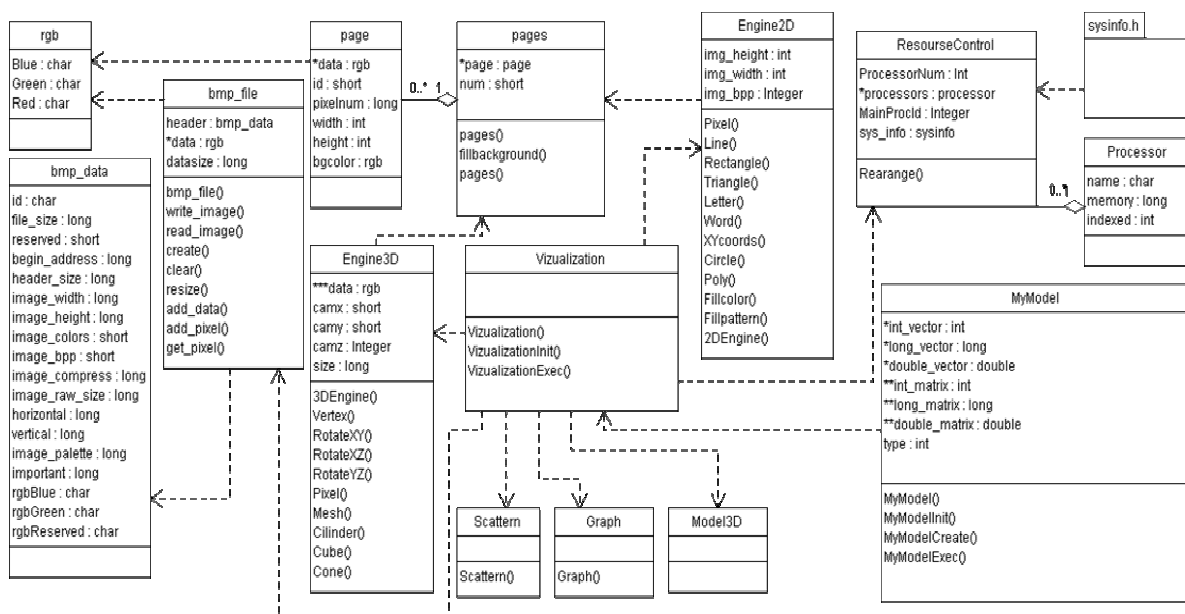
kompiuteryje. Naudojant gautus rezultatus ir vizualizavimo būdus duomenis pateikti grafiškai.

Programą iš principo išskaidoma į tris dalis, vienoje klasėje vykdomas algoritmas, kitoje vizualizavimo darbas ir t.t. Todėl 26 Pav. pateikta suprojektuota programos klasių diagrama. Ji nėra galutinė, nes atliekant programavimo darbus gali pritrūkti kokio metodo ir tuomet jis įterpiamas.

Viso dvimačio puslapio informacija saugoma dinaminiam vieno matavimo masyve „*data“. Taip pat saugomi raiškos parametrai bei visų taškų skaičius.

Šiam puslapiui saugoti buvo pasirinktas paprastas grafinis formatas t.y. Windows operacinės sistemos grafinis formatas *angl. windows bitmap*, kurio plėtinys „.bmp“.

Šis failas skaidomas į „header“, „paveikslėlio informaciją“, „spalvų paletę“ ir paveikslėlio duomenų failo dalį. Šiame darbe buvo naudojamas 24 bitų bmp formatas, todėl kiekvienam taškui skiriamos trys „char“ duomenų tipo reikšmės (RGB paletė), kiekviena spalva gali įgyti 256 reikšmes, o iš viso galime gauti $256 * 256 * 256 = 16,777,216$ atspalvių.



27 pav. Programos projekto klasių diagrama

Šaltinis: sudaryta darbo autoriaus

Aukščiau pateiktos klasių diagramos paaiškinimai:

1. Klasė ResourceControl

Šioje klasėje kaupiama informacija apie procesus, bei atmintį. Procesai gali būti tame pačiame fiziniame įrenginyje tuomet, gaunami atminties kiekiai yra bendri. Procesams šią atmintį reikia išdalinti. Klasė skirta procesų arba kitaip „nodų“ limitų apibrėžimams nustatyti.

2. Klasė MyModel

Šioje klasėje yra kuriamas ir vykdomas vartotojo sukurtas C++ kalba aprašytas algoritmas. Šis algoritmas kuriamas MPI bibliotekos pagrindu, išnaudojantis *Grid* klasterio skaičiavimo resursus. Gauti rezultatai persiunčiami į tą programos dalį kuri atsakinga už duomenų vizualizavimą.

3. Klasė Visualization

Vizualizavimo valdymo klasė, klasės diagramoje ji nėra išbaigta, nes yra daug duomenų tipų kuriuos ji galėtų priimti. Papildant programą nauju duomenų tipu, klasių diagramoje turėtų atsirasti papildomi metodai ir parametrai.

Ši klasė gauna signalus iš „MyModel“ klasės, kurie nusako koks vizualizavimo būdas bus naudojamas, kokie duomenų tipai bus perduoti. Klasė priima duomenis, išskviečia vizualizavimo funkcijas. Valdo vizualizavimo proceso eigą.

4. Klasės Engine2D, Engine3D

Šiose klasės yra dvimačio ir trimačio vaizdo „varikliai“. Šioje klasėse buvo numatyta kurti primityvu metodus, lapų valdymo, bei įvairiems parametrų saugoti.

5. Klasės pages, page

Page klasę sudaro puslapio parametrai, bei visų taškų masyvas. Šis masyvas yra dinaminis, todėl taškų skaičius yra kintantis. Pages klasėje apibrėžiami lapų klasės objektai.

6. Klasė Bmp_file

Darba su failais atliekanti klasė. Pagrindinė šios klasės užduotis saugoti sukurtą grafinį puslapį į bmp formato failą.

Klasių diagramą galima praplėsti, pavyzdžiui papildant failo saugojimo jpeg, png, ar net avi failo formatų klasėmis, papildant vizualizavimo modeliais ir t.t.

Kaip jau buvo rašyta anksčiau, programiniam produktui kurti buvo naudojama MPI biblioteka. Todėl prieš pradėdant kurti šią priemonę darbo reikia, susipažinti su paprasčiausia MPI programa. MPI biblioteka turi savo specifinę aplinką [26].

Žemiau pateikta pirmoji sveikas pasauli programa:

```
#include <mpi.h> // pridedama MPI programavimo biblioteka
#include <stdio.h>
```

```
int main(int argc, char *argv[]){
```

```
    // Vykdo visi procesai
```

```
    int numprocs; // kiek procesų
```

```
    int procnum; // proceso numeris
```

```
    // -----
```

```
    // Inicializuojame MPI biblioteka
```

```
    MPI_Init(&argc, &argv);
```

```
    // Gauname dabartinio proceso numeri
```

```

MPI_Comm_rank(MPI_COMM_WORLD, &procnum);

// Gaunam kiek procesų atlieka darba
MPI_Comm_size(MPI_COMM_WORLD, &numprocs);

// Vykdo tik 0 procesas
If (procnum == 0){
    Printf(„Aš nulinis“);
}
else{
// Vykdo visi procesai
printf ("Sveikas pasauli! Aš procesas %d iš %d procesų\n", procnum, numprocs);

// Užbaigiame MPI
MPI_Finalize();

// Vykdo visi procesai

// -----
return 0;
}

```

Iš pateikto pirmosios programos pavyzdžio matyti, kad išskirti atskirus procesus galima naudojant C++ programavimo kalbos „if“ komandos pagalba. Šiuo atveju galime įvykdyti užduotį tik 1 kartą.

Tačiau projektuojamoje programoje reikalingas keletas užduočių nuoseklus vykdymas, todėl tam buvo panaudotas C++ programavimo kalbos ciklo sakinyss „for“ ir sąlygos sakinyss „switch“. Ši programos kodo dalis pateikta žemiau kaip pavyzdys.

```

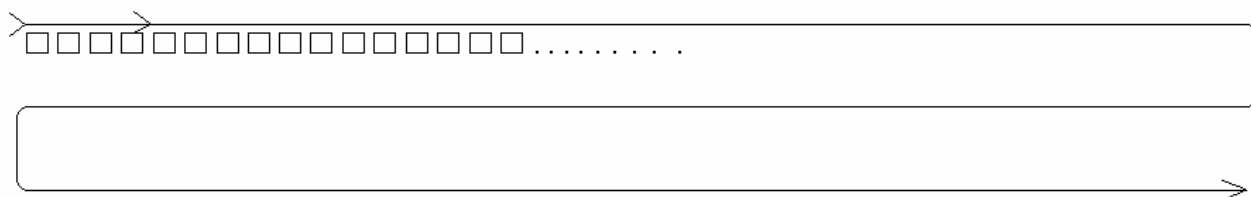
For (int i=0;i<=2;i++){
    Switch(i){
        Case:0: { // Užduotis 0
            If (procnum!=0){
                // Vykdo visi kiti procesoriai išskyrus 0
            }
            Else{
                // Vykdo nulinis procesas
            }
            break;
        }
        Case:1: // Užduotis 1 break;
        Case:2: // Užduotis 2 break;
    }
}

```

Programos vykdymo metu, yra vykdomas „for“ ciklas. Kintamasis „i“ įgyja reikšmes nuo 0 iki 2 ir reiškia užduoties numerį. Šiuo atveju gali būti išskirtos 3 užduotys. „Switch“ sakinio esančio cikle pagalba užduotys yra atskiriamos. Kai užduotis prieina eilę, užduoties viduje dar yra „if“ sąlyga, kurios pagalba užduotis išskaidoma procesams.

Atvaizdo formavimo principas. Šiame darbe vienam taškui aprašyti naudojama struktūra rgb, kuri nusako taško spalvą ir permatomumą angl. alpha channel. Šios struktūros masyvas *data yra taškų masyvas, tai yra visas formuojamas paveikslėlis. Šis masyvas yra

vienmatis, tačiau kai gaunamas paveikslėlis matomas dvimatis vaizdas. Taškai formulės pagalba verčiami iš dvimačio į vienmatį formatą. Jie saugomi faile, kurio saugojimo tvarka pateikta paveikslėlyje.



28 pav. Informacijos rašymo į failą tvarka

Šaltinis: sudaryta darbo autoriaus

Paveikslėlio visi taškai saugomi vienmačiame dinaminiame masyve. Tam, kad būtų galima piešti taškelį turime perversti iš (x,y) koordinatčių sistemos į vieno mato sistemą.

Pagrindinė formulė:

$$z = ((\text{plotis} * (y - 1)) + x) - 1;$$

Kur z yra koordinatė vienmačiame masyve, x, y yra taškai, o plotis tai paveikslėlio plotis pvz. 1024 taškai plotis, $x = 100$ $y = 200$ gauname $z = ((1024*(200-1))+100)-1 = 203875$.

Sekantis žingsnis po programos projektavimo yra programos įgyvendinimas. Įgyvendinant sukurta projektą iškilo įvairių problemų. Šios problemos aprašytos kitame poskyryje.

3.6. Problemų ir jų sprendimo aprašymai ir pagrindimai

1. *Grid telkinio grafinės aplinkos „UI“ problemos*

Naudojant „ui.grid.su.lt“ kartais vartotojas nebegali prisijungti prie grafinės aplinkos. Darbo autoriaus buvo pastebėta, kad užbaigiant prisijungimo sesiją, realiai programos neišjungiamos galbūt dėl konfigūracijos netikslumų ar pan. Kitą kartą jungiantis sukuriama tų pačių programų procesų kopijos taip atsiranda po 3 ir daugiau programų dublikatų. Viršijus leistinų sistemoje dublikatų limitą nebegalima prisijungti prie grafinės aplinkos. Taip atsitikus vartotojui pateikiama sistemos perspėjimo lentelė *angl. gnome panel is running* pranešanti apie tam tikros programos veikimo ir neleidžiama prieiga prie grafinės aplinkos per VNC protokolą. Tačiau jungiantis į klasterio terminalinę darbo aplinką t. y. prieiga per SSH protokolą, galime prisijungti. Per terminalinę prieiga galime valdyti sistemoje veikiančius procesus, todėl naudodami procesų valdymo komandas (pvz. `ps -U ansa` ir `kill`) išjungiamo po vieną paleistus programos procesus, išskyrus SSH prieigos ir terminalo programos

procesus. Po to pabandžius jungtis per VNC protokolą sistema vėl leidžia prisijungti ir iškilusi problema išspręsta.

2. UIPnP diegimo problemos

UIPnP universalus, tačiau „reiklus“ paketas. Paketo diegimo proceso metu, reikalinga Glibc2.3 ir kiti papildomu paketai. Dažniausia įdiegiant standartinę operacinę sistemą tų paketų nebūna, arba paketai būna naujesni ir nesuderinami su senesniais. Dėl šios priežasties, UIPnP buvo įdiegta į senesnę Ubuntu 5.01 versiją ir reikalingos bibliotekos buvo papildytos naudojant „Synaptic Packter Manager“ priemonę.

3. OpenMPI bibliotekos naudojimo problemos

Darbo praktinės dalies įgyvendinimo laiku, iškilo problema. Prieš savaitę veikusi programa praėjus kelioms dienoms nebeveikia. Programos vykdymas *Grid* telkinyje gražina klaidą, kurioje pranešama apie nerastą sukompiliuotą programą.

Po autoriaus bandymų išspręsti problemą savarankiškai ir konsultacijos su vienu iš Litgrid telkinių administratoriumi, pavyko išspręsti šią problemą. Šios problemos sprendimas aprašytas patarimų, pastebėjimų, rekomendacijų skyriaus 1 poskyryje.

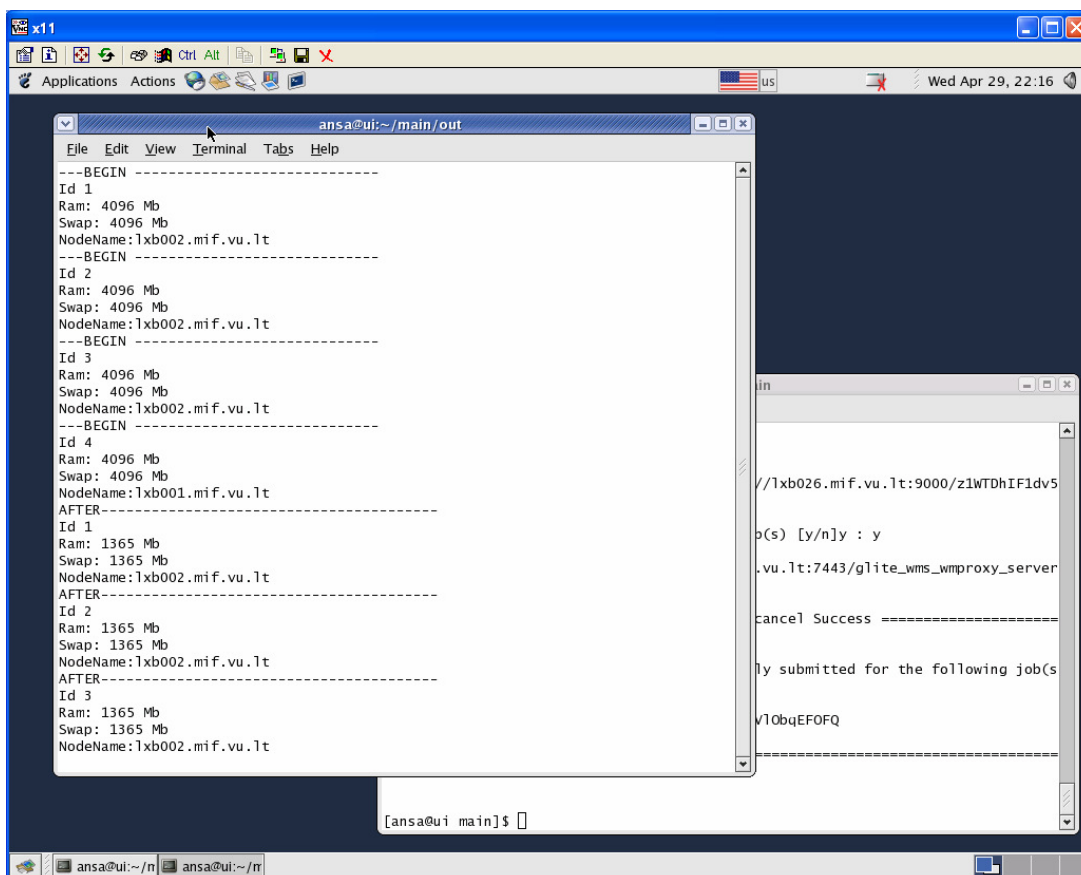
4. Sertifikato galiojimo problema

Darbe buvo rašoma, kad pasinaudoti *Grid* telkinio resursai reikalingas specialus vartotojo identifikacijos sertifikatas. Šis sertifikatas galioja 1 metus. Po metų jo galiojimas pasibaigia tuomet jį reikia pratęsti [1].

IV. DARBO REZULTATŲ ANALIZĖ

Darbo metu buvo sukurta priemonė duomenų vizualizavimui. Buvo sukurtas dvimačio duomenų vizualizavimo „variklis“ leidžiantis kurti grafinius primityvus (taškas, linija, keturkampis, apskritimas, elipsė ir kt.), bei sukurtas vienas iš paprasčiausių duomenų vizualizavimo metodų dvimatis grafikas.

Šiaulių Universiteto telkinio grafinė aplinka „ui.grid.su.lt“ pateikta žemiau esančiame 29 paveikslėlyje. Jame peržiūrimas įvykdytos programos rezultatas, čia buvo vykdomas vienas iš programinės priemonės modulių, atsakantis už telkinio operatyviosios ir laikinosios atminties paskirstymą procesams. 29 paveikslėlyje *Grid* telkinio elemente kurio vardas „lxb002.mif.vu.lt“ iš viso yra 4096 megabaitų operatyviosios atminties. Šiame elemente buvo sukurti 3 procesai ir jie visi praktiškai dalinasi bendra operatyviaja atmintimi, todėl 4096 Mb atmintį reikia dalinti į 3 dalis. Pateiktame paveikslėlyje žemiau po žodžio „poto“ *angl. after*, matome jau paskirstytus procesams atminties dydžius.

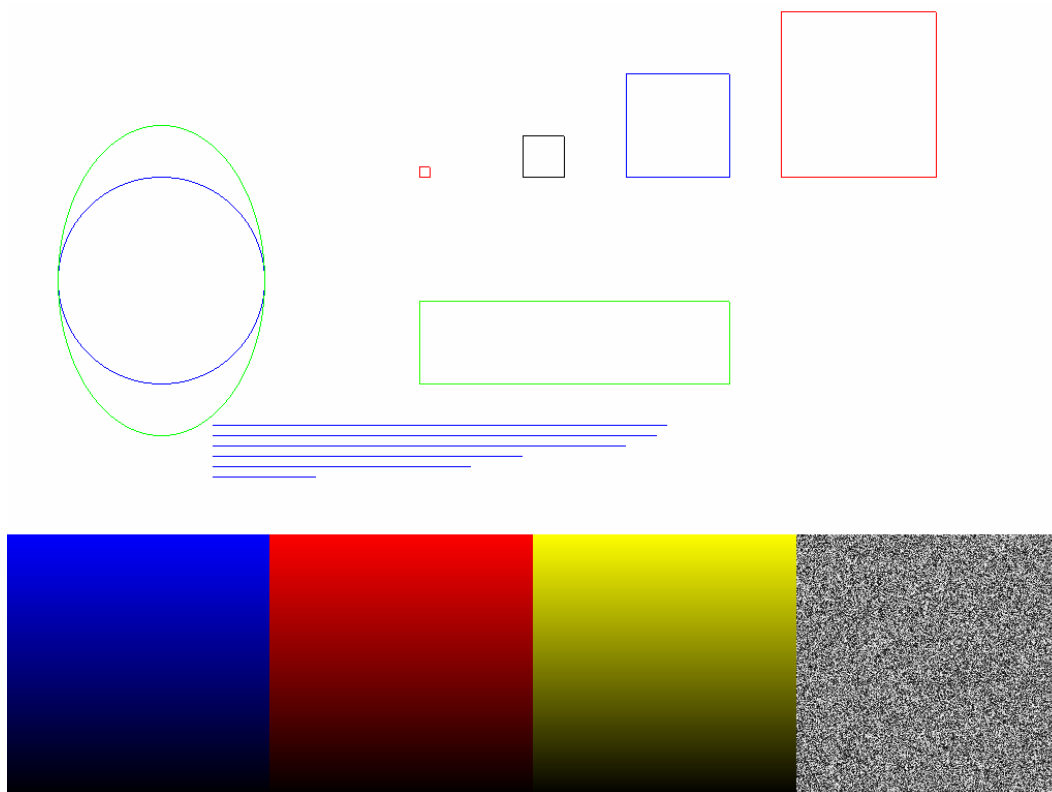


```
ans@ui:~/main/out
File Edit View Terminal Tabs Help
-----BEGIN-----
Id 1
Ram: 4096 Mb
Swap: 4096 Mb
NodeName:lxb002.mif.vu.lt
-----BEGIN-----
Id 2
Ram: 4096 Mb
Swap: 4096 Mb
NodeName:lxb002.mif.vu.lt
-----BEGIN-----
Id 3
Ram: 4096 Mb
Swap: 4096 Mb
NodeName:lxb002.mif.vu.lt
-----BEGIN-----
Id 4
Ram: 4096 Mb
Swap: 4096 Mb
NodeName:lxb001.mif.vu.lt
AFTER-----
Id 1
Ram: 1365 Mb
Swap: 1365 Mb
NodeName:lxb002.mif.vu.lt
AFTER-----
Id 2
Ram: 1365 Mb
Swap: 1365 Mb
NodeName:lxb002.mif.vu.lt
AFTER-----
Id 3
Ram: 1365 Mb
Swap: 1365 Mb
NodeName:lxb002.mif.vu.lt
[ans@ui main]$
```

29 pav. Resursų kontrolės modulis vykdomas grid telkinyje, atminties paskirstymas sukurtiems procesams

Šaltinis: sudaryta darbo autoriaus

Pirmiausiai priemonėje buvo sukurtas dvimatės grafikos variklis, leidžiantis kurti įvairius grafikos primityvus linija *angl. line*, taškelius *angl. pixel*, stačiakampius *angl. rectangle*, apskritimus *angl. circle* ir kt. Šių primityvu piešimas pateiktas 30 pav.



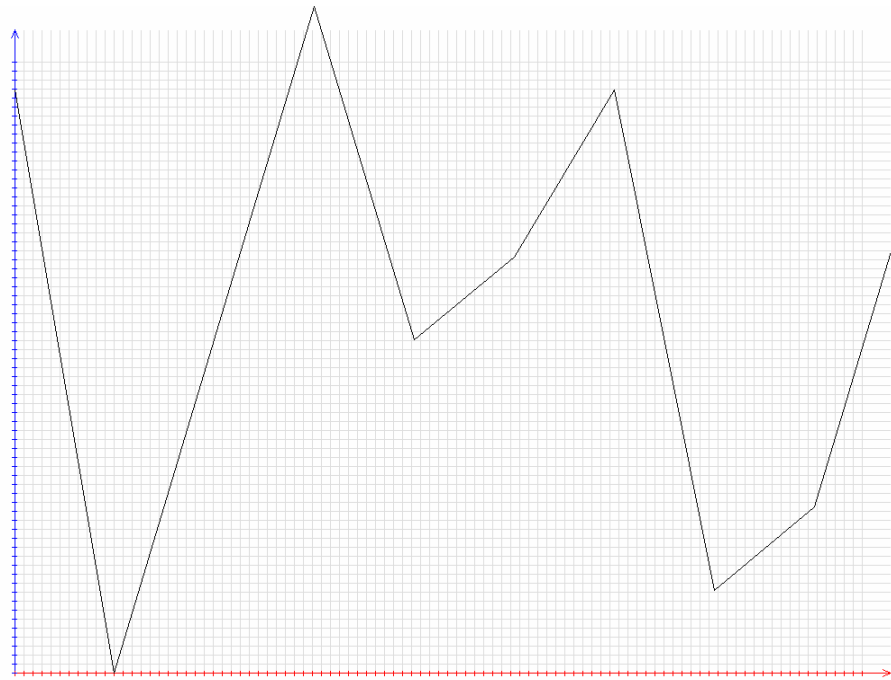
30 pav. Dvimačio vaizdo grafiniai primityvai

Šaltinis: sudaryta darbo autoriaus

Naudojant grafikos primityvus, galima kurti įvairius duomenų vizualizavimo metodus įvairius grafikus, histogramas, taškines, linijines diagramas bei nestandartinius vizualizavimo metodus pavyzdžiui karčio žemėlapi *angl. heat map*.

Darbo praktinėje dalyje buvo sukurtas paprasčiausio duomenų grafiko vizualizavimo metodas. Jo kūrimo eigą:

1. Apskaičiuojamos vizualizuojamu duomenų intervalų galai
2. Randama neigiamų, teigiamų skaičių, vienetų skaičius, nustatomi įvairūs parametrai pageidaujamas duomenų vaizdavimo intervalas, tarpai tarp jų ir t.t.
3. Piešiamos koordinatinių ašys ir sugraduojamos.
4. Pagal duomenų dydžius, jie sutankinami arba išdidinami.
5. Kiekvienam reikšminiam duomenų gabalėliui, apskaičiuojamos koordinatės ir naudojant linijos primityvą braižomas grafikas.
6. Kuo daugiau duomenų gauname tuo tikslesnis grafikas. 31 pav., 32 pav.



31 pav. Duomenų vizualizavimas naudojant grafiką intervale $[-10,10]$ ir atsitiktinius skaičius generuojanti funkcija

Šaltinis: sudaryta darbo autoriaus



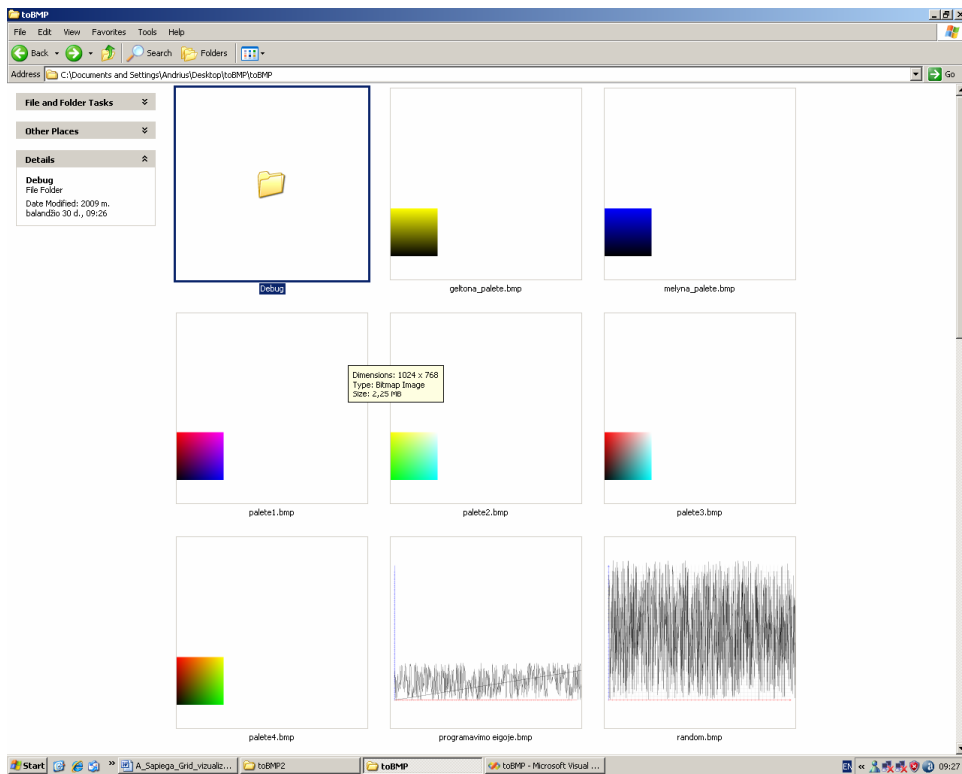
32 pav. Funkcijos $x = y$ kubu, kai $y [1;1000]$ reikšmių grafikas

Šaltinis: sudaryta darbo autoriaus

Aukščiau esančiame 32 pav. pateikiama funkcijos $x = y^3$ grafikas. Tai yra algoritmo dalyje sukamas ciklas kai reikšmės nuo $[1;1000]$ ir kiekviena gauta reikšmė saugoma

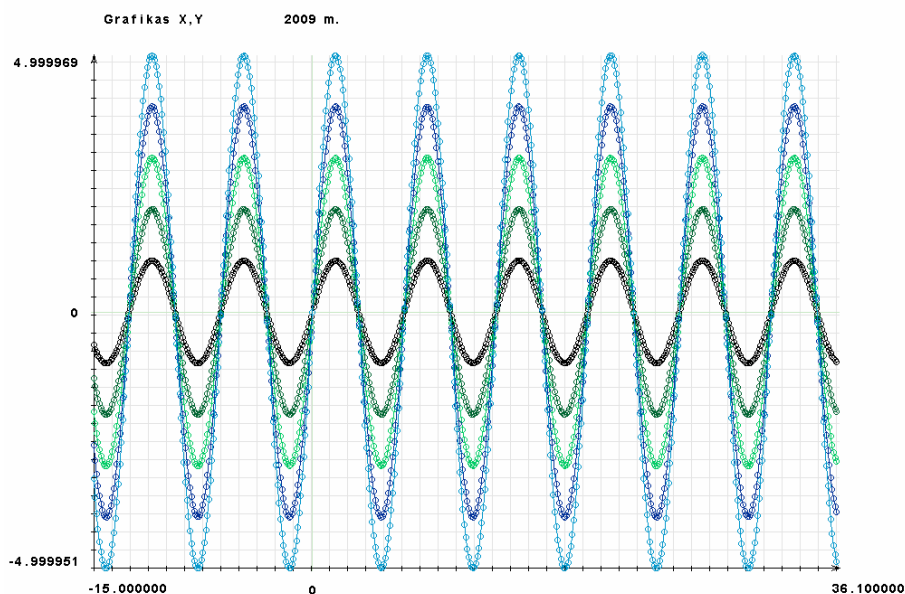
vienmačiame duomenų masyve. Šis masyvas perduodamas grafiko braižymo funkcijai ir pagal šiuos duomenis nubraižomas grafikas.

Žemiau esančiame, paveikslėlyje 33 pav. pavaizduotos gautų vizualizavimo duomenų piktogramos Windows XP operacinėje sistemoje, matomos spalvų paletes, bei atsitiktinių duomenų grafikai.



33 pav. Spalvų gamos ir atsitiktinių duomenų grafikai

Šaltinis: sudaryta darbo autoriaus



34 pav. Funkcijos $(u \cdot u) + \sin(x)$ funkcijos grafikas

Šaltinis: sudaryta darbo autoriaus

Viršuje esančiame paveikslėlyje pavaizduotas galutinis grafiko metodas 34 pav. Šis metodas papildė intervalais, tekstinių užrašų galimybe, bei galimybė įdėti keletą brėžiamu linijų. Kaip matyti iš paveikslėlio nubrėžti 5 sinuso grafikai, jiems šiuo automatiškai priskiriama spalva.

Duomenų siuntimas į *Grid* tinklą vyksta naudojant specialias komandas, jų rašymas ir kartojimas atima nemažai laiko. Tam, kad šis procesas vyktų greičiau yra naudojami „shell skriptai“. Vienas tokių skriptų buvo sukurtas šio darbo praktinėje dalyje, jis palengvina užduoties siuntimą, bei būsenos stebėjimą. Išskirtinė sukurti skripto savybė, baigus darbą, būsena atlikta *angl. done*, pasigirsta garsinis signalas, tai labai patogiu, nes nereikia vizualiai stebėti programos vykdymo *Grid* telkinyje. Šį skriptą ir jo naudojimo aprašymą rasite prie programos išėjties tekstų kompaktiniame diske.

Praktinio darbo rezultatai:

- Gautos prieigos prie *Grid* tinklu, gauti sertifikatai ir t.t.
- Įdiegta Ubuntu 5.01 operacinėje sistema su UIPnP paketu.
- Sukurtas pusiau automatinio užduoties siuntimo ir stebėjimo „shell“ skriptas.
- Sukurta priemonė duomenų vizualizavimui mokymo procese

PATARIMAI, PASTEBĖJIMAI, REKOMENDACIJOS

1. MPI bibliotekos programos vykdymo katalogo kelio problemos sprendimas

Ankstesniame 4.5. poskyryje buvo aprašytos darbe iškilusios problemos. Viena iš jų buvo programos vykdymo katalogo kelio problema.

Šiai problemai išspręsti reikėjo, įvykdyti keletą komandų. *Grid* klasteriui siunčiama programa, komandų vykdymo failas ir užduoties reikalavimų aprašas arba anlg. job description language.

Darbui atlikti naudojame tam tikrą programinę įrangą, tai gali būti openMPI, MPI2, MATH_R paketai ir bibliotekos. Tam kad vykdoma programa rastų reikiamą programinę įrangą klasteryje saugomi šių katalogų keliai, tačiau jie ne visada pateikiami automatiškai, dėl klasterio programinės įrangos ar serverio administratorių klaidų. Todėl tenka paleisti klasteryje esantį skriptą, atsakantį už tų kelių programai pateikimą.

Komandų vykdymo faile *.sh, o tiksliau pačioje šio failo pradžioje, įterpiama tokia komanda:

```
"$VO_BALTICGRID_SW_DIR/E-PARALLEL-OPENMPI-1.2.3-GCC-4.2.0/env.sh"
```

Šia komanda paleidžiamas skriptas, kuris suteikia tam tikros programinės įrangos kelius kompiliuojamai ir vykdomai programai.

Pirmoji komandos dalis „VO_BALTICGRID_SW_DIR/E-PARALLEL-OPENMPI-1.2.3-GCC-4.2.0“ nusako kokia programinę įrangą naudojame. Šio darbo atveju buvo naudojama OPENMPI-1.2.3 lygiagretaus programavimo biblioteka ir GCC-4.2.0 kompiliatorius. Antroji komandos dalis nurodo skripto pavadinimą, o prieš visą komandą esantis taškelis reiškia šios komandos paleidimą. Programinės įrangos sąrašą klasteryje galime peržiūrėti anksčiau darbe minėtoje balticgrid infosite klasterių informacijos svetainėje.[1]

Nusiuntus pataisytą vykdymo skriptą ir programą į klasterį yra paleidžiamas skriptas, nusakomi naudojamos programinės įrangos keliai ir programa įvykdoma be klaidų.

Šiame darbe naudojamas vykdymo skriptas darbo autoriaus buvo patobulintas. Įkeldami anksčiau aprašytą komandą padarome prielaidą, kad *Grid* klasteris šią programinę įrangą turi, tačiau taip yra ne visada, todėl tikslinga tai tikrinti. Tam naudojama sąlygos komanda:

```
if [ -f "$VO_BALTICGRID_SW_DIR/E-PARALLEL-OPENMPI-1.2.3-GCC-4.2.0/env.sh" ]; then  
    . "$VO_BALTICGRID_SW_DIR/E-PARALLEL-OPENMPI-1.2.3-GCC-4.2.0/env.sh"  
fi
```

Šio skripto pagalba patikriname ar programinės įrangos skriptas egzistuoja jei egzistuoja tuomet jį įvykdome.

IŠVADOS

Išanalizavus mokslinę literatūrą galime teigti, kad *Grid* technologija yra nutolę vienas nuo kito *Grid* telkiniai ir skirti sudėtingiems ir dideliems uždaviniams spręsti. *Grid* technologijos skaičiavimų pajėgumai yra prilyginami superkompiuteriams ir taikomi įvairiose mokslo srityse: informatikoje, fizikoje, chemijoje, matematikoje, medicinoje ir kt. *Grid* technologija yra perspektyvi technologija, kurios panaudojimo sritys dar vis plečiasi, o viena iš jų duomenų vizualizavimas.

Išanalizavus *Grid* technologijos taikymo prielaidas galima teigti, kad dažniausiai *Grid* technologija naudojama mokslo įstaigose ir verslo organizacijose. *Grid* technologija mokslo institucijose dažniausiai naudojama atliekant įvairius mokslinius tyrimus. Šios pajėgos telkiasi ties moksliniais išradimais grafikos ir kompiuterių lygiagretaus skaičiavimo mokymu, tiria jų tarpusavio ryšius.

Ištirtos *Grid* technologijos naudojimo mokymo procese prielaidos leido darbo autoriui išskirti keturis *Grid* technologijos naudojimo mokymo procese metodus: virtualios observatorijos, laboratorinių darbų, praktinių darbų, mokomųjų tinklų metodas. Šio darbo praktinės dalies įgyvendinimui buvo sukurta priemonė, kuri gali būti naudojama laboratorinių ir praktinių darbų atlikimui.

Egzistuoja trys prieigos prie *Grid* technologijos būdai: *Grid* telkinio grafinės aplinkos prieiga, prieiga iš vartotojo kompiuterio, prieiga naudojant specializuotą programinę įrangą pavyzdžiui Migration Desktop. Nustatyta, kad praktiniam darbui atlikti tinkamiausios yra *Grid* telkinio grafinės aplinkos prieiga ir prieiga iš vartotojo kompiuterio.

Magistro darbo praktinėje dalyje buvo įdiegta ir aprašyta programinė įranga prieigai iš vartotojo kompiuterio, suprojektuota ir įgyvendinta priemonė duomenų vizualizavimui *Grid* telkinyje bei pateikti praktiniai duomenų vizualizavimo rezultatai.

LITERATŪRA

- [1] Lietuvos grid tinklas – Litgrid. Prieiga per Internetą:
< <http://www.litgrid.lt/> > [žiūrėta 2008-10-01].
- [2] Ieva Tamutytė (2008). *Ministras Algirdas Monkevičius su delegacija dalyvavo didžiojo hadronų greitintuvo atidarymo ceremonijoje*. Švietimo ir mokslo ministerija. Prieiga per internetą: < <http://www.smm.lt/naujienos/pranesimai.htm?id=2192> > [žiūrėta 2009-03-25].
- [3] Usama M. Fayyad, Georges G. Grinstein, Andreas Wierse. *Information visualization in data mining and knowledge discovery*.
- [4] UIPnP instaliavimo, bei naudojimo instrukcija. Prieiga per Internetą:
< <http://voms-web.balticgrid.org/UIPnP-BG/> > [žiūrėta 2009-01-10]
- [5] Kompiuterio reikalavimai Windows XP operacinei sistemai. Prieiga per Internetą:
< <http://support.microsoft.com/kb/314865> > [žiūrėta 2009-01-09]
- [6] Kompiuterio reikalavimai Ubuntu 5.01 operacinei sistemai. Prieiga per Internetą:
< <http://www.ubuntu.com/products/WhatIsUbuntu/desktopedition> > [žiūrėta 2009-02-02]
- [7] Kompiuterio reikalavimai Visual Studio 2008 kompiliatoriui. Prieiga per Internetą:
<<http://download.microsoft.com/download/9/a/e/9ae0f6cc-7032-408e-9ca7-989f9e4af4ec/VS2008Readme.htm> > [žiūrėta 2008-11-15]
- [8] Programos ArgoUML šaltinis Prieiga per Internetą:
< <http://argouml.tigris.org> > [žiūrėta 2009-02-10]
- [9] Programos tightVNC šaltinis Prieiga per Internetą:
< <http://www.tightvnc.com> > [žiūrėta 2009-03-25]
- [10] Programos winSCP šaltinis Prieiga per Internetą:
< <http://winscp.net/eng/index.php> > [žiūrėta 2009-04-05]
- [11] Carl Kesselman. *The Grid, Grid Services and the Semantic Web: Technologies and Opportunities*. Information Sciences Institute. University of Southern California. Prieiga per Internetą: < <http://cseminar.web.cern.ch/cseminar/2002/2002-OtherFormats/t-20021211.ppt> > [žiūrėta 2009-03-31]
- [12] Geoffrey Fox. *Remarks on Education and the Grid*. Indiana University Bloomington. Prieiga per Internetą: < <http://grids.ucs.indiana.edu/ptliupages/presentations/edgridoct03.ppt> > [žiūrėta 2009-04-27]
- [13] Loretta Moore, Cornelius Toole. *Brokering Grid Services for Distributed Visualization*. Prieiga per Internetą:
<http://www.compframe.org/compframe2005/CompFrame05_Program/files/06032158.pdf > [žiūrėta 2009-03-31]
- [14] NEC corporation. National Institute of informatics. Naregi, Grid Visualization system Prieiga per Internetą: < http://www.naregi.org/papers/data/05_sc2005_10.pdf > [žiūrėta 2009-04-27]
- [15] Telkinio programinė įranga ir statistika. Prieiga per Internetą:
< <http://infosite.balticgrid.org/infosite/?mod=ceinfo> > [žiūrėta 2009-04-26]

- [20] Chakrabarti Grid computing security
- [21] Luis Ferreira,Victors Berstis, Jonathan Armstrong. Introduction to Grid Computing with Globus
- [22] Ian Foster Carl Kesselman Steven Tuecke. The anatomy of the grid
- [23] April J. Wells. Grid Application Systems Design
- [24] L. Jovaiša „Enciklopedinis Edukologijos Žodynas“
- [25] U. Fayyad, G. G. Grinstein, A. Wierse. Information Visualization in Data Mining and Knowledge Discovery. Prieiga per Internetą:
<<http://books.google.com/books?id=rYFvnyPRwkgC&printsec=frontcover&dq=Information+Visualization+in+data+Mining+and+Knowledge&hl=lt#PPR9,M1>>
- [žiūrėta 2009-04-28]
- [26] MPI Message Passing interface Standart Version 1.3 Prieiga per Internetą:
< <http://www.mpi-forum.org/docs/mpi1-report.pdf> > [žiūrėta 2009-05-01]
- [27] R. Čiegis (2001). *Lygiagretieji algoritmai*. Vilnius: Technika
- [28] G. Dzemyda, O. Kurasova, J. Žilinskas (2008). *Daugiamatčių duomenų vizualizavimo metodai*. Matematikos ir informatikos institutas. Vilnius: Mokslo aidai.
- [29] Prieiga per Internetą: < <https://computing.llnl.gov/?set=training&page=index> > [žiūrėta 2009-04-02]
- [30] Prieiga per Internetą: < http://ca.balticgrid.org/root_cert.html > [žiūrėta 2009-04-03]
- [31] Prieiga per Internetą: < https://hpcv.projects.sara.nl/wiki/index.php/Using_the_Grid > [žiūrėta 2009-04-05]
- [32] Prieiga per Internetą: < <http://www.porthjuniors.ndo.co.uk/Linkspages/othergfl.htm> > [žiūrėta 2009-04-04]
- [33] INT.EU.GRID organizacija.
Prieiga per Internetą: < <http://www.i2g.eu/> > [žiūrėta 2009-04-23]
- [33] VIZNET organizacija.
Prieiga per Internetą: < <http://www.viznet.ac.uk> > [žiūrėta 2009-04-28]
- [34] BERKELEY LAB organizacija.
Prieiga per Internetą: < <http://www-vis.lbl.gov/Presentations/> > [žiūrėta 2009-04-26]
- [35] INSTITUTE OF GRAPHICS AND PARALLEL PROCESSING.
Prieiga per Internetą: < <http://www.gup.jku.at/> > [žiūrėta 2009-04-27]
- [36] INSTITUTE FOR BIOCOMPUTATION AND PHYSICS OF COMPLEX SYSTEMS
Prieiga per Internetą: < <http://bifi.unizar.es/> > [žiūrėta 2009-04-22]
- [37] CROSSGRID.
Prieiga per Internetą: < <http://www.eu-crossgrid.org/>> [žiūrėta 2009-03-16]
- [37] Advanced Linux Programming
- [39] Interaktyvaus vizualizavimo naudojant „java“ pavyzdys.
Prieiga per Internetą: < <http://prefuse.org/gallery/zipdecode/> > [žiūrėta 2009-02-06]
- [40] Migrating Desktop programa.

- Prieiga per Internetą: <<http://desktop.psn.pl/>> [žiūrėta 2008-12-25]
- [41] Joshy Joseph (2004). *Grid Computing* IBM Press.
- [42] P. Shread. Wachovia Gets Boost From Grid.
Prieiga per Internetą: <<http://www.gridcomputingplanet.com/features/article.php/1553081>>
[žiūrėta 2009-04-10]
- [43] R. Šarikova (2004). *Daugiamatčių duomenų vizualizavimo metodų tyrimas*.
- [44] D. Čeikauskaitė (2007) *Daugiamatčių duomenų vizualizavimo metodų internetinės realizacijos ir jų tyrimas*.
- [45] S. K. Card, J. D. Mackinley, B. Shneiderman. Readings in Information Visualization.
Prieiga per Internetą:
<<http://books.google.com/books?id=wdh2gqWfQmgC&printsec=frontcover&dq=Readings+in+Information+Visualization&hl=lt#PPA242,M1>> [žiūrėta 2009-05-11]
- [46] Vizualizavimas graph 3d 2d ir t.t. Prieiga per Internetą:
<<http://books.google.com/books?id=olQ8PXKq-d8C&pg=PA17&dq=static+visualization+book&lr=&hl=lt#PPR10,M1>> [žiūrėta 2009-05-01]
- [47] Stephan Diehl. Software Visualization. 2007. Prieiga per Internetą:
<<http://books.google.com/books?id=8OADPxGJOa8C&pg=PA4&dq=static+visualization+book&lr=&hl=lt#PPA3,M1>> [žiūrėta 2009-04-20]
- [48] Jim Foley. Software visualization. 1998. Prieiga per Internetą:
<<http://books.google.com/books?id=13PaPtQqtHUC&pg=PA4&dq=static+data+visualization&lr=&hl=lt#PPP13,M1>> [žiūrėta 2009-03-10]
- [49] Gvid projektas.
Prieiga per Internetą: <<http://www.gup.jku.at/research/projects/gvid>> [žiūrėta 2009-01-22]
- [50] ParaView programa.
Prieiga per Internetą: <<http://www.paraview.org/>> [žiūrėta 2009-04-12]

SANTRAUKA

Grid technologija mokymo procese: duomenų vizualizavimas

Magistro darbe išanalizuotos *Grid* technologijos panaudojimo mokymo procese perspektyvos. Šiame darbe aprašytos *Grid* tinklų, *Grid* technologijos sąvokos bei sandara. Remiantis įvairiais moksliniais šaltiniais išanalizuotos *Grid* technologijos naudojimo mokymo procese perspektyvos ir mokslo sritys. Aprašyti duomenų vizualizavimo naudojant *Grid* tinklus būdai ir išskirti duomenų vizualizavimo mokymo procese metodai. Darbo praktinėje dalyje buvo išnagrinėti ir praktiškai išbandyti priegos prie *Grid* telkinio būdai ir sukurta priemonė statiniam duomenų vizualizavimui naudojant *Grid* technologiją mokymo procese atlikti.

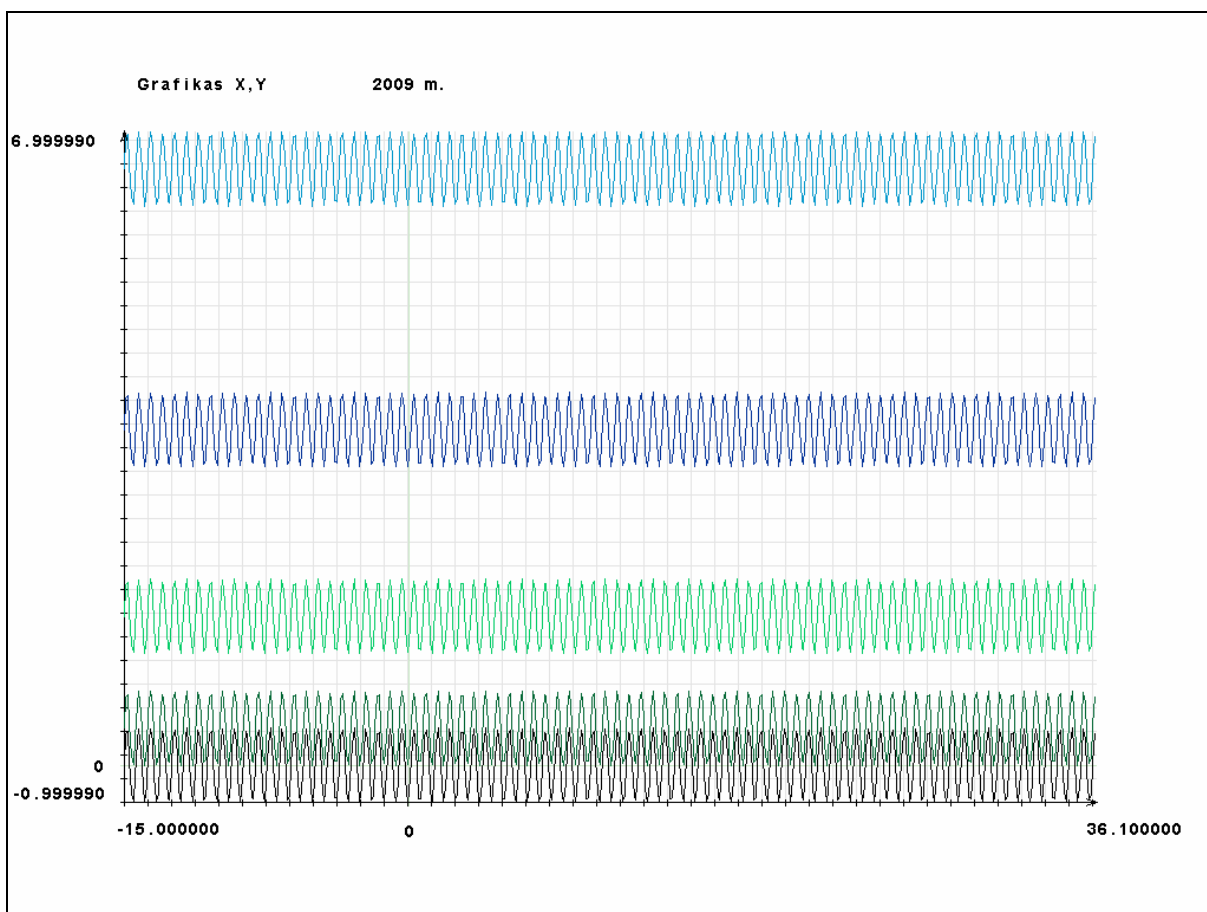
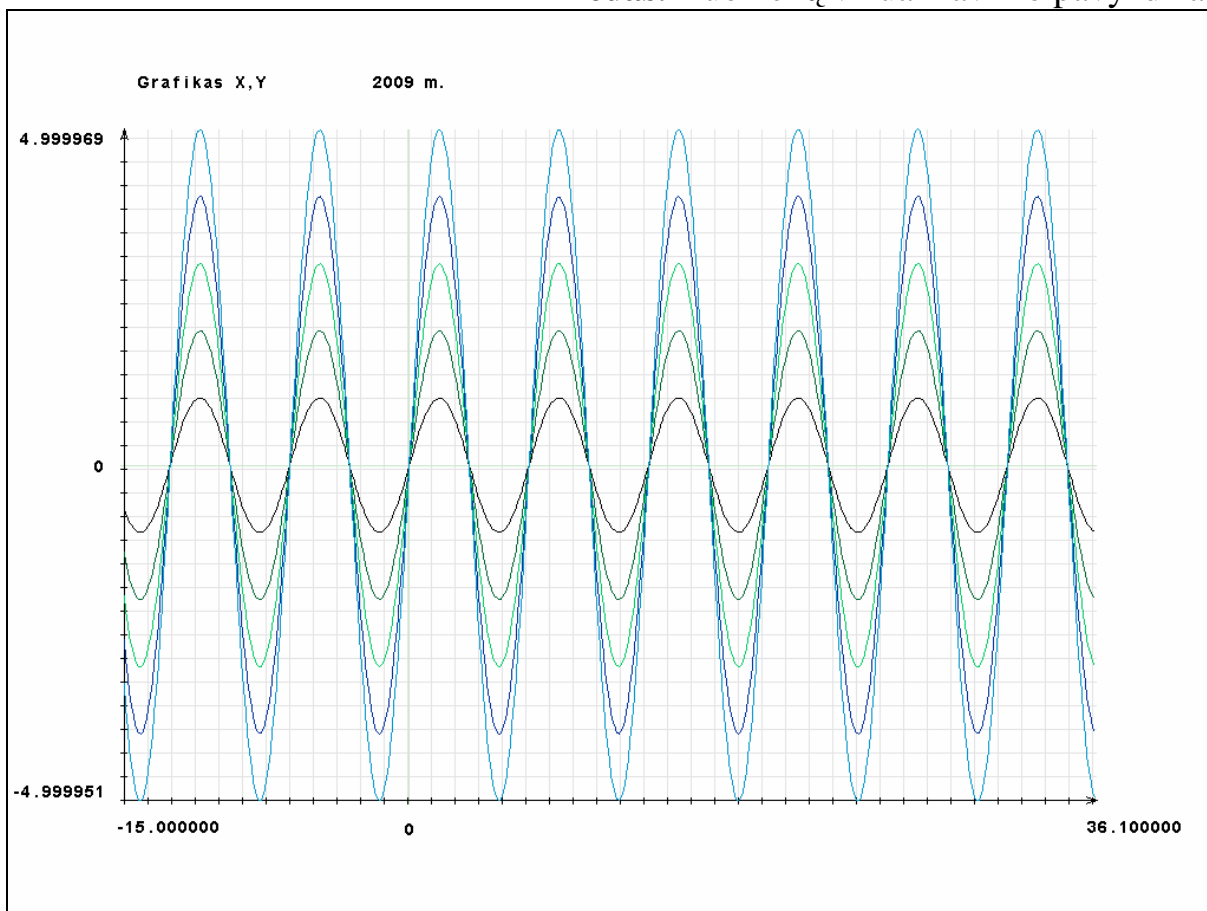
SUMMARY

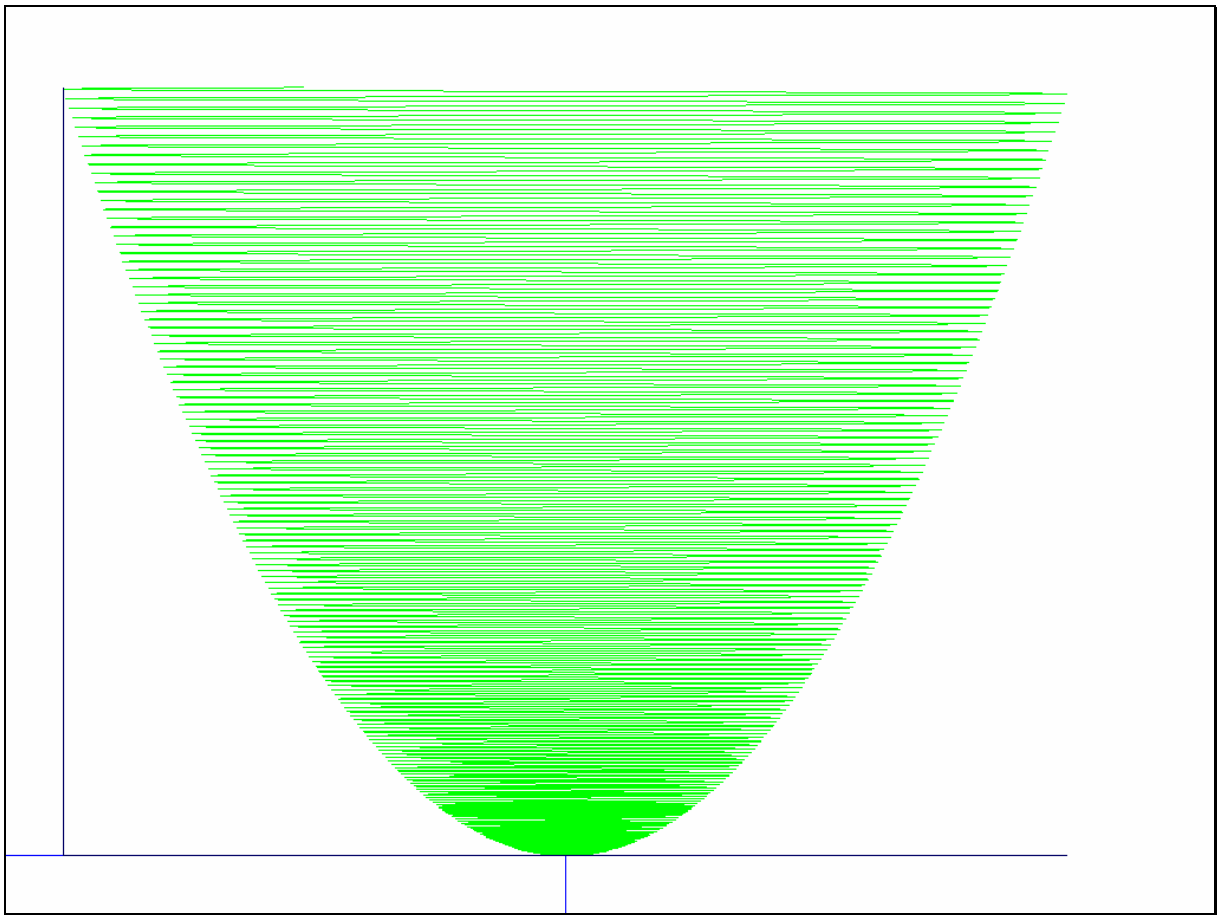
Grid technology for the educational process: data visualization

Master's work was analyzed the use of *Grid* technologies in education process perspective. This work describes *Grid* networks and the concept of *Grid* technology. According to various scientific sources there was reviewed the use of *Grid* technologies in education process and the prospects of science fields. Described data visualization techniques using *Grid* networks in data visualization and were separated data visualization methods of education process. Working in practice have been explored and tested in practice, access to the Grid, created and designed an application for static data visualization using *Grid* technology in the education process.

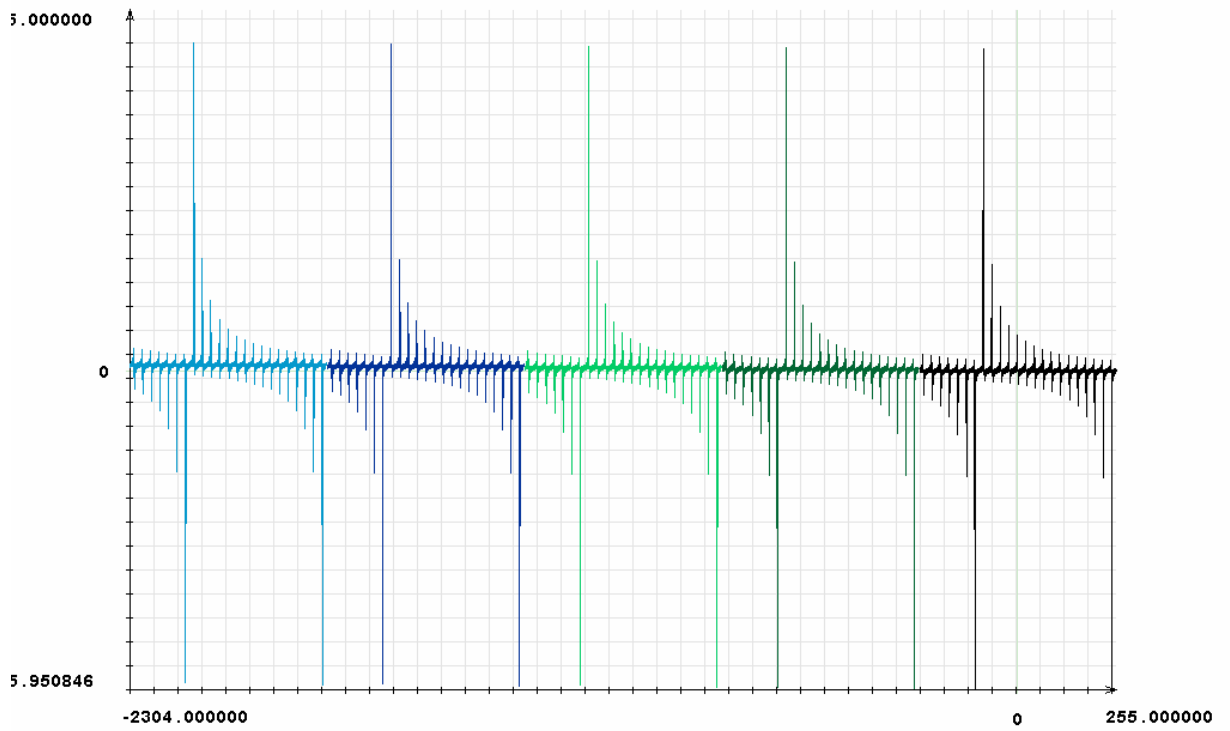
PRIEDAI

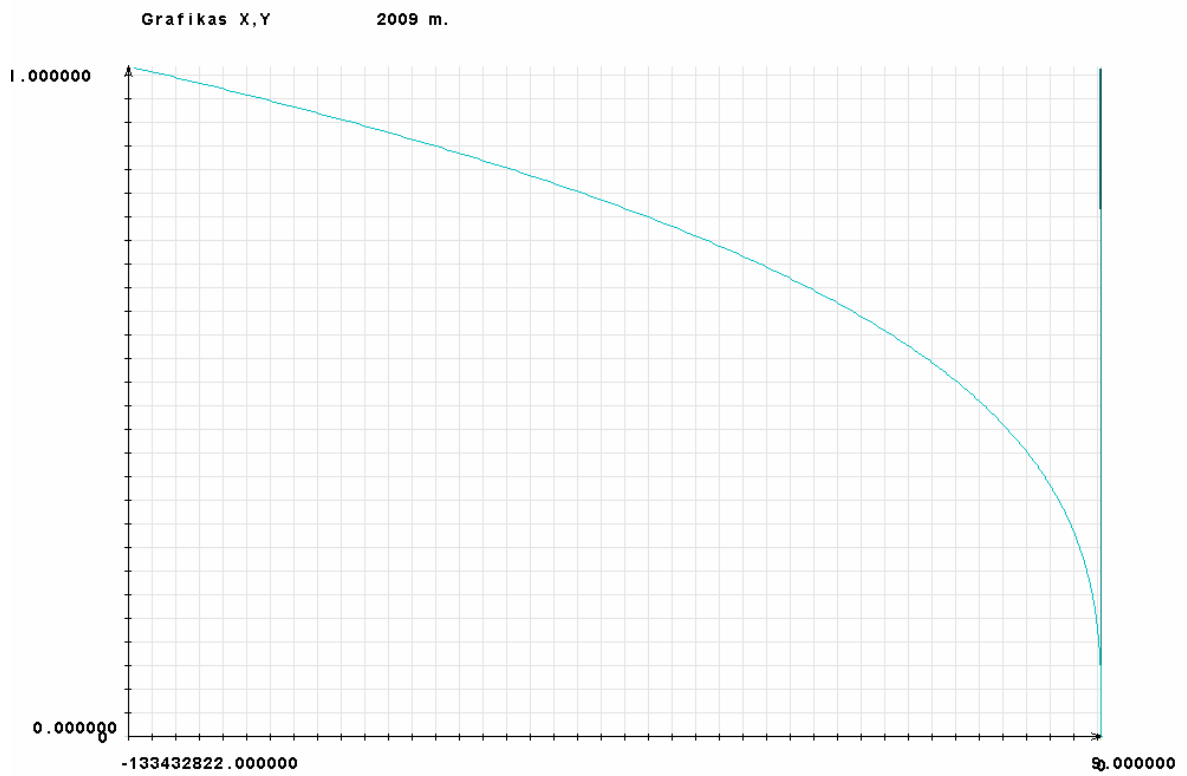
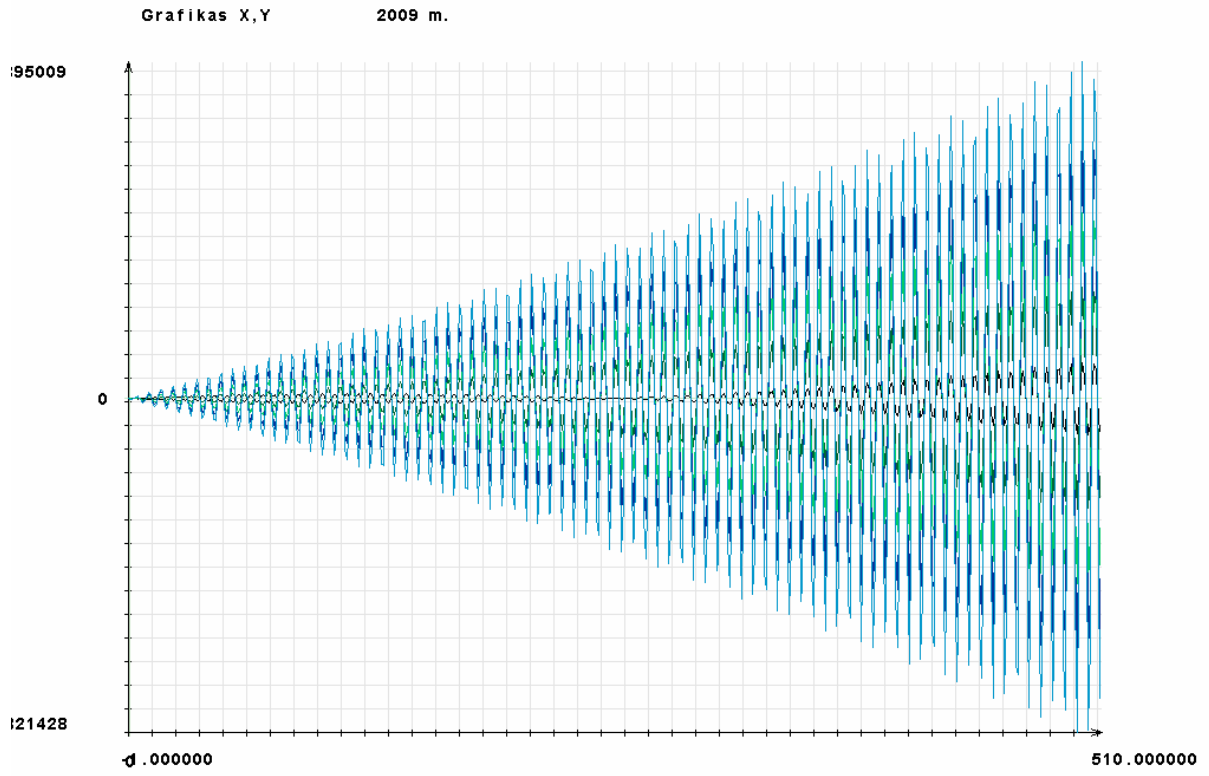
1 Priedas. Duomenų vizualizavimo pavyzdžiai





Grafikas X,Y 2009 m.





2 Priedas. Pagrindinių klasių metodų paaiškinimai.

Klasė Engine2D

1. Taško piešimas lapui p. X, Y koordinatės. R,G,B spalvos angl Red Green Blue, Alfa permatomumas. Ši funkcija leidžia generuoti permatomą tašką.

```
bool setPixel(page *p,long x,long y,char R,char G,char B,float alpha = 1);
```

2. Taškas su alfa parametru ir spalvos rgb struktūra.

```
bool setPixel(page *p,long x,long y,rgb *Color,float alpha);
```

3. Taško funkcija, skiriasi spalvos paėmimu. Per parametrus duodama spalvos struktūra.

```
bool setPixel(page *p,long x,long y,rgb *color);
```

Taško piešimas trimatėje perspektyvoje.

```
4. bool setPixel3d(page *p,float x, float y, float z,char Blue,char Green,char Red);
```

Linija trimatei perspektyvos erdvei.

```
5. bool Line3d(page *p,float x, float y, float z,float x1, float y1, float z1,char Blue,char Green,char Red);
```

Gaunama taško koordinatėje X,Y spalva color. Spalva imama per parametraž.

```
6. bool getPixel(page *p,long x,long y,rgb *color);
```

Linijos piešimas.

```
7. bool Line(page *p,long x0, long y0, long x1, long y1,char R,char G,char B);
```

Linijos piešimas su rgb struktūros spalva.

```
8. bool Line(page *p,long x0, long y0, long x1, long y1,rgb Color);
```

9. Stačiakampio piešimas.

```
bool Rectangle(page *p,long x,long y,long x1,long y1,char R,char G,char B);
```

10. Apskritimo piešimas.

```
bool Circle(page *p,long x0, long y0, long radius,char R,char G,char B);
```

11. Elipsės piešimas.

```
bool Ellipse(page *p,long xc, long yc, long a, long b,char R,char G,char B);
```

12. Nepilno apskritimo piešimas.

```
bool Arch(page *p,float xc,float yc, float Radius,float Anglestart, float AngleFinish,float Increment);
```

13. Spalvu char tipo vertimas į struktūra funkcijos pagalba. Gražinama rgb tipo struktūra.

```
rgb *rrgb(char R,char G,char B); // kovertavimas is CHAR i *rgb
```

14. Teksto šriftų užkrovimas į masyvus, pasiruošimas darbui.

```
bool loadFont(char *filename,page *p);
```

15. Nupiešiama simbolis p puslapyje x, y koordinatėse.

```
bool setLetter(page *p,long x,long y,char symbol,rgb *color);
```

16. Piešiamas „string“ formato tekstas.

```
bool setText(page *p,long x,long y,string text);
```

Klasė Graph

Pridedama viena x, y duomenų dalis arba viena grafiko linija.

```
bool Add2D(double *vectorx,double *vectory,long numx,long numy);
```

Pridedama duomenų dalis pagal x

```
bool Add1Dx(double *vectorx);
```

Pridedama duomenų dalis pagal y

```
bool Add1Dy(double *vectory);
```

Gaunamas galutinis rezultatas pagrindinis grafiko piešimo metodas

```
bool DrawLines(page *p);
```

Konstruktorius

```
Graph();
```

Destruktorius

```
~Graph();
```

Klasė Bmp_file

Puslapio leafe įrašymas į windows bitmap formato failą

```
short write_image(char *filename,bmp_data *header,page *leafe )
```

Puslapio nuskaitymas iš failo.

```
short read_image(page *p){};
```

Paveikslo dydžio keitimas.

```
short resize(){};
```