

TARTU ÜLIKOOL  
MATEMAATIKA-INFORMAATIKATEADUSKOND

Arvutiteaduse instituut

Karl Kruusamäe

**ARVUTIKLASTRITE LOOMISE JA HALDAMISE  
TARKVARAD**

Bakalaureusetöö (4 AP)

Juhendaja: Prof. Eero Vainikko

Tartu 2006



# SISUKORD

<b>SISUKORD</b> .....	<b>3</b>
<b>SISSEJUHATUS</b> .....	<b>5</b>
<b>PEATÜKK 1. MITME PROTSESSORIGA SÜSTEEMID</b> .....	<b>7</b>
1.1. TENTALISEERITUD MULTIPROTSESSORSÜSTEEMID .....	8
UMA.....	8
NUMA .....	8
1.2. MITME ARVUTIGA SÜSTEEMID ehk ARVUTIKLASTRID .....	10
ARVUTIKLASTRITE JAOTUS STRUKTUURI JÄRGI .....	10
SÜMMEETRILINE KLASTER .....	10
ASÜMMEETRILINE KLASTER.....	11
LAIENDATUD KLASTER.....	12
ARVUTIKLASTRITE JAOTUS FUNKTSIONAALSUSE JÄRGI .....	12
HÕLPSA JUURDEPÄÄSU KLASTRID (HA klastrid) .....	13
KOORMUSE TASAKAALUSTAMISE KLASTRID (LB klastrid) .....	13
ARVUTUSKLASTRID (HPC klastrid) .....	13
VÕREKLASTRID .....	14
<b>PEATÜKK 2. KLASTRITE TARKVARA</b> .....	<b>15</b>
2.1. openMOSIX .....	16
ÜLEVAADE .....	16
SEADISTAMINE JA HALDAMINE .....	17
RIIST- JA TARKVARALISED NÕUDED .....	17
KLASTRI SEADISTAMINE .....	17
TARKVARA PAIGALDAMINE JA SEADISTAMINE .....	18
KLASTRI HALDAMINE .....	19
2.2. ROCKS DISTRIBUTSIOON .....	20
ÜLEVAADE .....	20
SEADISTAMINE JA HALDAMINE .....	20

RIIST- JA TARKVARALISED NÕUDED .....	20
KLASTRI SEADISTAMINE .....	21
TARKVARA PAIGALDAMINE JA SEADISTAMINE .....	21
KLASTRI HALDAMINE .....	22
2.3. OSCAR .....	23
ÜLEVAADE .....	23
SEADISTAMINE JA HALDAMINE .....	24
RIIST- JA TARKVARALISED NÕUDED .....	24
KLASTRI SEADISTAMINE .....	25
TARKVARA PAIGALDAMINE JA SEADISTAMINE .....	25
KLASTRI HALDAMINE .....	26
2.4. ClusterKNOPPIX .....	27
ÜLEVAADE .....	27
MIS ON KNOPPIX, CLUSTERKNOPPIX? .....	27
SEADISTAMINE JA HALDAMINE .....	27
2.5. BCCD.....	28
ÜLEVAADE .....	28
SEADISTAMINE JA HALDAMINE .....	28
<b>KOKKUVÕTE .....</b>	<b>29</b>
<b>VIITED.....</b>	<b>30</b>
<b>ABSTRACT.....</b>	<b>31</b>

## SISSEJUHATUS

Tänapäeva maailmas on kasvav vajadus stabiilsete ja suure jõudlusega süsteemide järele. Iga ettevõtte tahab pakkuda oma klientidele kiiret ja alati saadaval olevat teenust. Teiseks on vajalik suuremahuliste analüüside tegemine, et prognoosida võimalikult täpselt kõikvõimalikke tuleviku arenguid. Kõike seda soovitakse saavutada nii madalate hindade juures kui vähegi võimalik. See kirjeldus katab ainult ühe valdkonna, kus vajatakse arvutiklastreid. Neid saab kasutada igal pool ning nõudlus aina kasvab.

Arvutiklastrite loomine on olnud alati keeruline protsess nii asutusele kui ka otseselt töid teostavale süsteemihaldurile. Õnneks on viimaste aastate jooksul toimunud märkimisväärne areng kobardamise (arvutiklastrite loomise) protsessi lihtsustumiseks. Arendatud on mitmeid tarkvarakomplekte, mis peaksid muutma arvutiklastrite loomise ja haldamise lihtsamaks.

Hoolimata tohutust arengust antud valdkonnas, on siiski enamikel arvutiteadusega seotud inimestel ettekujutus, et kobararvutite loomine on midagi keerulist ja segast. Käesoleva töö eesmärgiks ongi tutvustada kõigile lihtsalt kättesaadavaid vahendeid, mis aitavad võrdlemisi kergelt luua funktsionaalseid arvutusklastreid.

Käesolev töö sisaldab kahte põhipeatükki. Esimeses peatükis tutvustatakse mitme protsessoriga arvutisüsteeme. Antud peatükk omakorda jaguneb kaheks alapeatükiks, millest esimene (peatükk 1.1) käsitleb tsentraliseeritud multiprotsessorsüsteeme. Allikana on kasutatud J. D. Sloani raamatut *High Performance Linux Clusters with OSCAR, Rocks, OpenMosix, and MPI* [1].

Peatükis 1.2 tutvustatakse lähemalt mitme arvutiga süsteeme ehk arvutiklastreid. Aluseks on võetud *Wikipedia, the free encyclopedia* [2] teemakohased kirjed ning samuti ka eelnevalt mainitud raamat [1].

Bakalaureusetöö teine peatükk tegeleb aga juba täpsemalt kobardamiseks vajamineva tarkvaraga. Terve teise peatüki ulatuses on paiguti tuginetud J. D. Sloani sellekohasele raamatule [1]. Peatükis 2.1 tutvustatakse openMosix kobararvuti tarkvara. Lähtutud on openMosixi ametlikust dokumentatsioonist [3] ja Kris Buytaerti artiklist [4], mis käsitleb antud tarkvara.

Peatükk 2.2 tegeleb Rocks distributsiooni tarkvarapakettide ülevaatega. Baseerub ta suures osas Rocks distributsiooni dokumentatsioonil [5]. Peatükis 2.3 saavad kirjeldatud OSCAR tarkvara erinevused eelnevalt nimetatud toodetest. Allikana on kasutatud OSCAR tarkvarakomplekti ametlikku paigaldusjuhendit [6].

Peatükid 2.4 ja 2.5 käsitlevad nn kiirkobardamistarkvara; kasutatud on nii KNOPPIXi [7], ClusterKNOPPIXi [8] kui ka BCCD [9] ametlikke võrgulehekülgi.

Tarkvarakomplekte tutvustades on suurem rõhk pandud idee kirjeldamisele. Töö eesmärgiks ei ole anda põhjalik kasutusjuhend käsiteldavatele tarkvaradele, vaid tutvustada põgusalt lihtsamaid ja populaarsemaid võimalusi kobararvuti loomiseks.

Iga tarkvarapaketi juures on toodud välja üldine kirjeldus, minimaalsed süsteemi nõuded, lihtne ülevaade paigaldusprotsessist ning haldamisvahendite olemasolu. Kindlasti võiks käesolev töö olla kasuks isikule, kellel on plaanis lähitulevikus ise arvutusklastrer valmis ehitada. Üheks käesoleva töö eesmärgiks süstida julgust neisse isikutesse, kes sooviksid ise kobardamisega tegeleda, aga siiani on arvanud, et tegu on keerulise võrgutehnoloogilise ülesandega.

Kõiki käesolevas töös kajastatud tarkvarakomplekte ühendab see, et nad on mõeldud Linux arvutusklastrite jaoks, nad on avatud lähtekoodil põhinevad vabavarad ning nende paigaldamine on arendajate poolt tehtud nii lihtsaks, kui vähegi hetkeseisuga võimalik.

Käesoleva bakalaureusetöö lugejalt eeldatakse keskmisi teadmisi võrgutehnoloogiast ja süsteemihaldusest, sest nende valdkondadega seotud terminoloogia lahtiseletamisele ei ole suurt rõhku pandud.

## PEATÜKK 1. MITME PROTSESSORIGA SÜSTEEMID

Suuremad arvutuskiirused ei ole vajalikult pelgalt mugavuse pärast. Kiired arvutid lubavad meil lahendada suuremaid probleeme, leida lahendeid kiiremini ja suurema täpsusega. Traditsionaalsed arvutusklastrid on juba õigustanud oma kasutust mitmeski eluvaldkonnas – ilmaennustamine, osakeste füüsika, statistilised arvutused jne. Suure jõudlusega arvutusvõimalused on loonud uued teaduslikud lähenemisviisid. Modelleerimine on teostatav ja arvestatav alternatiiv traditsioonilistele lähenemisviisidele (eksperimentaalsele ja teoreetilisele).

Klastrid omavad kasvavat tähendust ka majanduslikel põhjustel. Suured arvutusjõudlused on osutunud vajalikuks nii andmeanalüüsis kui ka kujutiste visualiseerimises (ingl k „*image rendering*“).

Arvutusjõudluse parendamiseks on kolm põhivõimalust: kasutada kiiremat algoritmi, kasutada kiiremat arvutit või jagada arvutusülesanne mitme erineva arvuti vahel [1].

Tihti on arvutusprotsessi aegluses süüdi kohmakas lahendamise algoritm, mis sisaldab liigseid instruktsioone. Seega lahendusaja kahandamiseks on vaja lihtsalt redigeerida lahendusalgoritmi. Siiski peab silmas pidama, et ka täielikult korrektsed lahendusalgoritmid võtavad suurte andmemahtude juures palju aega.

Ühe lahendusena on võimalus osta kiirem arvuti, mis kindlasti vähendab probleemi lahendamiseks kuluvat aega. Kiiremad arvutid on aga väga kallid ning hetke tehnoloogia seab piirid arvuti kiirustele.

Siin tulebki sisse võimalus jaotada probleem mitmeks osaks ning lahendada erinevate protsessorite peal. Mitme protsessoriga süsteemid saame jaotada kaheks:

- Tsentraliseeritud multiprotsessorsüsteemid (mitu protsessorit ühe arvuti sees).
- Mitme arvutiga süsteemid ehk klastrid.

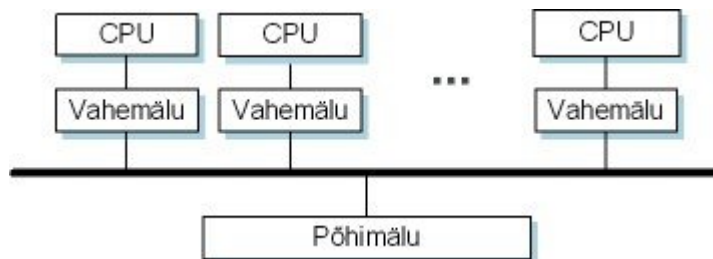
Järgnevalt tutvustatakse mõlemat tüüpi lühidalt.

## 1.1. TSENTRALISEERITUD MULTIPROTSESSORSÜSTEEMID

Tsentraliseeritud multiprotsessorisüsteemid („*centralized multiprocessors*“) jagunevad mäluhalduse tõttu arhitektuuriliselt kaheks: UMA-süsteemideks (*unified memory access*) ja NUMA-süsteemideks (*nonunified memory access*).

### UMA

UMA masinate protsessoritel on ühine jagatud mälu (joonis 1). Sellist tüüpi masinaid nimetatakse ka SMP süsteemideks ehk sümmeetrilisteks multiprotsessorisüsteemideks. Sõltumata protsessorist, kaardistavad identsed mäluaadressid samu asukohti füüsilises mälus. Põhimälu on võrdselt ligipääsetav kõigile CPU-dele. Tõhustamiseks suhtlust mäluiga on iga protsessoril oma vahemälu („*cache*“).



**Joonis 1.** UMA arhitektuur

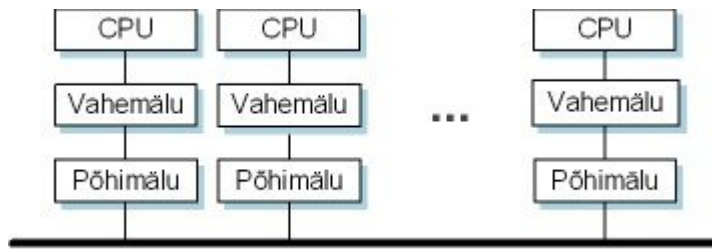
UMA arhitektuuriga käib kaasas kaks võrdlemisi suurt probleemi. Esimeseks on sünkroniseerimine – kommunikatsioon protsesside vahel ja juurdepääs välisseadmetele peab olema koordineeritud, et vältida võimalikke konflikte. Teiseks põhiprobleemiks on vahemälu kooskõla põhimäluga. Kui kaks erinevat CPU-d kasutavad sama põhimälu osa ning üks muudab sealset väärtust, siis kuidas tagada muutus ka teise protsessori vahemälus. Kõige levinum lahendusviis on nuhkimine („*snooping*“). Iga vahemälu kuulab pealt kõiki mälupealduisi ja kui muutusi tehakse aadressidel, mis sisalduvad vahemälus, siis uuendatakse andmeid vahemällu.

### NUMA

Teiseks mäluhaldamise põhjal tekkinud arhitektuuriks on NUMA. Seda tüüpi masinate protsessoritest igaüks haldab oma isiklikku mälu (joonis 2). Kogu põhimälu on jagatud erinevate CPU-de vahel, kuid kõigile protsessidele on kättesaadav kogu mälu. Sõltumata protsessorist, endiselt iga identne mäluaadress kaardistab sama asukohta füüsilises mälus. Mälu mitteühne kasutamine saavutatakse sellega, et teatud osad



mälust paistavad olevat märksa aeglasemad kui teised, sest mäluosa, mis on „lähim“ protsessorile, on kiiremini pöörduv selle CPU poolt.



**Joonis 2.** NUMA arhitektuur

Kummagi multiprotsessor süsteemi puhul on vajalik operatsioonisüsteemi poolne tugi. Enamik tänapäevaseid operatsioonisüsteeme toetab SMP süsteeme ning NUMA arhitektuuri tugi on ka püsivalt arenev.

## **1.2. MITME ARVUTIGA SÜSTEEMID ehk ARVUTIKLASTRID**

Arvutiklaster ehk kobararvuti on grupp vabalt ühendatud arvuteid, mis töötavad tihedalt koos, nii et üldises mõttes saab neid vaadelda kui ühte arvutit. Klaster on tavaliselt, aga mitte alati, ühendatud läbi kiire kohtvõrgu („LAN“). Klaster luuakse eesmärgiga saavutada suurem kiirus ja usaldusväärsus kui üksikul arvutil ning nad on tavaliselt odavamad kui samaväärse usaldusväärsuse ja kiirusega üksikarvutid.

Arvutiklastri on kolm põhilist elementi – kogum individuaalseid arvuteid; võrk, mis ühendab need arvutid; ning tarkvara, mis tänu võrgule võimaldab töökoormuse jagamist teiste arvutitega.

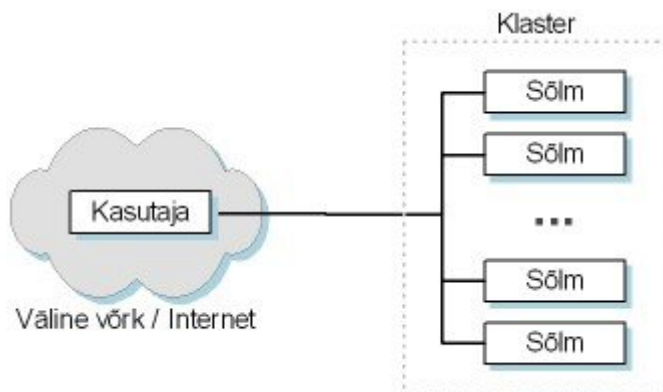
### **ARVUTIKLASTRITE JAOTUS STRUKTUURI JÄRGI**

Võttes aluseks kobararvutite struktuuri võib nad jagada kolmeks:

- Sümmeetriline klaster.
- Asümmeetriline klaster.
- Laiendatud klaster.

### **SÜMMEETRILINE KLASTER**

Lihtsaim moodus klasterit üles seada on kasutada sümmeetrilist struktuuri (joonis 3). Sellisel juhul võivad kõik sõlmed funktsioneerida ka üksikarvutitena. Sümmeetrilise klasteri loomine on otsene ja lihtne. Selleks on vaja ühendada arvutid ühtsesse alamvõrku („*subnetwork*“) ning lisada struktuurile sobiv spetsiifiline kobararvuti tarkvara. Sellist struktuuri kohtab eelkõige NOW (*Network of Workstations*) klasterite juures. NOW, vahel ka COW (*Cluster of Workstations*), on klaster, mille sõlmedeks on tööjaama arvutid, mida kasutatakse ka individuaalselt, näiteks ülikooli arvutiklass.

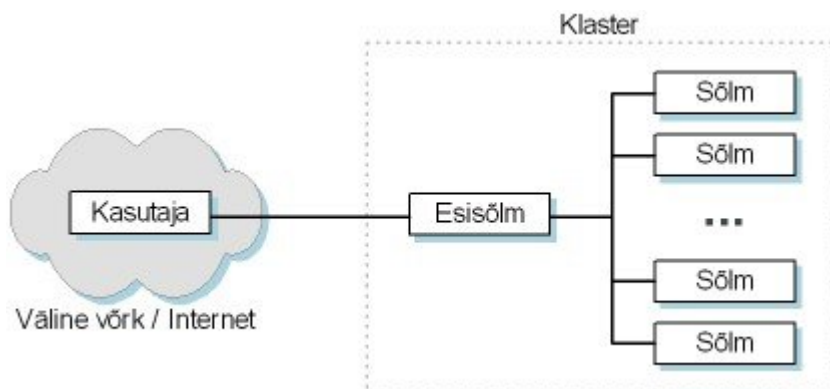


**Joonis 3.** Sümmeetrilise klasteri struktuur

Kuigi sümmeetrilised klasterid on lihtsa arhitektuuriga, omavad nad nii mitmeidki negatiivseid külgi. Esiteks on väga raske tagada turvalisust, sest selleks tuleb seadistada kõiki masinaid. Siit paistab ka teine probleem – sellise klasteri haldamine on väga keeruline. Lisaks võib tekkida probleeme töökoormuse jaotumisega klasteril, mistõttu ei saavutata optimaalset jõudlust.

### ASÜMMEETRILINE KLASTER

Spetsiifilisematel kobararvutitel kasutatakse tihedamini asümmeetrilist arhitektuuri (joonis 4). Asümmeetrilise struktuuri korral on üks arvuti esisõlmeks („*head node*“) ehk esiarvutiks („*front-end*“). See funktsioneerib lüüsina kasutajate ning ülejäänud klasteri sõlmede vahel. Ülejäänud sõlmedel on tihtipeale ainult minimaalne operatsioonisüsteem ja riistvara, nad on pühendunud klasteri teenindamisele. Kuna kogu liiklus peab läbima esiarvutit, siis asümmeetriline struktuur pakub suurt turvalisust, sest kasutajad on usaldusväärsed ning ülejäänud sõlmed on maailmast füüsiliselt eraldatud. Tugevdada tuleb ainult esiarvutit.



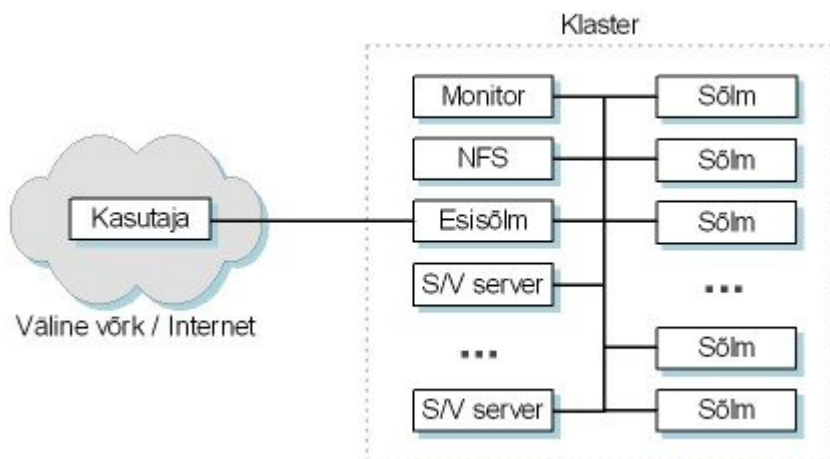
**Joonis 4.** Asümmeetrilise klasteri struktuur

Esiarvuti paistab teistele sõlmedele justkui põhiserverina. Kuna esiarvuti suhtleb nii avaliku võrgu kui ka klastriga, siis seadistatakse ta ülejäänud klastri sõlmedest erinevalt. Tavaliselt on lihtsam hoida kõik seadistused selles ühes masinas, mistõttu lihtsustub oluliselt ülejäänud sõlmede seadistamine ja haldamine.

Asümmeetriliste klastrite peamine puudus on jõudluse piirangud, mis on tingitud esiarvuti kiirusest. Seetõttu peaks esisõlmel kasutama märksa suurema jõudlusega arvutit kui ülejäänud sõlmedel.

## LAIENDATUD KLASTER

Kuigi esiarvuti jõudluse suurendamine asümmeetrilise arhitektuuri korral võib aidata väiksemaid klastreid, ei pruugi sellest piisata suuremat sorti klastri korral. Sellisel juhul tuleks kobararvutile lisada uusi servereid. Üks sõlm võiks tervenisti toimida kui klastri juhtimiskeskus, kus toimub ka süsteemi monitooring, teine aga sisend-väljund serverina („I/O server“ ingl k) jne. Sellisel juhul on tegemist laiendatud klastriga (joonis 5).



**Joonis 5.** Laiendatud klastri struktuur

Laiendatud struktuuri suurimaks puuduseks on tema teatav keerulisus. Sellise arhitektuuri rakendamiseks on vaja osavat süsteemi administraatorit.

## ARVUTIKLASTRITE JAOTUS FUNKTSIONAALSUSE JÄRGI

Funktsionaalsuse põhjal eristatakse nelja erinevat arvutiklastri tüüpi:

1. Hõlpsa juurdepääsu klastrid.
2. Koormuse tasakaalustamise klastrid.

3. Arvutusklastrid.

4. Võreklastrid.

Tihti võib kohata jaotust ainult kolmeks, sellistel juhtudel ei ole mainitud võreklastreid, sest need väljuvad tavapärase arvutiklastri mõiste raamidest.

### **HÕLPSA JUURDEPÄÄSU KLASTRID (HA klastrid)**

Hõlpsa juurdepääsu klastreid („*high-availability cluster*“) rakendatakse eelkõige teenuste kättesaadavuse parendamiseks. Kui eesmärgiks on süsteem, mis teenindab kliente 99,9 protsenti ajast, siis tuleks kasutada HA klastrit. Nende tööpõhimõte on omada tagavara sõlmesid („*node*“), mis pakuvad teenust siis, kui mõni süsteemi komponent langeb ära. Teenust pakub üks masin, teised sõlmed on kuulavas režiimis. Kui teenust pakuv masin peaks ära langema, siis võtab järgmine sõlm teenuse pakkumise rolli üle. Tavaliselt kasutatakse HA klastrit puhul kahte sõlme, mis on ühtlasi ka minimaalne võimalik hõlpsa juurdepääsu klastrite suurus. HA klastreid nimetatakse osades allikates ka tõrkesiirde („*failover*“) klastriteks.

### **KOORMUSE TASAKAALUSTAMISE KLASTRID (LB klastrid)**

Koormuse tasakaalustamise klastreid („*load balancing cluster*“) eesmärgiks on tagada suurem jõudlus jagades töö mitme erineva arvuti vahel. LB klastreid korraldab lastakse kogu töökoormus läbi ühe või mitme eesarvuti („*front-end node*“), mis omakorda jaotab koormuse põhiservertite („*back-end server*“) kogumisse. Kuigi seda tüüpi kobararvuteid luuakse jõudluse kasvatamise eesmärgiga, omavad nad tihti ka HA klastreid omadusi. Selliseid arvutiklastreid kutsutakse tihtipeale serverifarmideks.

### **ARVUTUSKLASTRID (HPC klastrid)**

Arvutusklastril („*high-performance computing cluster*“) jaotatakse arvutusülesanne kobararvuti erinevate sõlmede vahel, mille tulemusena kasvab arvutusjõudlus. Kuigi arvutusklastril toimub samuti koormuse jaotumine, tehakse kirjanduses vahet HPC ja LB kobararvutitel. LB klastreid kasutatakse veebiteenuste pakkumiseks, samas kui HPC klastreid kasutatakse üldiselt teadusarvutuste tegemiseks. Üks levinumaid HPC klastreid on kobararvuti, mille sõlmede operatsioonisüsteemiks on Linux ning paralleeliseerimist realiseerib spetsiaalne tasuta tarkvara. Sellist seadistust nimetatakse ka Beowulf klatriks või tarbeklatriks, kuna kasutatakse nn. COTS (*commodity*,

*off-the-shelf*) arvuteid. COTS arvutiteks nimetatakse tavalisi kõigile kättesaadavaid ja poes müüdavaid arvuteid.

HPC klasteri sõlmedel jooksvad tööd ja protsessid suhtlevad arvutamise ajal omavahel aktiivselt. Seega on võimalikud arvutused, kus ühe protsessi või töö vahetulemused võivad mõjutada teisel sõlmel toimuvaid arvutusi.

## **VÕREKLASTRID**

Võreklasterite („*grid cluster*“) tehnoloogia on väga lähedalt seotud arvutiklasteritega. Peamine erinevus võre- ja traditsiooniliste arvutiklasterite vahel on see, et võred ühendavad omavahel arvuteid, mis teineteist täielikult ei usalda. Seetõttu võiks võresid vaadata kui arvutusvahendit, mitte aga kui üksikarvutit. Lisaks toetavad võred märksa heterogeensemaid seadmete kogumeid, kui tavalised arvutiklasterid.

Võreklasterites jooksutatavad tööd on üksteisest sõltumatud, mistõttu ei pea nad vahetama infot arvutamise protsessi jooksul. Võre juhib tööde paigutamist arvutitesse, mis teevad tööd sõltumata ülejäänud võreklasterist. Ressursid nagu salvestuselement võivad olla küll ühised kõigile sõlmedele, aga ühe töö vahetulemused ei mõjuta teisel sõlmel toimuva töö või protsessi tulemusi.

## PEATÜKK 2. KLASTRITE TARKVARA

Juba varem sai mainitud, et kobararvutitel on eelkõige kolm põhilist komponenti – kogum individuaalseid arvuteid; võrk, mis neid arvuteid ühendab; ning tarkvara, mis kasutab võrku, et jaotada protsesse teistele klatri sõlmedele.

Tavapärase klatri ülesseadmine eeldab paljude tarkvara toodete installeerimist ja seadistamist, lisaks peab muutma, ka olemasolevat tarkvara. Laheduseks on nn kobardamise tarkvarakomplektid (Rocks, OSCAR jne). Nende kasutamise eeliseks on see, et klatri võib üles seada märkimisväärselt kiiresti ning kasutajalt ei eeldata väga suuri teadmisi hajussüsteemide valdkonnas.

Selliste tarkvarakomplektide puuduseks on klatri administraatori distantseerumine süsteemist. Kuna tarkvara teeb nii mõnegi seadistusega seotud otsuse süsteemiadministraatori eest, tekib olukord, kus haldur ei pruugi täpselt teada, kuidas antud kobararvuti või süsteem üldiselt töötab. See aga muudab süsteemi haavatavaks ning probleemi kerkimise korral ei pruugi keegi seda lahendada osata.

Üldiselt võib öelda, et järgnevalt käsitletavat tarkvarad võimaldavad kerge vaevaga luua ja hooldada väikese ja keskmise suurusega arvutiklastreid, kuid stabiilse ja suure süsteemi ehitamiseks on vajalik ka süsteemilooja spetsiifilised teadmised kobardamise valdkonnast.

## 2.1. openMOSIX

### ÜLEVAADE

OpenMosix oli algselt MOSIX (*Multicomputer Operating System for Unix*) projekti üks haru. OpenMosix projekt sai alguse kui MOSIX lahkus üldise avaliku litsentsi alt. Tänapäevaks on openMosixist saanud iseseisev projekt, mille juhiks on mees nimega Moshe Bar.

OpenMosix on tarkvarapakett, mis muudab kohtvõrgus olevad GNU/Linux operatsioonisüsteemiga arvutid arvutusklatriks. Ta automaatselt tasakaalustab koormuse erinevate klatri sõlmede vahel. Sõlmi võib lisada ja eemaldada jooksvast klatriks, tekitamata häireid süsteemi töös. Töökoormus jaotatakse sõlmede vahel vastavalt nende töö- ja võrgukiirusele. Kuna openMosix on osa Linuxi tuumast („kernel“) ja seega säilitab täieliku ühilduvuse Linuxiga, siis kasutajate programmid, failid ja muud rakendused töötavad muudatusteta edasi. Tavaline kasutaja ei näe erinevust Linuxi ja openMosix süsteemi vahel – tema jaoks funktsioneerib terve klaster kui üks kiire GNU/Linux süsteem.

OpenMosix tarkvara kätkeb endas nii Linuxi tuuma paikamisprogrammi („patch“) kui ka vajalikke vahendeid openMosix klatri juhtimiseks ja hooldamiseks. Sisemine koormust jaotav programm on nähtamatu, ta migreerib erinevad protsessid dünaamiliselt teistele klatri sõlmedele. Eeliseks on parem koormuse jaotumine sõlmede vahel. Klaster ise üritab pidevalt optimeerida tööjaotust erinevatel masinatel. Muidugi on ka süsteemi administraatoril võimalik käsitsi kontrollida automaatset töökoormuse jaotamist.

Et toetada protsesside migreerumist, jaotab OpenMosix protsessid kaheks osaks – **kasutaja** ja **süsteemi osaks**. Kasutaja osa sisaldab programmi koodi, pinu, andmeid jms. ning see osa võib migreeruda. Süsteemi osa sisaldab tuuma pinu ning kirjeldust selle kohta, milliste ressurssidega on protsess seotud. Süsteemi osa ei migreeru, vaid püsib UHN-is ehk ainulaadses kodusõlmes („*unique home node*“).

Selle nähtamatu protsesse migreeriva süsteemi tõttu, paistab kogu klaster nagu suur SMP (*sümmeetriline multiprotsessorsüsteem*) masin, millel on sama palju protsessoreid kui klatriks sõlmi. Samas on openMosix klaster tõenäoliselt odavam kui SMP arvuti.



OpenMosix pakub ka võimsat optimeeritud failisüsteemi (oMFS) HPC rakenduste jaoks, mis on sarnane NFS failisüsteemiga, aga erinevalt sellest, pakub ka vahemälu, ajatemplite ja linkimise kooskõla.

Rakenduse jooksutamiseks openMosix klastril ei ole vajalik uuesti kompileerimine või teiste teekide integreerimine. Programmid nagu Flac, Bladeenc, Povray ja mjpeg vahendid funktsioneerivad ilma kõikvõimalikke muudatusi tegemata, sama kehtib ka MPI kohta. See ongi üks põhilisi erinevusi ja eeldusi, mis on openMosixil enamiku muu klastrite tarkvaraga võrreldes.

MPI rakendused töötavad veel paremini openMosix keskkonnas. Kuigi protsess algab kodusõlmel, teeb klaster kindlaks, kas poleks parem jooksutada seda mõne muu, vähem koormatud sõlme peal. OpenMosix kasutab arenenud algoritmi, et teha kindlaks, milline sõlm on sobivaim antud rakenduse jooksutamiseks. Seega juba eelnevalt paralleelsed protsessid võivad samuti openMosix klastril.

## **SEADISTAMINE JA HALDAMINE**

### **RIIST- JA TARKVARALISED NÕUDED**

Kuna openMosix on sedavõrd lihtne seadistada, siis puuduvad tal ka erilised eelnõudmised klastrile ja tema sõlmedele. Vaja on vähemalt kahte IA-32 arhitektuuriga arvutit, mis on omavahel ühendatud LAN-i abil. Kasutada võib nii ristiga kaablit, kommutaatorit („*switch*“) kui ka jaoturit („*hub*“). Igal arvutil peab olema installeeritud Linux tööjaam, mille tuuma versioon on vähemalt 2.4. OpenMosix peaks töötama ükskõik millise Linuxi distributsiooni peal. Võrgukaart peab olema korrektselt konfigureeritud. Ühtlasi peaks olema eraldatud piisav kogus ruumi saalesektsiooni („*swap*“) jaoks, kui palju täpselt, pole öeldud, aga mida rohkem seda parem.

### **KLASTRI SEADISTAMINE**

OpenMosixi ametlik dokumentatsioon [3] pakub välja kolm võimalust seadistada oma klaster.

Esimene võimalus kannab nimetust *single-pool*. Selle konfiguratsiooni korral kasutatakse kõiki servereid ja tööjaamasid ühes klastris. Iga masin on osa kobararvutist ja võib saata töid ükskõik millisele teisele sõlmele antud klastris. Samas peab ta ise ka töid vastu võtma.

Teist seadistust nimetatakse *server-pooliks*. Sellisel juhul moodustavad klastri ainult serverid, samas kui tööjaamad on eraldiseisvad arvutid, milledele pole vajalik installeerida openMosix tuuma. Et jooksutada rakendusi kobararvutil, peab kasutaja meldima serverisse. Sellise konfiguratsiooni korral ei migreerita protsesse tööjaamadesse.

Kolmandaks võimaluseks on *adaptive-pool*. See meenutab *server-pooli*, sest tavalisel ajal moodustavad klastri ainult serverid, aga kui kasutaja logib välja oma tööjaamast, siis käivitub selles arvutis *script*, mis muudab ta osaks klastrist.

Kuigi tavapärase openMosixi klastri struktuur on *sümmeetriline* (joonis 3), võib kerge vaevaga muuta selle struktuuri *asümmeetriliseks* (joonis 4). Näiteks *server-pool* seadistuse korral võib olla mugavam, kui kasutajad saavad logida sisse ainult ühte masinasse, mis omakorda migreerib protsessid edasi ülejäänud sõlmedele. Et muuta üks sõlm esiarvutiks, peab see masin uskuma, et ta on kõige aeglasem arvuti kogu klastris, siis ta migreerib kõik tööd edasi. OpenMosixi kasutajavahendid pakuvad võimalust muuta sõlme virtuaalset kiirust.

## **TARKVARA PAIGALDAMINE JA SEADISTAMINE**

OpenMosix klastri installimine toimub kahes osas: kõigepealt tuuma paikamine ja seejärel kasutajavahendite installimine. OpenMosixi kodulehelt (<http://openmosix.sourceforge.net>) ongi võimalik vajalikud failid kahe paketina alla laadida.

Ehitamiseks openMosix klastrit, on vaja installeerida openMosixiga laiendatud tuum kõigile kobararvuti sõlmedele. Tuuma installimiseks on kaks võimalust. Võib laadida alla juba eelnevalt kompileeritud tuuma ning installeerida selle. Teine võimalus on laadida alla originaal tuuma lähtekood. Sellisel juhul tuleb rakendada openMosix paikeprogrammi sellele lähtekoodile, kompileerida paigatud tuum ning seejärel viimane installeerida. OpenMosixi dokumentatsioon [3] sisaldab samm-sammult juhendeid nii mõnegi laialdaselt levinud Linuxi distributsiooni jaoks. Arvestama peab ka sellega, et kõikidel sõlmedel peab olema sama versioon paigatud tuumast. Sõlmedel ei pea olema sama tuuma kujutis („*image*“), vaid sama versiooni numbriga tuum. Seega võib erinevatel sõlmedel kompileerida tuumad uuesti, et toetada erinevat riistvara klastri sõlmedel.

Kui tuum paigatud, tuleks installeerida kasutajavahendid. Viimaste installimine ei tohiks käia üle jõu kellelgi, kes omab teadmisi Linux-süsteemide haldamisest.

OpenMosixil on automaatse avastamise („*auto-discovery*“) deemon nimega *omdiscd*, mis võimaldab automaatselt seadistada openMosixi klastrit. Seega ei ole vajalik käsitsi konfigureerimine. Sõlme automaatse avastamise deemon edastab pakette, mis annab teistele masinatele teada, et ta on openMosixi sõlm. Kui on soov lisada uus arvuti openMosix klastrisse, siis piisab sellest kui jooksutada *omdiscd* sellel masinal. Muidugi peab arvestama sellega, et arvutil oleks korrektselt seadistatud võrgukaart ja ruutimine.

### **KLASTRI HALDAMINE**

OpenMosixi tarkvarapakett sisaldab kõike, mida võib vaja minna antud klastrit töötamiseks ja juhtimiseks. Siiski on soovitatav installeerida ka üks kolmanda osapoole tarkvarapakett nimega openMosixview. See on ainult openMosix klastrit jaoks mõeldud graafilise kasutajaliidesega haldamise tarkvara. Täpsemalt on ta juba olemasolevate openMosix kasutajavahendite laiendus, mistõttu peavad viimased ka eelnevalt installitud olema. Ühtlasi peab arvestama, et openMosixview paigaldamine võib nõuda lisapakettide paigaldamist, et lahendada tekkivaid sõltuvusprobleeme.

## **2.2. ROCKS DISTRIBUTSIOON**

### **ÜLEVAADE**

Rocks distributsioon on kogum avatud lähtekoodil baseeruvat tarkvara, mille abil saab imelihtsalt üles seada HPC klastreid. Rocks meeskonna eesmärgiks on muuta kobardamine võimalikult lihtsaks tegevuseks ning läbi selle viia suuremahuliste teadusarvutuste võimalused kõigi teadlasteni. Arvutiklastrite loomine ja haldamine on endiselt väga keeruline ülesanne enamikele süsteemiadministraatoritele. Rocks pakub mitmeidki vahendeid, mis abistavad süsteemihalduril luua ja hooldada arvutusklastreid.

Rocks distributsioon sisaldab endas RedHat Linuxit, seega ei ole võimalik ega ka vajalik installeerida Rocks'i juba olemasoleva Linux distributsiooni peale. Rocks toetab x86, x86\_64 ja IA-64 protsessorarhitektuure ning kuna ta on ehitatud RedHat Linux'i peale, siis on tal olemas tugi kõigile riistvara seadmetele, mis on toetatud ka RedHat Linux'i poolt. See kindlasti muudab Rocks'i kasutamise mugavamaks. Võrguliidestest on toetatud nii *ethernet* kui ka *myrinet*.

Vaikeväärtustega installeerimisprotsess kulgeb ülikiiresti ja enamikel juhtudel probleemideta. Sellisel juhul tehakse teatavad seadistuse ja kasutatavate tarkvarapakettide valikud administraatori eest ära automaatselt.

Kuna Rocks kasutab sõlmede installeerimiseks nii *Kickstarti* kui ka *Anacondat*, siis saab ehitada heterogeenseid kobararvuteid, milles sisaldub erinevaid riistvara arhitektuure.

Rocks klaster kasutab asümmeetrilist struktuuri (joonis 6), kus kasutatakse esiarvutit ning ühte või enam arvutussõlme.

### **SEADISTAMINE JA HALDAMINE**

#### **RIIST- JA TARKVARALISED NÕUDED**

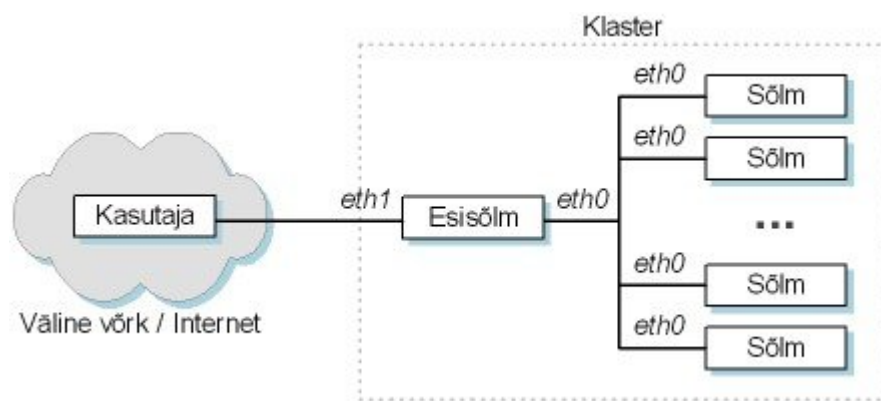
Kuna esiarvutil ja arvutussõlmel on täita erinevad rollid antud klasteri struktuuri korral, siis on neile ka erinevad nõuded. Üldiselt tuleb arvestada, et Rocks klasteri puhul ei ole võimalik, et arvutussõlmedel ei ole kõvakettaid, seega on vaja, et kõik klasteri sõlmed oleksid x86, x86\_64 ja IA-64 arhitektuuriga ning sisaldaksid kõvakettaid. Muidugi peab sõlmedel olema töötav võrguliideskaart.

Esiarvutil peab olema vähemalt 16 GB suurune kõvaketas ning põhimälu mahuga 512 MB. Lisaks on vajalik vähemalt kahe füüsilise väratiga („*port*“) etherneti liides.

Arvutussõlmel („*compute node*“) on samuti vajalik 16 GB suurune kõvaketas ja 512-megabaidine põhimälu, kuid piisab ka ühest füüsilisest etherneti väratist.

## KLASTRI SEADISTAMINE

Rocks klaster kasutab sümmeetrilist kobardamise struktuuri ning dikteerib sõlmede ühendamise järgnevalt:



**Joonis 6.** Rocks kobararvuti kohustuslik struktuur

Etherneti liides arvutussõlmedel (*eth0*) peab olema ühendatud klasteri etherneti kommutaatori külge. Seda võrku nimetatakse privaatseks, see tähendab, kõik selle võrgu liiklus on füüsiliselt eraldatud välisest avalikust võrgust ehk internetist.

Eesarvutil on vaja kahte etherneti liidest. Liides, millele Linux vastendab *eth0*, peab olema ühendatud samasse võrku, kuhu arvutussõlmedki ning *eth1* peab olema ühendatud välisesse võrku (näiteks: internetti või organisatsiooni *intranetti*).

Kui klaster on füüsiliselt valmis, võiks iga sõlme seada alglaadima ilma klaviatuuri toeta. See protseduur eeldab BIOS-e seadistuse muutmist.

## TARKVARA PAIGALDAMINE JA SEADISTAMINE

Rocks tarkvara installeerimiseks on vajalik vähemalt kolme erineva CD-kujutise allalaadimine Rocks ametlikult kodulehelt (<http://www.rocksclusters.org>). Alternatiivina võib allalaadida hoopis DVD-kujutise, mis sisaldab kahe CD jagu lisatarkvarapakette, mis ei ole otseselt vajalikud toimiva Rocks klasteri saamiseks.

Tarkvara paigaldus toimub kahes etapis: kõigepealt seadistatakse esiarvuti ning seejärel arvutussõlmed.

Ehitamaks Rocks kobararvutit on kõigepealt vajalik installeerida esiarvutile Rocks distributsioon. Installeerimine on üsna sarnane tavalise Linux distributsiooni paigaldamisega. Seega isikul, kes on edukalt suutnud paigaldada mõnda Linuxi distributsiooni, ei tohiks tekkida probleeme esimasinale tarkvara installeerimisega. Ühtlasi on Rocks'i kasutajajuhendis [5] põhjalik õpetus selle kohta. Kõige tähelepanelikum tasub olla installeerimisprotsessi alguses, kus on väga lühike hetk, mil kasutaja saab valida, kas ta soovib paigaldada tarkvara esiarvutile või arvutussõlmele. Vaikimisi eeldatakse, et paigaldada tuleb arvutussõlme tarkvara.

Arvutussõlmele tarkvara installeerimine peaks minema ülikiiresti, sest Rocks on töötanud palju selle nimel, et vähendada põhisõlmede seadistusele kuluvat aega. Nimelt on üks Rocks'i arenduse eesmärke, et kui mõni sõlm peaks töötamast lakkama, siis on kasulikum ta uuesti installeerida. Rocks klasteri arvutussõlme tarkvara paigaldamine kestab ajaliselt 10 - 20 minutit. Arvutussõlme tarkvarapaigaldamine võib toimuda nii CD kui ka PXE (*Preboot eXecution Environment*) liidese vahendusel.

## **KLASTRI HALDAMINE**

Rocks klasteri haldamise muudab eriti mugavaks sisseehitatud juhtimisvahendid. Vaikimisi on need vahendid kättesaadavad ainult klasteri seest, aga väikeste muudatuste tegemisel tulemüüris, saab võimalikuks ka kobararvuti haldamine üle avaliku võrgu. Haldamisvahendid on veebipõhised ning neile pääseb ligi kasutades ükskõik millist graafilise kasutajaliidese veebilehitsejat (näit *Mozilla Firefox*).

## 2.3. OSCAR

### ÜLEVAADE

OSCAR (*Open Source Cluster Application Resources*) on tarkvarakomplekt, mille eesmärgiks on lihtsustada klastrite loomist. Olles kogum avatud lähtekoodil baseeruvat tarkvara, sisaldab OSCAR kõike, mida on tõenäoliselt vaja HPC klatri loomiseks, muutes sellega lisapakettide allalaadimise ja installeerimise mittevajalikuks. Kui laadida alla, installeerida ja jooksutada OSCAR, siis tulemuseks on täielikult töötav kobararvuti.

OSCAR on eelkõige mõeldud arvutusklastrite loomiseks, kuid vähese vaevaga on seda kasutades võimalik luua ükskõik millist tüüpi kobararvuteid. Loodud on ka alamarendusgrupp ja käivitatud projekt HA-OSCAR, mis tegeleb hõlpsat juuredpääsu tagavate arvutiklastritega.

Erinevalt Rocks klastrist on OSCAR tarkvara kasutajal teatav vabadus Linux distributsiooni valiku osas. Arendus käib selles suunas, et OSCAR oleks täielikult distributsioonist sõltumatu.

OSCAR pakett on valik faile, mida kasutatakse tarkvara installeerimiseks OSCAR klastrile. See pakett võib olla lihtne, näiteks üks RPM fail, või hoopis keeruline, näiteks kogum mitmest RPM ja kõrvalistest installeerimise/seadistamise failidest. OSCAR paketid tagavad suure osa OSCAR klastrite funktsionaalsusest. Nad jagunevad kolme rühma:

- Aluspaketid („*core packages*“) on vajalikud OSCAR-i enda tööks, seega peavad saama installeeritud
- Lisapaketid („*included packages*“) on ametlikult OSCAR-iga kaasa pandud. Enamasti on nende pakettide arendajad ka OSCAR-i arendajad ning on läbinud mingil määral ametlikku testimist enne välja laskmist.
- Kolmanda osapoole paketid („*third party packages*“) ei sisaldu ametlikus OSCAR komplektis. Nad on lisandprogrammid („*add-on*“), mida saab lahti pakkida OSCAR puuse ja seetõttu installeerida kasutades OSCAR raamistikku.

## SEADISTAMINE JA HALDAMINE

### RIIST- JA TARKVARALISED NÕUDED

Kuna OSCAR ei sisalda ise Linux distributsiooni, siis peab selle valiku tegema süsteemiadministraator. Ametlikult toetab OSCAR järgnevaid Linux distributsioone ja vastavaid riistvara arhitektuure:

- Fedora Core 3 (x86)
- Fedora Core 2 (x86)
- Mandriva Linux 10.1 (x86)
- Mandriva Linux 10.0 (x86)
- RedHat Enterprise Linux 4 (x86, IA-64, x86\_64)
- RedHat Enterprise Linux 3 (x86, IA-64, x86\_64)

Täielikult toetatud väljalasete kloonid, eriti RedHat Enterprise Linux'i lähtekoodil baseeruvad tasuta distributsioonid nagu CentOS ja Scientific Linux peaksid töötama, aga pole ametlikult testitud.

OSCAR klaster koosneb ühest server-sõlmest ja ühest või enamast klient-sõlmest. Hetkel peab kõigil klient-sõlmedel olema homogeenne riistvara, aga see peaks tulevikus muutuma. Täpselt nagu Rocks klaster korral, on ka siin erinevad nõudmised esiarvutile ja tavalistele sõlmedele.

Vastavalt ametlikule dokumentatsioonile [6] peab esisõlme arvuti protsessor olema i586 või uuemat tüüpi. Kui klaster on plaanitud ühendada avalikku võrku, siis on vajalik vähemalt kahe TCP/IP toega võrguliideskaardi olemasolu esisõlmel. Kindlasti peab olema vähemalt 4 GB vaba kõvakettaruumi ning *täielikult toetatud* Linux'i distributsioon. Võib proovida kasutada mõnda muud Linux'i distributsiooni, aga sellisel juhul ei pruugi OSCAR klaster tööle minna.

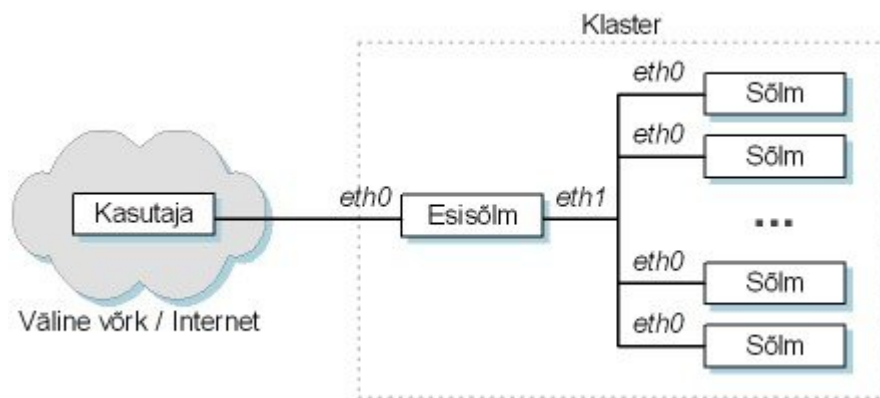
Klient-sõlmedel tuleb tähelepanu pöörata sellele, et kõik masinad oleksid homogeense arhitektuuriga ning neil kõigil peab olema installeeritud sama versioon ja distributsioon Linuxist, mis esisõlmelgi. Kõvakettamahtu peab olema vähemalt 2 GB ning igal



masinal peab olema vähemalt üks TCP/IP pinu toetav võrgukaart. Lisaks on vajalik PXE või *floppy*'ga juhitud BIOS.

## KLASTRI SEADISTAMINE

Kuigi OSCAR kasutab asümmeetrilist klasteri struktuuri (joonis 7), ei ole seda tüüpi kobara ehitamine nii range nõue kui Rocks klasteri puhul. Tähelepanu tuleks juhtida veel ühele nüansilisele erinevusele Rocks ja OSCAR seadistuste vahel. OSCAR vastendab avalikule võrgule *eth0* liidese ning *eth1* liides vastendub võrgule, kuhu on ühendatud kõik antud kobaarvuti sõlmed.



**Joonis 7.** OSCAR kobaarvuti asümmeetriline struktuur

## TARKVARA PAIGALDAMINE JA SEADISTAMINE

Esisõlme installeerimise esimeseks osaks on Linux distributsiooni paigaldamine antud masinasse. Seda muidugi juhul, kui arvutist puudub sobiv Linux distributsioon. Kuigi OSCAR vahevara võib paigaldada juba olemasolevale süsteemile, on siiski soovitatav installeerida ta uue e. puhta distributsiooni peale. On tugevalt soovitatav, et operatsioonisüsteemi uuendused installeeritakse peale OSCAR-i paigaldust, et vältida võimalikke tarkvara konflikte.

Laadimaks alla uusimat OSCAR tarkvara väljalaset, tuleks minna ametlikule OSCAR allalaadimise leheküljele (<http://oscar.openclustergroup.org/download>). Seal on valida kolme erineva OSCAR lindiarihiivi („*tarball*“) vahel.

- „*Regular*“ – kõik tavapärased paigaldusvahendid, mida tavaline kasutaja vajab OSCAR klasteri loomiseks ja haldamiseks.

- „*Extra Crispy*“ – sama mis *regular*, aga lisatud on enamike RPM-ide SRPM-id. SRPM-id on RPM arhiivid koos lähtekoodiga, nad ei ole vajalikud OSCAR klasteri tavapäraseks loomiseks ja töötamiseks. Antud pakett on suurem kui *regular* pakett ning mittevajalik enamikele kasutajatele.
- „*Secret Sauce*“ – see väljalase sisaldab ainult SRPM faile. Eelkõige mõeldud neile kasutajatele, kes algselt laadisid endale *regular* paketi ja hiljem otsustasid, et soovivad ka lähtekoodi.

Kuigi OSCAR klasteri tarkvara paigaldamiseks on väljaarendatud korralik graafiline kasutajaliides, peab arvestama, et OSCAR klasteri loomine on natuke keerulisem kui Rocks klasteri ülesseadmine. Süsteemilooja peab tundma loodavat süsteemi võrdlemisi hästi ning vajalikud on ka keskmisest suuremad teadmised võrgutehnoloogiast. OSCAR kobararvuti ehitamine toimub ideaalis ainult esisõlme seadistades. Selle käigus valmivad ka sõlmedele vajalikud süsteemikujutised, mis PXE alglaadimise abil klient-sõlmedele edastatakse. Kui klient-sõlmedel puudub PXE alglaadimise tugi, siis on võimalik luua nendest süsteemikujutistest ka ISO kujutised, mida saab kerge vaevaga CD plaadile kirjutada ning niimoodi klient-sõlmedele tarkvara paigaldada.

OSCAR tarkvara paigaldusjuhendis [6] on väga detailselt kirjeldatud kogu paigaldusprotsessi, seega on ülimalt kasulik selle dokumendiga tutvumine.

## **KLASTERI HALDAMINE**

Kasutades OSCAR kobararvuti tarkvara paigaldamise *wizard*it, on võimalik eemaldada ja lisada klient-sõlmesid antud klasterisse. Põhiliseks haldusvahendiks on aga vabavaraline avatud lähtekoodil põhinev C3 ehk *Cluster Command and Control*. C3 sisaldab üle kümne käsurea utiliidi, mis mugavdavad arvutiklasteri juhtimist märkimisväärselt. Need käsurea utiliidid ei ole tegelikult midagi enam kui spetsiifilised Python keeles kirjutatud skriptid.

C3 on vaikimisi OSCAR-i juhtimise tarkvara ning seega ka osa aluspakettidest, kuid soovi korral võib kasutada mõnda muud haldamise tarkvara (näit *Ganglia* või *Columon*).

## **2.4. ClusterKNOPPIX**

### **ÜLEVAADE**

Kõige kiirem moodus töötava arvutiklastri loomiseks on nn. kiirklasteri („*instant clusters*“) tarkvara kasutamine. Tegu on Linux *live* CD distributsioonidega, millesse on juba sisseehitatud klastrite tarkvara – tavaliselt openMosix.

*Live* CD väljalasked on sellised Linuxid, mille puhul ei ole vajalik operatsioonisüsteemi paigaldamine arvuti kõvakettale. Tuleb lihtsalt sisestada arvutisse *live* CD ning seejärel süsteem selle abil alglaadida.

### **MIS ON KNOPPIX, CLUSTERKNOPPIX?**

KNOPPIX on buuditav *live* süsteem, mis on paigutatud kas CD või DVD plaadile, ning ta sisaldab tüüpilist valikut GNU/Linux tarkvarast. Samuti sisaldab ta automaatset riistvara avastamise süsteemi ning sisseehitatud tuge mitmele helikaardile, graafikakaardile, SCSI ja USB vahendile ning muudele perifeeriaseadmetele. KNOPPIXit võib kasutada tavalises töölauarvutis, õpetamise eesmärgil, süsteemi taastamise vahendina või tarkvara demonstratsioonide alusplatvormina. KNOPPIX kasutab kohest lahtipakkimise viisi, mille abil võib CD sisaldada kuni 2 GB käivitavat tarkvara ning DVD isegi üle 8 GB tarkvara.

ClusterKNOPPIX ei ole midagi muud, kui modifitseeritud KNOPPIX väljalase, mis kasutab openMosix tuuma. Lisatud on ka openMosixi kasutajavahendid ning openMosixview.

### **SEADISTAMINE JA HALDAMINE**

ClusterKNOPPIXi abil kobrarvuti loomine on imelihtne – tuleb ainult kõik kohtvõrku ühendatud sõlmed *live* CD abil käima laadida. Arvestama peaks, et sellist süsteemi ei saa kasutada LB ega HA klastrite puhul, sest klaster püsib kuni järgmise alglaadimiseni. Soovi korral on võimalik ClusterKNOPPIXi paigutamine kõvakettale. Selle kohta leiab vastava juhendi ClusterKNOPPIXi kodulehelt.

Klastri haldamiseks on ClusterKNOPPIXile lisatud nii openMosix kasutajavahendid kui ka openMosixview.

## 2.5. BCCD

### ÜLEVAADE

BCCD (*Bootable Cluster CD*) on veel üks näide kiirkobardamise tarkvarast. BCCD orientatsioon on eelkõige õppimisele suunatud. BCCD võimaldab luua arvutiklastreid juba eelseadistatud kohtvõrkudes. Ta ei mõjuta arvutis eelnevalt olemasolevaid seadistusi, seega ideaalne kasutamaks koolide arvutiklassides kobardamist näitlikustava materjalina. Põhiline erinevus ClusterKNOPPIXist seisneb selles, et BCCD sisaldab ka muid arvutiklastrite tarkvaravahendeid peale openMosixi ning openMosixview. Järgnev on kokkuvõttev nimekiri lisatud vahenditest:

- Linuxi tuum 2.4.25.
- gcc, g77 ja palju muid arendusvahendeid.
- C3 (Cluster Command and Control) vahendid.
- MPICH, LAM-MPI ja PVM.
- OpenMosix ja kasutajavahendid ning openMosixview.
- Täielik *openPBS Scheduleri* tugi.
- Octave, Mozilla Firebird ja veel umbes 1400 kasutajavahendit.
- Kohtvõrgu seadistamise vahendid.

### SEADISTAMINE JA HALDAMINE

BCCD klasteri ülesseadmine toimub praktiliselt sama rutiini alusel, mis ClusterKNOPPIXilgi. Tuleb alla laadida CD kujutis, see plaadile kirjutada ning selle abil sõlm käima laadida. Haldamiseks on siin siiski rohkem vahendeid kui ClusterKNOPPIXil – lisatud on ka C3.

## KOKKUVÕTE

Käesoleva bakalaureusetöö eesmärgiks oli tutvustada lihtsaid tarkvaralists vahendeid, mille abil saab kiiresti ning lihtsalt luua ja hooldada Linux operatsioonisüsteemiga arvutusklastreid. Kasutajatele sai näidatud, et väikese ja keskmise suurusega arvutusklastri ehitamine ja hooldamine ei pea olema väga keeruline protsess.

Lõputöös on lühidalt selgitatud üldist kobararvutite teooriat – tutvustatud erinevaid arvutiklastrite tüüpe, kirjeldatud, kus neid kasutatakse. Ühtlasi on käsitletud kolme erinevat klastrite loomiseks vajalikku tarkvarakomplekti, lisaks on vaadeldud ka kahte kiirkobardamise tarkvara näidet.

Seega on valminud lühike, kuid ülevaatlik kirjeldus arvutiklastrite loomiseks ja haldamiseks vajamineva tarkvara hetkeseisust ning võimalustest.

## VIITED

1. J. D. Sloan, *High Performance Linux Clusters with OSCAR, Rocks, OpenMosix, and MPI* (O'Reilly, 2004).
2. Computer Cluster – Wikipedia, the free encyclopedia. (2006) [WWW]  
[http://en.wikipedia.org/wiki/Computer\\_cluster](http://en.wikipedia.org/wiki/Computer_cluster) (28.04.2006).
3. Kris Buytaert, *The openMosix HOWTO*. (03.03.2005) [PDF]  
<http://howto.x-tend.be/Mosix-HOWTO/openMosix-HOWTO-download/openMosix-HOWTO.pdf.gz> (01.05.2006).
4. Kris Buytaert, „Introducing openMosix“. (19.02.2004) [WWW]  
<http://www.linuxdevcenter.com/pub/a/linux/2004/02/19/openmosix.html>  
(07.05.2006).
5. Rocks Cluster Distribution: Users Guide: User's Guide for Rocks version 4.1 Edition. (UC Regents, 2006) [PDF]  
<http://www.rocksclusters.org/rocks-documentation/4.1/rocks-usersguide-4.1.pdf>  
(27.04.2006).
6. How to Install an OSCAR Cluster. Software Version 4.2. Documentation Version 4.2. (The Open Cluster Group, 28.10.2005) [PDF]  
<http://oscar.openclustergroup.org/public/docs/oscar4.2/oscar4.2-install.pdf>  
(27.04.2006).
7. What is KNOPPIX®? (2006) [WWW]  
<http://www.knoppix.com> (23.05.2006).
8. ClusterKnoppix. (2006) [WWW]  
<http://clusterknoppix.sw.be> (23.05.2006).
9. Bootable Cluster CD. (2005) [WWW]  
<http://bccd.cs.uni.edu> (23.05.2006).

# ABSTRACT

## SOFTWARE PACKAGES FOR CREATION AND MANAGEMENT OF COMPUTER CLUSTERS

Karl Kruusamäe

This bachelor thesis consists of two main parts. In the first part different multiprocessor systems are introduced, in the second part - software packages for creation and management of computer clusters are discussed.

Multiprocessor systems are divided into two main groups: centralized multiprocessors, and multi computer systems. Centralized multiprocessor systems cover the types of computer, which use more than one CPU. The basic problem, concerning centralized multiprocessors, is how to organize memory use, since ordinarily all the processors have to share the memory.

Multi computer system, or computer cluster, is a group of loosely coupled computers that work together closely so that in many respects they can be viewed as though they are a single computer.

Computer clusters are divided into four categories depending on the application area. There are high-availability clusters, load balancing clusters, high-performance computing clusters and grid clusters.

In this thesis, five different clustering software products are reviewed – openMosix, Rocks Cluster Distribution, OSCAR, ClusterKNOPPIX, and BCCD.

OpenMosix is software that extends the Linux kernel so that processes can migrate transparently throughout the cluster, distributing the workload between different nodes of the system.

Rocks Cluster Distribution is a collection of open-source software for creating and managing high-performance computer clusters. Since it is built on RedHat Linux, it is impossible to install Rocks on top of existing Linux distribution.

OSCAR (*Open Source Cluster Application Resources*) is a software bundle that contains loads of different software packages that might be needed in order to

successfully create and manage a computer cluster. Unlike Rocks, it is required to install OSCAR on top of an existing Linux distribution.

ClusterKNOPPIX and BCCD (*Bootable Cluster CD*) are live system CDs, which make it possible to build a temporary cluster on top of existing computer system without making any lasting configuration changes to it. They both use openMosix kernel for distribution of workload but BCCD has a few more tools for parallelism than ClusterKNOPPIX.

This bachelor thesis has hopefully brought clustering closer to the system administrators who so far have considered it to be a too difficult task.