

Aalto-yliopisto
Perustieteiden korkeakoulu
Teknillisen fysiikan ja matematiikan tutkinto-ohjelma

Ontologiamuokkaimen käyttö laskentaklusterin tehokkuusanalyysissä

kandidaatintyö
25.3.2013

Santtu Klemetilä

Työn saa tallentaa ja julkistaa Aalto-yliopiston avoimilla verkkosivuilla.
Muilta osin kaikki oikeudet pidätetään.

AALTO-YLIOPISTO PERUSTIETEIDEN KORKEAKOULU PL 11000, 00076 Aalto http://www.aalto.fi	KANDIDAATINTYÖN TIIVISTELMÄ	
Tekijä: Santtu Klemetilä		
Työn nimi: Ontologiamuokkaimen käyttö laskentaklusterin tehokkuusanalyyssissä		
Tutkinto-ohjelma: Teknillisen fysiikan ja matematiikan tutkinto-ohjelma		
Pääaine: Systemitieteet	Pääaineen koodi: F3010	
Vastuopettaja(t): Prof. Ahti Salo		
Ohjaaja(t): FT Markopekka Niinimäki		
<p>Tiivistelmä:</p> <p>Tämän kandidaatintyön tavoitteena on arvioida ontologiamuokkainta hajanaisen ja heterogeenisen datan integrointivälineenä. Sovelluksen jatkotoimenpiteenä on tarkoitus analysoida ontologiamuokkaimella kartoitetun erään laskentaklusterin energiankulutus- ja työnsuoritusdataa.</p> <p>Teoriaosuudessa esitetään OLAP-ontologioiden peruseriaatteet sekä tarkasteltavan laskentaklusterin perustiedot. Tutkimusmenetelmäosiossa muodostetaan ontologiamuokkaimella laskentaklusteriontologia, johon kartoitetaan kahdentyyppistä dataa. Kartoitettu data saatetaan R-tilastolaskentaohjelmistolla analysoitavaan muotoon.</p> <p>Työn tulososiossa erityyppiset datat yhdistetään aikaleimojensa perusteella ja niistä luodaan regressiomalli, jossa laskentaklusterin energiankulutuksesta selittävinä tekijöinä ovat laskentatöiden lukumäärä, muistinkäyttö ja input/output-dataliikenteen määrä. Energiankulutuksessa huomataan suuri lepoteho, eikä koneiden suoritämäärällä ole kovinkaan suurta vaikutusta laskentaklusterin kokonaisenergiankulutukseen.</p> <p>Datasta lasketaan myös koneiden teoreettinen vähimmäistarve, jonka ylittämisen seurauksia tutkitaan toisaalta energiansäästön kannalta ja toisaalta laskentatöiden läpikulkuajan muuttumisen kannalta. Kun oletetaan laskentatyöt täydellisesti ositettavaksi ja käyttäjien odotusajan minimointi merkityksettömäksi, saadaan laskentaklusterin energiankulutukselle 74 prosentin potentiaalinen säästö.</p>		
Päivämäärä: 25.3.2013	Kieli: suomi	Sivumäärä: 19 + 2
Avainsanat: dataintegraatio, OLAP-ontologia, laskentaklusteri, energiansäästö		

Sisältö

1	Johdanto	1
2	Teoreettinen tausta	2
2.1	Ontologia ja RDF	2
2.2	Klusteridata	4
3	Tutkimusongelma ja -menetelmät	5
3.1	Ontologian määrittely ja kartoitus	5
3.2	Klusteridatan analyysi	9
4	Tulokset	10
5	Tarkastelu	15
A	Ontologialinkit	20
B	Suurennoskuvia analyysin kuvaajista	21

1 Johdanto

Tiedonkäsittelyssä ongelma on usein informaation hajanaisuus ja datalähteiden heterogeenisyys. Ihminen osaa yleensä erotella esitetystä datasta sen ulottuvuudet ja arvot ilman ulkopuolelta tulevia määrittelyjä, mutta tietokoneelle tällainen informaation poiminta on vaikeampaa. Tämän vuoksi on kehitetty tiedonvaihtoon sopivia malleja, joilla tietoa voidaan siirtää ohjelmasta toiseen tietokoneille ymmärrettävämmässä muodossa, jolloin edelleenkäyttö ei vaadi ihmisen avustusta. Käyttäjälle esitettävä välitettävä skeema (mediated schema) mahdollistaa datalähteiden suhteen kartoitettuna dataintegraation ja sitä kautta tiedon hyväksikäytön [4].

CERN:issä toimivan HIP:n (Helsinki Institute of Physics), Tampereen yliopiston ja Lausannen yliopiston yhteisprojekti EXHADA (Extracting, Harmonizing, and Analyzing Heterogenous and Distributed Data) on laaja projekti tällaisen hajanaisen ja heterogeenisen datan poimimiseen, yhtenäistämiseen ja analysoimiseen tarkoitettujen mekanismien ja tekniikoiden kehittämiseksi [11]. Tämän työn perustana on projektin yksi osatyökalu.

Jotta tietoa voidaan tunnistaa, on aluksi määriteltävä siihen liittyvien osasten merkityksiä ja merkitysvyyksiä. Tähän liittyy filosofiasta tietojenkäsittelytieteisiin omaksuttu hieman muunneltu käsite ontologia, jossa sovellusalueeseen liitetään halutut käsitteet ja käsitesuhteet [3]. Tietokoneet eivät itsestään ymmärrä esimerkiksi sanan "Pasi" liittämistä joukkoon "ihmiset", minkä vuoksi ontologiamäärittelyt ovat tärkeitä eri tietolähteiden yhdistelyssä kokonaisuudeksi. Työssä käytettävä työkalu on ontologioiden määrittelyyn ja muokkaamiseen kehitetty ontologiamuokkain, jonka käytön tavoitteena on mallintaa kokeellisen fysiikan laskentatöiden suoritusta.

Ontologiasta voidaan muodostaa edelleen siirrettävä malli, jossa ilmaisu on standardoidun omaista, jolloin lähteenä olevaa dataa voidaan kartoittaa liittämällä datajoukkoja mallin osajoukkoihin. Ontologiatyökalulla malli tallentuu W3C:n (World Wide Web Consortium) standardoimassa RDF-muodossa (Resource Description Framework) [1], josta ontologiatieto voidaan tavoitteiden mukaisesti välittää tietokoneohjelmien kesken ja käyttää tietoa hyväksi ontologiaan liittyvän datan analyysissä, kun ontologiamallin pohjalta voidaan siihen linkitetystä datasta poimia ja käsitteistää kunkin tilanteen vaatima osuus [10].

Tässä työssä ontologiamuokkainta käytetään Euroopan hiukkasfysiikan tutkimuskeskuksen CERN:in klusteridatan kartoittamiseen ja siirtämiseen analysoitavaksi R-tilastolaskentaohjelmistoon. Tietokoneklusterista kerätty datalogi sisältää tietoja klusteriin lähetetyistä laskentatöistä, joiden ominaisuuksia tässä työssä tilas-

tollisesti tarkastellaan. Suurten työmäärien tapauksessa voidaan tutkia tehtävien jakautumista esimerkiksi kellonaikojen ja serverikoneiden suhteen, jolloin voidaan arvioida laskentatehon käytön optimaalisuutta. Tarkastelun kohteena on myös tietokoneklusterin energiankulutusdataa, jota on kerätty samalta ajalta laskentatyödatan kanssa. Yhdistämällä datat voidaan tarkastella erilaisten laskentatyökokonaisuuksien kokonaisenergiankulutusta.

2 Teoreettinen tausta

2.1 Ontologia ja RDF

Tietojenkäsittelyssä tärkeä ongelma on tietokoneilta puuttuva äly. Luonnollisen tekstin jäsentäminen ja kontekstuaalisen tiedon tulkitseminen tekstin yhteydestä on tietokonesuoritteisesti erittäin hankalaa. Usein tieto on myös esitetty lähteissä eri tavoin, minkä vuoksi tiedon yhdistely eri formaateista ei ole välttämättä kovinkaan suoraviivaista. Tämä kaikki monimutkaistaa tiedon hakemista ja yhdistelyä sekä näiden kahden automatisointia [5].

Tämä ongelma tiedostettiin jo 1990-luvulla, minkä vuoksi Web-standardeja kehittävä W3C tunnisti tarpeen alkaa kehittää semanttista webiä, jossa ideana on webin sisällön merkityksen kertovan tiedon tuottaminen ja välittäminen [8]. Semanttisen webin kannalta tärkeää on kuhunkin asiakokonaisuuteen liittyvät sanastot ja muuttujasuhteet, joilla asiasisällöt voidaan jakaa ja sovittaa tiettyihin ryhmiin ja merkityksiin. Näitä käsittekokonaisuuksia kutsutaan ontologioiksi.

Ontologiamäärittelyissä ei ole automaattisesti mitään universaalia jaottelua käsitteille, joka vähentäisi datan käsittelyyn liittyvää ongelmaa. Monille ympäristöille on kuitenkin määritelty ontologioita, joita on pyritty standardoimaan, jotta ontologioinnista olisi laajempaa hyötyä. Esimerkiksi ihmisten sairauksille tehty niin kutsuttu Disease Ontology helpottaa tautien ja niihin liittyvien oireiden kartoittamisessa sopiviin lääketieteellisiin koodeihin [2].

Ontologia on kuitenkin vasta käsittekokonaisuus, johon ei sinänsä suoraan vielä liity mitään esitysformaattia. Resource Description Framework (RDF) on W3C:n standardoima malli tiedon jäsentelyyn, joka on yleisesti käytössä ontologioiden yhteydessä. RDF:ssä tieto jäsenellään lausumiin, jotka muodostetaan subjekti-predikaatti-objekti -kolmikolla. Jäsentelyyn ilmaiseamiseen käytetty esitysformaatti on yleisimmin XML-pohjainen (Extensible Markup Language) RDF/XML-formaatti. Esimerkiksi RDF:n määrittelysivulle [1] voitaisiin muodostaa toteama "Dokumentin (subjekti) toimittaja on (predikaatti) Dave Beckett (objekti)". Myös Dave Bec-

ket voidaan määrittää olioksi, jolle voidaan määritellä esimerkiksi toteama "Toimittajaolion nimi on Dave Becket". Tämä kirjoitettu RDF/XML-muotoon on seuraavaa:

```
<rdf:Description rdf:about="http://www.w3.org/TR/REC-rdf-syntax">
  <ex:editor>
    <rdf:Description>
      <ex:fullName>Dave Beckett</ex:fullName>
    </rdf:Description>
  </ex:editor>
</rdf:Description>
```

Tässä työssä käytetään OLAP-metodeille perustuvia OLAP-ontologioita (määritelty [9]). OLAP-ontologia muodostuu olioita määrittelevistä ulottuvuuksista (*dimension*) ja olioista tunnistettavista mitoista (*measure*) sekä näiden välisistä riippuvuuksista. Ulottuvuudet (kuten *User*) sisältävät tiedoissaan sisäisen hierarkiansa (*User: Owner → Group → Department*), joista määräytyy ontologiaan kuuluvat olioiden ominaisuuksia jaottelevat luokat (*Owner, Group, Department*), joille hierarkia-asettelun mukaan asetetaan ominaisuuksia (*Owner*-luokalla ominaisuus *hasGroup*). Mitat (työn kesto *wallclock*) sisältävät tiedoissaan riippuvuutensa ulottuvuuksiin (työn alkamisaika *start_time*), joiden perusteella tiettyä mittaa tarkasteltaessa voidaan päättää raakadatasta poimittavat merkitykselliset tiedot. Olennaista ontologiassa on dataolioiden määrittäminen ulottuvuuksista määräytyvien luokkien perusteella, jolloin raakadata voidaan kartoittaa ontologian ulottuvuuksien suhteen.

OLAP-ontologioiden mittoihin on käsittelyn mahdollistamiseksi aina liitettävä summautuvuustyyppi (engl. *summarizability type*). Mitoille on määritelty tyypit varasto (*stock*), virta (*flow*) ja yksikköarvo (*value-per-unit*) perustuen niiden kykyyn summautua ajan yli [12]. Varastotyyppistä mittaa ei voida summata ajan yli, koska se viittaa mitan tilaan tiettyä ajanhetkenä (esimerkiksi varastossa olevan polttoaineen määrä). Sitä vastoin virtatyyppistä mittaa voidaan summata, koska se kertoo tietyn aikavälin sisällä mitatun arvon (polttoainemyynti tiettyä vuonna). Yksikköarvotyyppisellä mitalla on varastotyyppisen mitan tavoin yhteen ajanhetkeen kiinnitettävä arvo eikä se summaudu ajan yli, mutta se kertoo yhden yksikön arvon (polttoaineen litrahinta) eikä yksiköiden määrää. Mittojen summautuvuuden aikariippuvuuden vuoksi myös ulottuvuudet on tapana jaotella temporaalisiin ja ei-temporaalisiin dimensioihin.

2.2 Klusteridata

Nykyaikainen hiukkastutkimus edellyttää hiukkasten kvanttimekaanisuuteen liittyvän tilastollisen luonteen vuoksi suurta määrää tapahtumakirjauksia. Datan suuresta määrästä johtuen on myös datasta tehtävän analyysin laskentaan käytettävällä laskentakoneistolla raskas kuormitus ja se voi vaatia suurenkin joukon koneita. Koska resurssit ovat aina kuitenkin rajallisia, on tärkeää optimoida sekä itse suoritettavia laskentatöitä että laskentakoneiston käyttöä. Laskentatyöt ovat yksilöllisiä ja eri käyttäjien muodostamia, joten niiden toimintaan on sinänsä hankala puuttua koneiston kautta, kun taas koneiston käytöstä voidaan säätää tehtävien jakoa eri koneille sekä tehtävien ajoitusta. Tällaista yhdistettyjen tietokoneiden joukkoa, jota toisaalta voidaan tarkastella yhtenä systeiminä, kutsutaan tietokoneklusteriksi.

Tässä työssä tarkasteltava klusteridata on dokumenteissa [7] ja [6] esitellyn CMS- ja ALICE-kokeiden käytössä olevan Jade-klusterin dataa. Klusteri on arkkitehtuuriltaan yhtenäinen pitäen sisällään 64 konetta, joissa jokaisessa on kaksitoista 2,4 GHz:n prosessoria ja 32 GB muistia. Jononhallintaan klusteri käyttää Sun Grid Enginen versiota 6.2u4. Ohjelmisto antaa yhdelle prosessorilla enintään yhden työn kerrallaan.

Ensimmäinen data on noin puolen vuoden jaksolta ja se sisältää tietoja yhteensä noin 850 000 laskentatyöstä. Data sisältää laskentatöiden accounting-muotoista dataa, jonka muoto on selitetty dokumentissa [13]. Siitä saadaan laskentatöiden aikatauluun liittyvää tietoa, työn käyttäjän ja suorittavan koneen tunnisteita sekä työn suoritusta kuvaavia lukuja esimerkiksi käytetystä muistista ja siirretystä datasta. Tarkastelu keskittyy tässä työssä siten nimenomaan laskentakoneiston käytön tutkimiseen, sillä tietoa laskentatöiden sisällöstä ei ole. Tämä data on otoskooltaan suurempi, joten käytämme sitä klusterin käyttöluonteen analyysiin. Esimerkkinä accounting-datasta sen nimikerivi ja yhden laskentatyön datarivi:

```
qname:hostname:group:owner:job_name:job_number:account:priority:
  submission_time:start_time:end_time:failed:exit_status:
  ru_wallclock:ru_utime:ru_stime:ru_maxrss:ru_ixrss:ru_ismrss:
  ru_idrss:ru_isrss:ru_minflt:ru_majflt:ru_nswap:ru_inblock:
  ru_oublock:ru_msgsnd:ru_msgrcv:ru_nsignals:ru_nvcsw:
  ru_nivcsw:project:department:granted_pe:slots:task_number:
  cpu:mem:io:category:iow:pe_taskid:maxvmem:arid:ar_submission_time
arc:c26.local:ndgfops:ndgfops:arc_testjob_rls:349:sge:0:1268406240:
  1268406246:1268406288:0:0:42:0.418936:0.221966:0.000000:0:0:0:0:
  43639:0:0:0.000000:0:0:0:0:3315:184:NONE:priority:NONE:1:0:
  0.640902:0.020267:0.017791:-U priority -q arc -l h_cpu=900,
  h_rt=900,h_vmem=1000M,s_cpu=900,s_rt=900,s_vmem=1000M:0.000000:
  NONE:531939328.000000:0:0
```

Toinen datapaketti on saman tietokoneklusterin dataa vuoden myöhemmältä kuu-kauden mittaiselta ajanjaksolta sisältäen tiedot noin 500 000 työstä. Tähän data-pakettiin kuuluu accounting-datan lisäksi myös energiankulutusdataa. Nämä yh-distämällä voidaan vertailla erityyppisten laskentatöiden määrittämien ajanjakso-
jen keskimääräistä energiankulutusta ja tällä tavoin koettaa arvioida laskentatöi-
hin liittyvien tunnuslukujen yhteyttä energiankulutuksen muutoksiin. Energian-
kulutuksesta oleva tieto on kahden tunnin välein tallennettu keskimääräinen ener-
giankulutus, eikä sitä ole kohdistettu kuin 16 laskentatietokoneen pakelille, minkä
vuoksi emme käytä energiankulutusdataa yksittäisten laskentatöiden tarkasteluun.
Energiadatan ollessa vain osalle koneista, energiankulutusanalyysissä käytettävien
töiden määrä pienenee 10 000:een. Energiadatasta esimerkkinä sen nimikerivi ja
yhden aikajakson datarivi:

```
"Date","Power source 1","Power source 2","Power source 3","Power source
  4","Power source 5","Power source 6"
"2011-01-21 16:00:00","7.0704125000e+02","7.0704125000e+02",
  "0.0000000000e+00","7.0704125000e+02","0.0000000000e+00",
  "7.0704125000e+02"
```

Dokumentissa [7] kerrotaan laskentatöiden jakautuneisuudesta Monte Carlo -simulaatioihin ja analyysitöihin. Monte Carlo -simulaatioissa suorittamisajat ovat yleensä pit-
kiä, mutta vaativat vain vähän input/output-dataliikennettä (I/O), kun samaa
pienekköä ajoa suoritetaan vain erittäin monta kertaa peräkkäin. Analyysityöt
taasen vaativat suurta liikennettä levysysteemiltä koneelle, kun mittausdataa on
siirrettävä laskentakoneen käyttöön. Tämän vuoksi analyysityöt aiheuttavat suur-
ta I/O-liikennettä. Tämä kahtiajakautuneisuus tekee analyysistä moniulotteisem-
man vaatiessaan usemman tekijän huomioonottamisen.

3 Tutkimusongelma ja -menetelmät

Tähän työhön liittyy kaksi vaihetta: dataan liittyvän ontologian määrittäminen ontologiamuokkaimella ja data-analyysi R-ohjelmistolla. Laajan ja hajanaisen da-
tan tapauksessa ontologian määrittäminen ja tämän jälkeen ontologiamuokkaimel-
la tiedon vieminen laskentaohjelmistoon mahdollistaa analyysivaiheen, jossa data-
komponentit ovat paremmin käsiteltävissä.

3.1 Ontologian määrittely ja kartoitus

OlapEdit on osana EXHADA-projektia kehitetty ontologiamuokkain. Sen tarkoi-
tuksena on olla nopea ja yksinkertainen työkalu, jolla voidaan määritellä onto-

logioita, kartoittaa dataa ontologioiden suhteen sekä viedä kartoitukset eteenpäin tilastolaskentaohjelmistolle. Tämä kaikki tapahtuu graafisen käyttöliittymän kautta, jota ajetaan palvelinkoneella verkkoselaimen kautta.

Ontologia saadaan muokkaimella määriteltyä sekä tallennettua RDF/XML-formaattiin. Tämän jälkeen ontologiasta tallennetaan ohjelman ymmärtämään formaattiin tieto ontologiamuuttujien parittumisesta käyttäjän määrittämiin datalähteisiin, minkä jälkeen muokkaimella voidaan generoida R-ohjelmistoon syötettävä komentosarja, jolla haluttu ontologia ja siihen liittyvä data saadaan ladattua R-istunnon muuttujiksi tilastollista analyysiä varten. Lukuunottamatta graafista ulkoasua sekä ontologian määrittelyä ja tallennusta, pääosa ohjelman toteutuksesta sisältyi harjoitteluuni CERN:issä.

Muokkaimen käyttö alkaa ontologian määrittelemisellä. Tässä työssä tarkastellaan CERN:in klusteridataa, joten nimeksi ontologialle annetaan *Cluster data*. Seuraava vaihe on määrittellä ontologian ulottuvuudet, mitat sekä näiden väliset riippuvuudet. Mittoja luodessa on määriteltävä myös mitan datatyyppi sekä summautuvuustyyppi kuten myös dimensioita luodessa on määriteltävä niiden temporaalisuus. Kuvassa 1 on tutkimamme ontologian määrittely OlapEditissä.

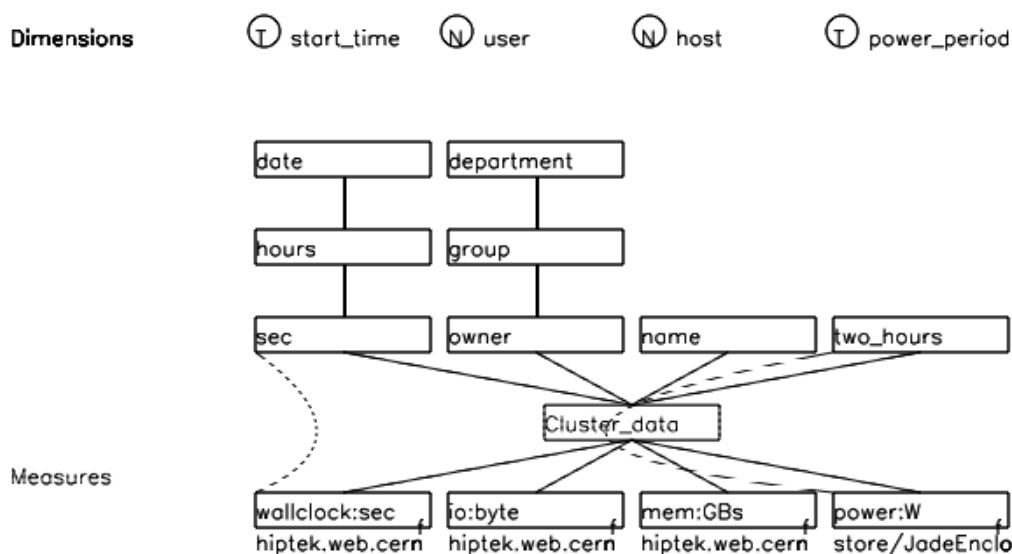
Ontologian tiedot tallentuvat muokkaimella luvussa 2.1 esiteltyyn RDF/XML-formaattiin tiedostoon palvelimelle muokkaimen jatkokäyttöä varten. Esimerkiksi tutkimamme ontologian RDF-tiedostosta poimitut laskentatyön kestoon viittavaan mitan (*wallclock*) määrittelyyn liittyvät osat näyttävät tiedostossa seuraavalaiselta:

```
<owl:Class rdf:about="#TemporalityType"/>
  <olapcore2:TemporalityType rdf:about="temporal"/>
  <olapcore2:TemporalityType rdf:about="non-temporal"/>
<owl:Class rdf:about="#Dimension"/>
  <olapcore2:Dimension rdf:about="#start_time">
    <olapcore2:hasDimensionTemporalityType rdf:resource="temporal"/>
    <olapcore2:hasDimensionLevel rdf:resource="sec"/>
    <olapcore2:hasDimensionLevel rdf:resource="hours"/>
    <olapcore2:hasDimensionLevel rdf:resource="date"/>
  </olapcore2:Dimension>
<olapcore2:MeasureSummarizabilityType rdf:about="float"/>
<olapcore2:MeasureSummarizabilityType rdf:about="stock"/>
<olapcore2:MeasureSummarizabilityType rdf:about="value-per-unit"/>
<owl:Class rdf:about="#MeasureUnit"/>
  <olapcore2:MeasureUnit rdf:about="sec"/>
<owl:Class rdf:about="#Measure"/>
  <olapcore2:Measure rdf:about="#wallclock">
    <olapcore2:hasMeasureUnit rdf:resource="sec"/>
    <olapcore2:hasMeasureSummarizabilityType rdf:resource="float"/>
    <olapcore2:hasDependencyOnDimension rdf:resource="start_time"/>
```

```

</olapcore2:Measure>
<owl:Class rdf:about="#hasDimensionLevel"/>
<olapcore2:DimensionLevel rdf:about="#sec">
<olapcore2:rollsUp rdf:resource="hours"/>
</olapcore2:DimensionLevel>
<olapcore2:DimensionLevel rdf:about="#hours">
<olapcore2:rollsUp rdf:resource="date"/>
</olapcore2:DimensionLevel>
<olapcore2:DimensionLevel rdf:about="#date">
</olapcore2:DimensionLevel>

```



Kuva 1: Klusteridataontologian määrittely OlapEditissä. Ylimpänä ovat ulottuvuuksien nimet temporaalisuustyyppineen (T=temporaalinen, N=eitemporaalinen) ja niiden alapuolella ulottuvuuksien tasot. Keskellä kuvaa on ontologian nimi, jonka alapuolella ontologian mitat mittayksikköineen, summautuvuustyyppineen (f=flow) sekä datalähdeosoitteineen.

Muokkaimella tehtävä datan kartoittaminen ontologian suhteen alkaa datalähteen tutkimisella muokkaimella. Ohjelma tukee myös tavallisen HTTP-autentikoinnin vaativaa datalähteen lukemista. Tämän jälkeen käyttäjän on suoritettava itse kartoitus. Muokkain hakee annetusta ulkoisesta tai palvelinkoneelle viittaavasta URL:sta CSV-tyyppisen (*comma-separated values*, pilkuilla erotellut arvot) datan muuttujanimet. Datalähteen CSV-tyyppisyyden käyttäjä voi muuttaa myös vastaavaan eri merkillä erotettuun rakenteeseen.

Lähteen muuttujien tunnistamisen jälkeen ohjelma tekee kyselyn, jossa käyttäjän tulee valita kullekin mittaan liittyvään muuttujaan vastinmuuttuja datalähteestä. Tämä vaihe suoritetaan jokaiselle mitalle, minkä jälkeen on ontologia-datalähteparille saatu muodostettua omanlaistaan `data_sources`-tiedostotyyppiä oleva kuvaus, jonka perusteella muokkaimella voidaan siirtyä viimeiseen vaiheeseen eli tiedon viemiseen tilastolaskentaohjelmistolle. Esimerkkinä `data_sources`-tiedoston rakenteesta jälleen *wallclock*-muuttujan kirjausrivi tiedostossa:

```
name="wallclock (sec)" source="http://hiptek.web.cern.ch/hiptek/
protected/accounting_feb2011_muok.txt"
authentication="username:password" delimiter=":"
variablename1="wallclock" variableget1="ru_wallclock"
variablename2="start_time:sec" variableget2="start_time"
variablename3="start_time:hours" variableget3="no_data_source"
variablename4="start_time:date" variableget4="no_data_source"
```

OlapEditin viimeinen toiminto generoi käyttäjälle R-tilastolaskentaohjelmistoon syötettävän komentosarjan, jolla ontologiaan määritellyt muuttujat saadaan halutuilla rakenteilla tallennettua istunnon muuttujiksi. Komentosarja muodostaa datalähteen ontologiamuuttujia vastaavista datavektoreista datakehyksiä (engl. *data frame*) sisältäviä ympäristöjä (engl. *environment*) siten, että ulottuvuudet talletetaan yhteen ympäristöön ja mitat toiseen. Ympäristöissä ulottuvuudet ovat datakehyksinä, joiden vektoreina ulottuvuuden eri tasot ovat, ja vastaavasti mitat mittadatan sisältävinä vektoreina. Esimerkiksi *wallclock*-muuttujalle OlapEdit generoi seuraavanlaisen R-komentosarjan ontologiainformaation tuomista varten:

```
dimensions=new.env(hash = TRUE, emptyenv())
measures=new.env(hash = TRUE, emptyenv())
require("RCurl")
data0<-read.table(textConnection(getURL("http://username:password@hiptek.
web.cern.ch/hiptek/protected/accounting_feb2011_muok.txt")), sep=":",
header=TRUE)
measures$wallclock <- data0[,c("ru_wallclock")]
dimensions$start_time$sec <- data0[,c("start_time")]
dimensions$start_time$hours <- 0
print("warning: data of dimensions$start_time$hours is empty. Use R to
map it")
dimensions$start_time$date <- 0
print("warning: data of dimensions$start_time$date is empty. Use R to map
it")
```

Yhteen ontologiaan liittyvät yhdentyyppiset tiedot tallentuvat muokkaimella aina samaan tiedostoon, jolloin kokonaisen ontologian lataaminen datoiin yksinkertaistuu. Muokkain myös tunnistaa samojen datalähteiden käytön eri mitoille, jolloin huolimatta usean mitan kartoittamisesta samasta ontologiasta datalähteitä

ei tarvitse ladata useaan kertaan. Tässä työhön liittyvän ontologian tiedostot ja niiden merkitykset ovat esitettyinä liitteessä A.

3.2 Klusteridatan analyysi

Tarkasteltuun ontologiaan on tässä työssä liitetty kaksi erityyppistä datalähdettä. Tämä vaikeuttaa datan yhdistämistä, sillä energiankulutusdata ei suoraan yhdisty laskentatyöoloihin, vaan laskentatöiden data on liitettävä energiankulutusdataa kohdentamalla laskentatöiden pilkottuja arvoja töiden aikaleimojen perusteella oikeisiin energiankulutusdatan määäämiin aikaväleihin. Esimerkiksi aikaleimansa perusteella puoliksi kahdelle kahden tunnin aikavälille jakautuneen laskentatyön muistirasitus voidaan kohdistaa puoliksi näille kahdelle aikavälille olettaen laskentatöiden olevan rasituksiltaan riittävän tarkasti tasaisia jaoteltavaksi tällä tavoin.

Datojen yhdistäminen kokonaisuudeksi suoritetaan R-ohjelmistolla. Accounting-datasta muodostetaan alku- ja loppuajoille taulukot, joissa jokaiselle ajankohdalle sijoitetaan kunkin laskentatyön tiedoista halutun suureen ajallisesti keskimääräinen arvo. Nämä taulukot yhdistetään yhdeksi datakehikseksi, johon lisätään vielä ajankohdiksi tasakaksituntiset. Datakehiksen rivit laitetaan ajankohdan perusteella suuruusjärjestykseen, minkä jälkeen lasketaan halutun suureen muutos kullakin ajankohdalla vähentämistä alkuaikataulukkoon lasketusta arvosta loppuaikataulukkoon laskettu arvo. Tämän jälkeen lasketaan muutosarvovektorin kumulatiivinen arvo alkaen pienimmästä ajanhetkestä, joka kerrotaan seuraavan ajankohdan alkuun kestäväällä ajalla oikean painotuksen aikaansaamiseksi. Luodaan uusi aikatietovektori, johon lasketaan alaspäin pyörivällä jakolaskulla kunkin ajankohdan kaksituntinen, johon se kuuluu, minkä jälkeen summataan kuhunkin kaksituntiseen kuuluvien ajanhetkien painottamalla saadut arvot. Näin on saatu energia-datan kanssa suoraan yhdistettävä datakehys.

Yhden laskentatyön tietojen jakaminen osiin ja toisaalta yhdistäminen muiden laskentöiden tietojen kanssa poistaa mahdollisuuden laskentatyödatan täysimääräisestä mielekkästä tarkastelusta energiankulutusdatan yhteydessä, sillä kahden tunnin aikaväleillä töitä suoritetaan useita. Täten esimerkiksi yhteen työhön liitettyä kategorisointia on oikeastaan mahdoton liittää energiankulutuksessa havaittaviin muutoksiin. Laskentatöistä voidaan kuitenkin tarkastella aikavälien sisäisiä keskiarvoja joidenkin muuttujien kuten lukumäärän, varatun muistin sekä I/O-liikenteen suhteen.

Aikaväleille voidaan kohdistaa täten kolme tekijää selittämään muutoksia energiankulutuksessa. Näiden selittävyyttä voidaan tutkia muodostamalla lineaarinen regressiomalli pienimmän neliösumman menetelmällä, jossa minimoidaan datapis-

Taulukko 1: Accounting-datojen perustietoja. Ensimmäinen data viittaa suurempaan accounting-dataan, toinen data viittaa energiadatan kanssa yhdistettävään lyhyemmän ajanjakson dataan. Muuttujanimien selitykset: *mem*=muistinkäyttö (GBs), *io*= I/O-liikenteen määrä (GB) ja *wallclock*= laskentatyön kesto (s).

	<i>mem</i>	<i>io</i>	<i>wallclock</i>
1. data			
Keskiarvo	62424	3,07	2292
Minimi	0	0	1
Maksimi	17179980670	21340	259202
2. data			
Keskiarvo	69942	0,936	953
Minimi	0	0	1
Maksimi	17179893861	758	259201

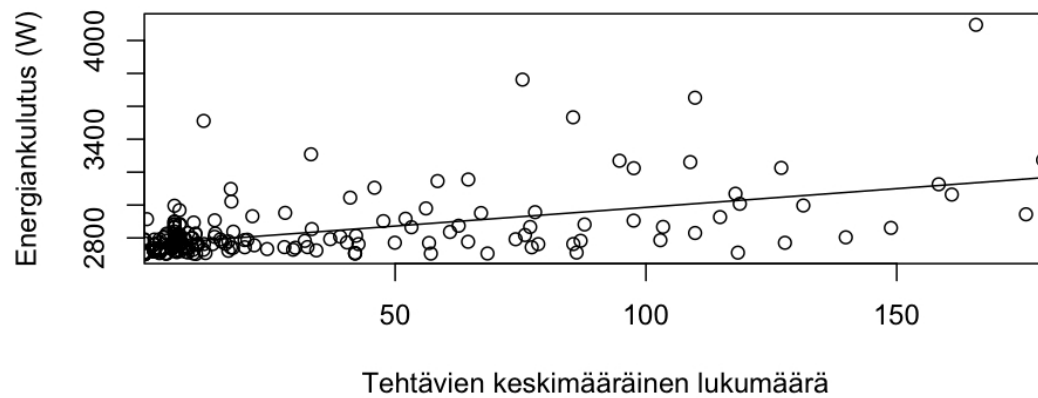
teiden ja regressiosuoran etäisyyksien neliöiden summa sopivilla regressiosuoran parametreilla. Regressiosuorasta voimme tehdä johtopäätöksiä laskentatöiden ominaisuuksien vaikutuksista energiankulutukseen, jonka perusteella voimme arvioida myös tutkimamme Jade-klusterin arkkitehtuurin optimaalisuutta energiankulutuksen suhteen.

Energiankulutuksen minimointi ei kuitenkaan välttämättä ole ainoa päämäärä, eikä minimin tavoittelu ole tarkoituksenmukaista erityisesti tilanteessa, jossa laskentatöiden suoritus on pitkällä aikavälillä epätasaista, jolloin minimi edellyttäisi suuria poikkeutuksia töiden läpikulkuajan suhteen suhteessa nykyhetkiseen klusteriin.

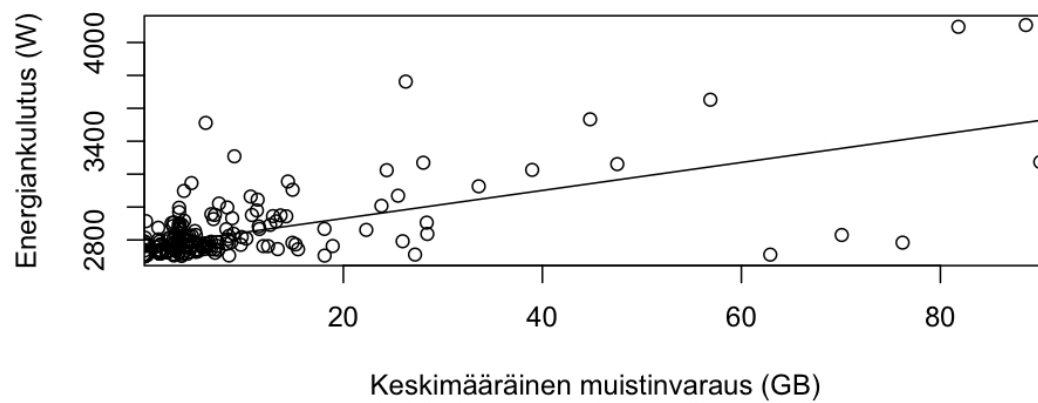
4 Tulokset

Taulukossa 1 on esitetty accounting-datojen perustietoja. Datat ovat *io*-muuttujan arvoja lukuunottamatta hyvin samanluontoisia. Datoja kuitenkin käsitellään eri tarkoituksiin, eikä niiden samankaltaisuudelle tarvitsekaan asettaa minkäänlaisia vaatimuksia. Ensimmäinen data on laajempi otos, minkä vuoksi sitä käytetään laskentaklusterin käytön luonteen analysoimiseen, kun taas vain toiseen dataan on liitettävissä energiankulutusdata, jonka avulla energiankulutuksen muutosta suhteessa datasta saataviin eri tekijöihin voidaan analysoida.

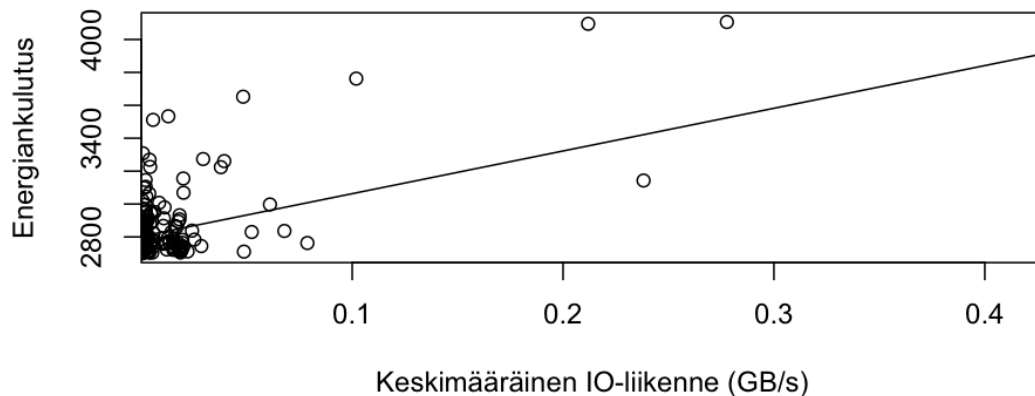
Kuvissa 2, 3 sekä 4 on esitetty laskentatöiden määrän, muistinvarauksen ja I/O-liikenteen lisääntymisen vaikutukset energiankulutuksen kasvamiseen. Suurennok-



Kuva 2: Tietokoneklusterin osajoukon energiankulutus käynnissäolevien laskutehtävien lukumäärän suhteen sekä pisteistöön sovitettu regressiosuora.



Kuva 3: Tietokoneklusterin osajoukon energiankulutus kahden tunnin ajanjakson keskimääräisen muistinvarauksen suhteen sekä pisteistöön sovitettu regressiosuora.



Kuva 4: Tietokoneklusterin osajoukon energiankulutus kahden tunnin ajanjakson keskimääräisen I/O-operaatioiden dataliikenteen suhteen sekä pisteistöön sovitettu regressiosuora.

set näiden kuvien tiheämmän pisteistön alueista ovat liitteessä B. Regressiosuorien osoittamien trendien ympärillä olevat suuret hajanaisuudet johtunevat ensinnäkin siitä, että mallit ovat vain yhden selittävän tekijän regressiomalleja, jolloin toiset tekijät eivät ole otettuna huomioon, mutta myös esimerkiksi mahdollisista laskentakoneiden uudelleenkäynnistymisistä sekä muista energiankulutukseen vaikuttavista tekijöistä.

Yhdistämällä tulokset saadaan kolmen selittävän tekijän regressiomalli pienimmän neliösumman menetelmällä. Mallin hyvyttä kuvaavat R^2 - ja korjattu R^2 -arvo ovat esitettyinä taulukossa 2 yhdessä regressiokertoimien nollihypoteesin (H_0 : Selittävän muuttujan X regressiokerroin $\beta_X = 0$) paikkansapitävyyden todennäköisyyttä kuvaavan p-arvon kanssa. Kolmen selittävän tekijän mallissa lukumäärätekijän p-arvo on yleisesti käytettyjen luottamusarvojen yläpuolella, minkä vuoksi alaspäin askeltamalla saadaan uusi regressiomalli, jossa selittävinä tekijöinä ovat keskimääräinen muistinvaraus ja keskimääräinen I/O-liikenne. Tällä tavoin saadussa mallissa molemmat selittävät tekijän ovat tilastollisesti merkitseviä, eivätkä R^2 -arvot juuri pienene suhteessa kolmen selittävän tekijän malliin.

Muutokset energiankulutuksessa eivät taulukon 2 regressiomallien R^2 -arvojen perusteella täysin selity klusterin accounting-datasta saatavilla selittäville tekijöillä. Voidaan kuitenkin jonkinasteisella varmuudella todeta riippuvuuden olemassaolo kuten intuition perusteellakin voitaisiin arvioida. Riippuvuus ei kuitenkaan ole vahva suhteessa mallien vakiotermeihin.

Taulukko 2: Energiankulutusta kuvaavien regressiomallien R^2 - ja korjatut R^2 -arvot sekä kunkin selittävän muuttujan regressiokertoimen p-arvo kussakin mallissa. Usean selittäjän mallista on siirrytty yksinkertaisempaan poistamalla selittävä tekijä, jolla on suurin p-arvo. Lyhenteet: lkm = laskentatöiden lukumäärä, mem = keskimääräinen muistinvaraus ja io = keskimääräinen I/O-liikenne

Selittävät muuttujat	R^2	korjattu R^2	$p(\beta_{lkm})$	$p(\beta_{mem})$	$p(\beta_{io})$
lkm, mem, io	0,4584	0,4511	0,063	$2,6 \cdot 10^{-7}$	$8,3 \cdot 10^{-4}$
mem, io	0,45	0,4451	-	$7,87 \cdot 10^{-15}$	$6,69 \cdot 10^{-4}$

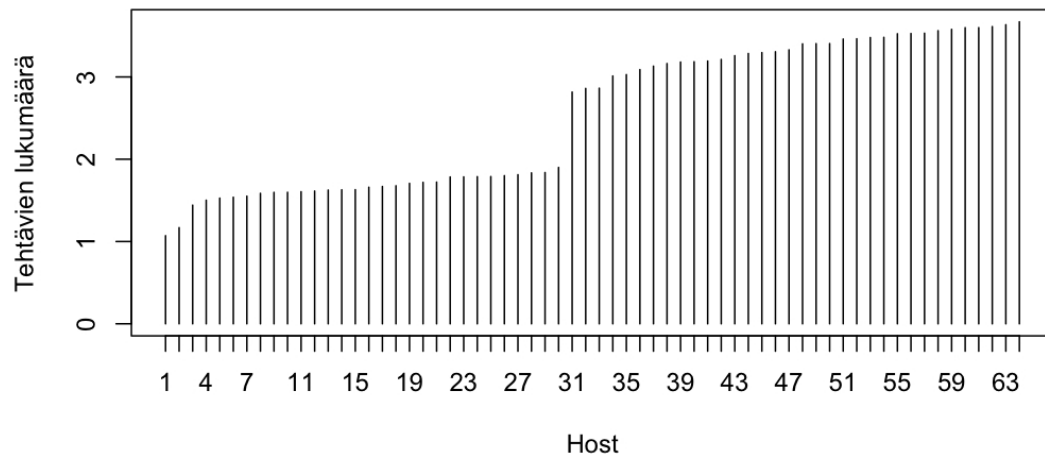
Mallien selitysasteen ja regressiokulmakertoimien suhteellisen pienuuden vuoksi huomattavampaa kuvissa onkin regressiosuorien vakiotermin suuruus. Suurin osa klusterin energiankulutuksesta on selvästikin vakioista ja syntyy jo klusterin pitämisestä valmiustilassa. Taulukon 2 toisessa regressiomallissa saadaan vakiotermin arvoksi noin 2760 Wattia. Koko energiankulutuksen minimiteho klusterille on noin 2700 Wattia, joka on koko ajanjakson energiankulutuksen keskiarvosta 95 % ja jopa energiankulutuksen maksimiarvosta 66 %.

Maksimiarvon suuruus suhteessa muihin arvoihin voi johtua jostain poikkeuksellisesta tekijästä kuten yhden tai useamman koneen uudelleenkäynnistämisestä, mikä vuoksi paremman kuvan lepotehon suhteellisesta suuruudesta antaa keskiarvosta keskihajonnan verran suurempi (pienin 91 % datamassasta) tehoarvo. Tästä arvosta datasta saatu klusterin minimiteho on 89 %. Koska klusterin lepoteho on näin suuri suhteessa kokonaistehoon, voidaan todeta, että energiankulutuksen optimoinnin kannalta olisi tärkeää koettaa rakentaa konemäärällisesti mahdollisimman pieni laskentaklusteri.

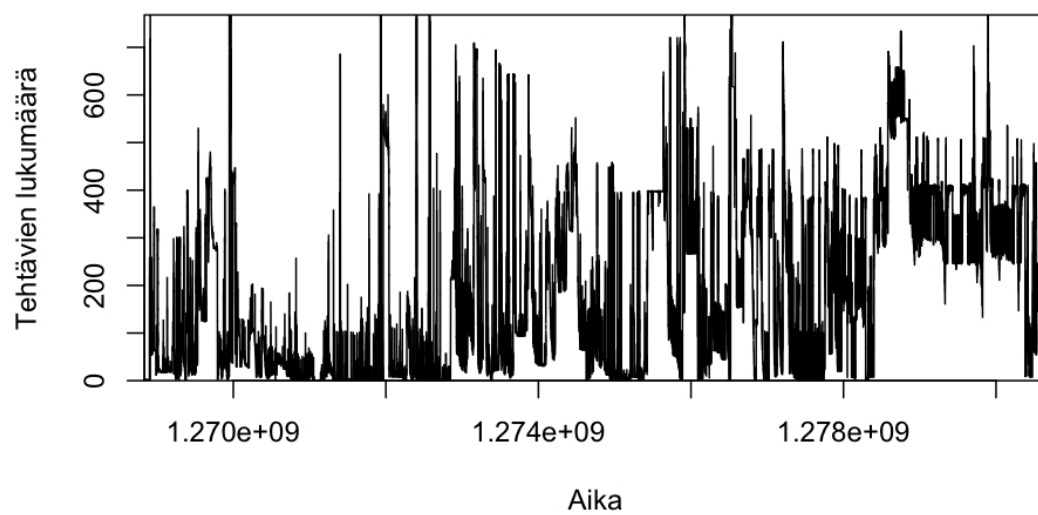
Kuvissa 5 ja 6 on kuvattu puolen vuoden ajalta olevan datan perusteella klusterin käyttöä. Kuvista huomataan, ettei Jade-klusterin käyttöaste ole kovinkaan lähellä sen kapasiteettia. Yksinkertaistamalla laskentatyöt täydellisesti jaettaviksi yksi työ prosessoria kohden, saadaan tarvittavaksi prosessorimääräksi yksinkertaisen kaavan

$$\frac{\text{töiden kestojen summa}}{\text{loppuaika} - \text{alkuaika}} = \frac{\Sigma(\text{wallclock})}{\max(\text{start_time} + \text{wallclock}) - \min(\text{start_time})}$$

perusteella 162 prosessoria, mikä tarkoittaa neljäätoista kappaletta Jade-klusterin koneiden mukaisia laskentakoneita. Klusterin konemäärää voitaisiin siis pienentään tämahetkisestä 64 koneesta noin 78 %, mikä tarkoittaa aiemmin saatujen tulosten perusteella noin 74 %:n energiansäästöä.



Kuva 5: Laskentatöiden keskimääräinen jakautuminen laskentakoneille. Koneita ei selvästikään käytetä tasaisesti, vaan jonohallintaohjelmisto suosii joitain koneita.



Kuva 6: Laskentatöiden lukumäärä koko ajalta. Klusterin kapasiteetti, 768 työtä, saavutetaan muutamassa kohdassa hetkittäisesti.

Laskentaklusterin päätarkoituksen vuoksi on kuitenkin myös huomioitava koneiden vähennyksen vaikutus laskentaoperaatioiden valmistumisaikaan. Laskentaoperaatiot on saatava suoritettua, ja käyttäjäkunnan mieltymyksistä riippuen voi olla suositeltavaa myös optimoida läpikulkuaikaa jonkinlaisella hyötyfunktiolla. Voidaan tutkia esimerkiksi normaalien työaikojen vaikutusta klusterin käyttöasteeseen, jolloin käyttöasteen tasauksen aiheuttama läpikulkuaajan kasvu olisi lyhyehköllä aikavälillä korjaantuva ongelma.

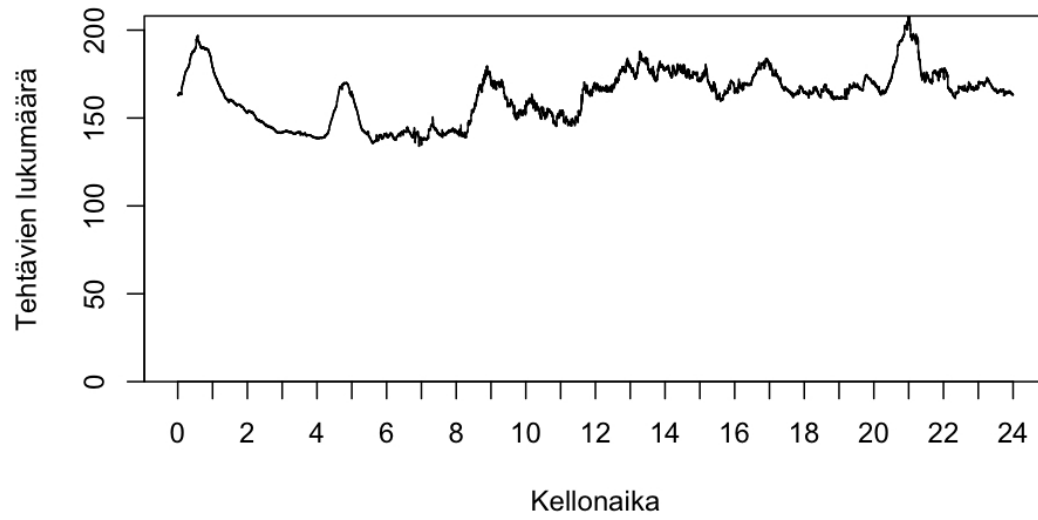
Kuvista 7 ja 8 nähdään laskentatöiden määrän jakautuminen kellonaikojen ja viikonpäivien suhteen. Ryhmien sisäisten varianssien keskinäistä yhtäsuuruutta testaavan Bartlettin testin nollahypoteesia (H_0 : Ryhmien sisäiset varianssit ovat keskenään yhtäsuuret) vastaaviksi p-arvoiksi saadaan $p_{klo}=0,509$ ja $p_{vkv}=0,998$, jotka viittaavat yleisillä merkitsevyytasoilla varianssianalyysin mahdollistamaan keskinäisten varianssien yhtäsuuruuteen. Varianssianalyysillä nollahypoteesia (H_0 : Ryhmien odotusarvot ovat keskenään yhtäsuuret) vastaaviksi p-arvoiksi saadaan $p_{klo}=0,484$ ja $p_{vkv}=0,338$, jotka viittaavat ryhmien homogeenisyyteen odotusarvon osalta.

Varianssianalyysin perusteella voidaan todeta, että esimerkiksi työaikojen aiheuttaman rytmin vaikutus laskentatöiden ajoaikoihin on merkityksetön. Ongelmat laskentaklusterin käytön tasaisuudesta ovat siis pidemmän aikavälin ilmiö, mikä osaltaan vähentää painetta tietokoneiden määrän pienentämiseen, erityisesti laskentatöiden läpikulkuaajan lyhyttä painottavan hyötyfunktion tapauksessa.

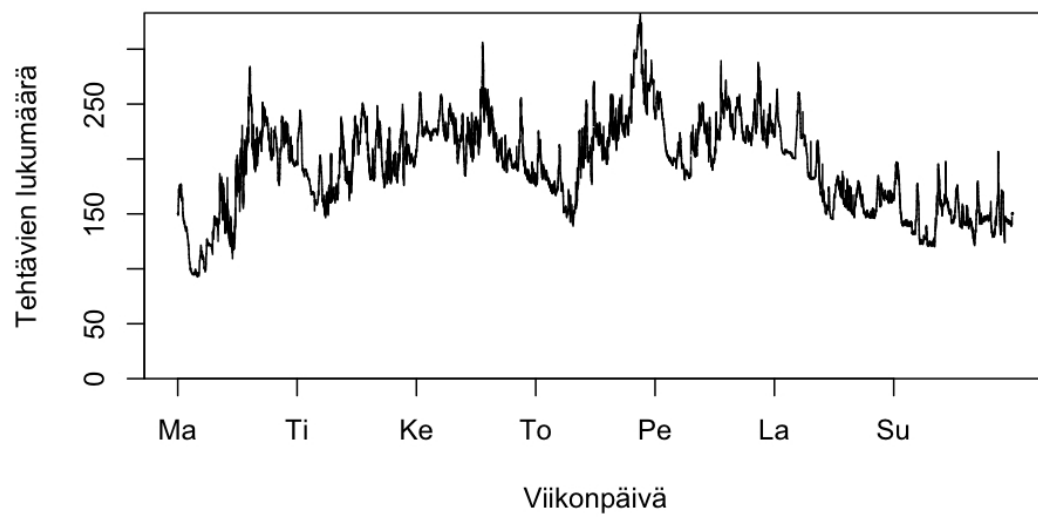
5 Tarkastelu

Datan hajanaisuuden ja heterogeenisyyden vuoksi datan hallinta ja integrointi ovat hankalia operaatioita tietokoneille. Asiakokonaisuuksien määrittely käsitteet ja niiden riippuvuuksia ja hierarkioita selittäviksi ontologioiksi mahdollistaa datakokonaisuuksien rakenteiden välittämisen tietokoneille ymmärrettävässä muodossa.

Työssä esitelty osana EXHADA-yhteisprojektia kehitetty ontologiamuokkain on työkalu ontologioiden määrittelyyn sekä ontologiatietueiden liittämiseen useisiin datalähteisiin. Ontologiatiedon muokkain tallentaa RDF/XML-muodossa, datalähteiden ja ontologian muuttujien kartoitetut vastaavuudet tallentuvat projektin omaan data_sources-tiedostomuotoon. Kartoitustiedosta muokkain osaa generoida R-tilastolaskentakielen koodia, jolla ontologian muuttujat saadaan datatietoi-
neen ladattua R-istuntoon käytettäväksi.



Kuva 7: Laskentatöiden jakautuminen kellonaikojen suhteen.



Kuva 8: Laskentatöiden jakautuminen viikontpäivien suhteen.

Tässä työssä ontologiaksi määriteltiin CERN:in laskentaklusteridata. Ontologiaan liitettiin kahdentyyppistä dataa, jotka aikaleimatyyppiensä poikkeavuuden vuoksi yhdistettiin toista dataa pakkaamalla kohdistetuksi toisen datan keskiarvotietoista dataa vastaavaksi. Yhdistetystä datasta muodostettiin regressiomalli, josta selviää laskentatöiden muistinvarauksen ja input/output-liikenteen tilastollisesti merkittävä vaikutus laskentaklusterin energiankulutukseen, vaikkakin mallin selitysaste on matalahko 0,45.

Muodostetusta mallista selviää kuitenkin laskentaklusterin lepotehon suuri osuus kokonaisenergiankulutuksesta. Koneiden käynnissäolon kuluttaman energian osuudeksi saatiin 95 % kokonaisenergiankulutuksesta. Näin suuresta osuudesta voidaan vetää johtopäätös energiankulutuksen minimin olevan saavutettavissa vähentämällä tietokoneiden määrä pienimpään tarvittavaan määrään. Potentiaalisesti pienennysmahdollisuudeksi saatiin tietokoneiden vähentäminen 78 prosentilla, mikä yhdessä lepotehosta saadun tiedon kanssa tarkoittaisi 74 prosentin energiansäästöä.

Viitteet

- [1] Becket, Dave (toim.): *RDF/XML Syntax Specification*. <http://www.w3.org/TR/REC-rdf-syntax/>, 2004.
- [2] *Disease Ontology*. <http://disease-ontology.org>, viitattu 31.7.2012.
- [3] Guarino, Nicola: *Formal Ontology, Conceptual Analysis and Knowledge Representation*. *International Journal of Human-Computer Studies*, 43(5-6):625–640, 1995.
- [4] Halevy, Alon, Anand Rajaraman ja Joann Ordille: *Data Integration: The Teenage Years*. Teoksessa *IN VLDB*, sivut 9–16, 2006.
- [5] Hendler, James, Tim Berners-Lee ja Eric Miller: *Integrating Applications on the Semantic Web*. *Journal of the Institute of Electrical Engineers of Japan*, 122:676–680, 2002.
- [6] *Resource Details for jade-cms.hip.fi*. <http://www.nordugrid.org/monitor/clusdes.php?host=jade-cms.hip.fi&port=2135>.
- [7] *Storage Element Performance Optimization for CMS Analysis Jobs*. <http://indico.cern.ch/getFile.py/access?contribId=243&sessionId=8&resId=0&materialId=poster&confId=149557>.
- [8] Miller, Eric: *An Introduction to the Resource Description Framework*. *Bulletin of the American Society for Information Science and Technology*, 25:15–19, 1998.
- [9] Niemi, Tapio, Santtu Toivonen, Marko Niinimäki ja Jyrki Nummenmaa: *Ontologies with Semantic Web/Grid in Data Integration for OLAP*. *International Journal on Semantic & Web Information Systems*, 3:25–49, 2007.
- [10] Niinimäki, Marko ja Tapio Niemi: *An ETL Process for OLAP Using RDF/OWL Ontologies*. *Journal of Data Semantics, JoDS XIII: Special Issue “Semantic Data Warehouses”*, sivut 97–119, 2009.
- [11] Niinimäki, Marko, Tapio Niemi, Stephen Martin, Jyrki Nummenmaa ja Peter Thanisch: *Timely Report Production from WWW Data Sources*. Teoksessa *BIR 2011 Work-shops, LNBIP 106*, sivut 184–195. Springer Berlin Heidelberg, 2012.
- [12] Shoshani, Arie: *Summarizability*. Teoksessa *Encyclopedia of Database Systems*, sivut 2880–2884. Springer US, 2009.

- [13] *Ubuntu Manpage: Accounting - Grid Engine Accounting File Format.* http://manpages.ubuntu.com/manpages/intrepid/man5/sge_accounting.5.html, viitattu 1.8.2012.

A Ontologialinkit

Työssä käytettyyn ontologiaan *Cluster_data* liittyvät linkit löytyvät HIP:n palvelimelta osoitteessa <http://portal.hip.fi/olapedit/store/index.html> listattuina.

Ontology-linkki ohjaa OLAPedit-näkymään, jossa *Cluster_data* on muokattavana ontologiana. Kaikki ontologiamuokkaimen serverikoneella ajettavia komentosarjoja vaativat toiminnot vaativat kuitenkin autentikoinnin.

RDF-linkki näyttää ontologian määrittelyn RDF/XML-muodossa. Tätä tiedostoa ontologiamuokkain käyttää ontologiatiedon säilyttämiseen ja tallentamiseen.

Data source -linkki näyttää ontologian *data_sources*-tiedoston, joka sisältää ontologiaan liitettävän kartoitustiedon. Yhdellä rivillä on kaikki yhteen mittaan liittyvä kartoitustieto: mitan nimi, mittayksikkö, datalähde, muuttujanimet ja niiden vastaavuudet datalähteessä sekä mahdolliset autentikointiin vaadittavat kirjautumistiedot.

R-loader-linkki avaa R-tiedoston, jonka sisällön suorittamalla voi kartoitetut tiedot ladata R-session muuttujiksi. Komentosarjaa voi käyttää myös osana R-skriptiä datankeruuosuutena. Komentosarjan alussa on myös määritelty muutama funktio, joilla ladattujen tietojen käsittely yksinkertaistuu.

B Suurennoskuvia analyysin kuvaajista

