Määrittelydokumentti

Makromolekyylikompleksien rekonstruktioprosessin
rinnakkaistaminen Beowulf-ympäristössä

Antti Perälä
Ilkka Koskenniemi
Matti Kääriäinen
Peter Lamberg
Taneli Mielikäinen
Valtteri Rahkonen

Helsinki 23. helmikuuta 2000

Ohjelmistotuotantoprojekti: Määrittelydokumentti
Ohjaaja: Satu Sihvo

Asiakkaat:
Janne Ravantti
professori Dennis Bamford

Vastuuhenkilö: Juha Taina

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Sisältö

Kuvat

1 Johdanto

Virukset ovat pieniä (n. 150 – 3000 Ångströmiä) organismeja, joilla on oma DNA:sta tai RNA:sta muodostuva perimä. Virukset aiheuttavat monia sairauksia, joten niiden tutkiminen on lääketieteellisesti tärkeää. Ne ovat yksinkertaisuutensa vuoksi myös tärkeitä mallijärjestelmiä biologisessa perustutkimuksessa. Jo yli viidenkymmenen vuoden ajan elektronimikroskopia on näytellyt merkittävää roolia virusten rakenteen tutkimuksessa. 1970-luvun alusta lähtien viruksista on laadittu kolmiulotteisia malleja, jotka ovat mahdollistaneet virusten rakenteen kvantitatiivisen tutkimuksen ja mm. täsmälääkkeiden kehityksen.

Tiheysmallien laatimisessa tärkeimpiä kuvanmuodostustekniikoita ovat röntgenkristallografia ja kryoelektronimikroskopia, joista ensimmäisen tarkkuus on 2–3 ja toisen 20–50 Ångströmin luokkaa [Frank96, Frank92].

Röntgenkristallografiassa partikkelit kiteytetään hilaksi, jonka diffraktiokuvioista voidaan rekonstruoida kolmiulotteinen malli. Aina partikkelia ei kuitenkaan osata kiteyttää. Tällöin kuvaaminen täytyy suorittaa elektronimikroskoopilla.

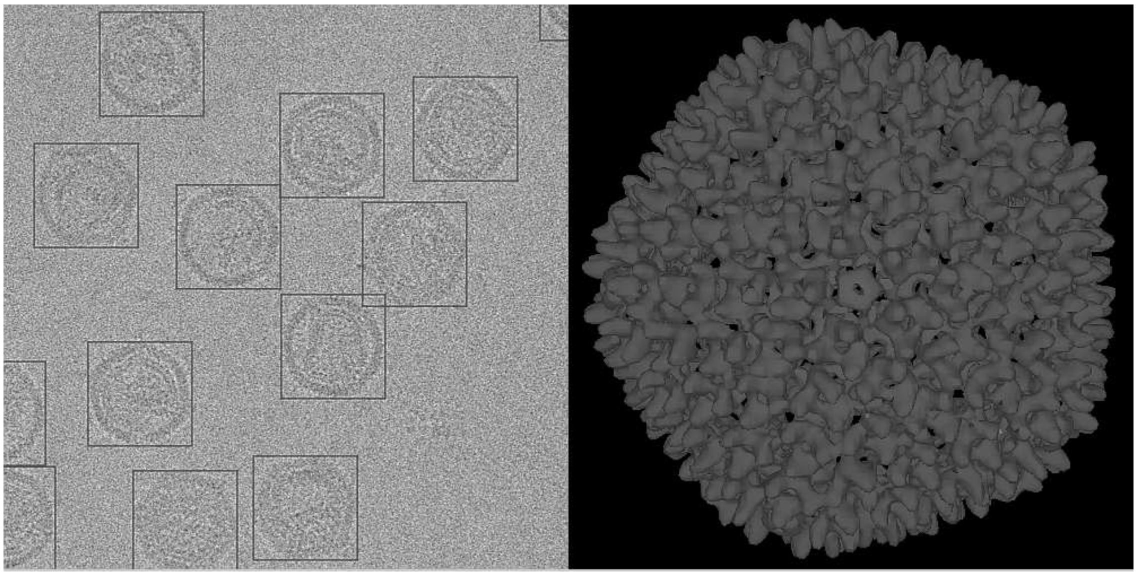
Kryoelektronimikroskopiassa kuvattavat partikkelit ovat vedessä, joka jäädytetään hyvin nopeasti, ettei veteen ehtisi muodostumaan jään kiderakennetta. Näin saatuun näytteeseen kohdistetaan elektronisuihku, joka muodostaa filmille[1] läpivalaisukuvan. Läpi päässeen säteilyn intensiteetin heikkeneminen on likimäärin suoraan verrannollinen kyseisellä kohdalla olevaan massaan.

Joukosta kaksiulotteisia läpivalaisukuvia (identtisistä kappaleista) voidaan periaatteessa muodostaa kolmiulotteinen malli, joka on asymptoottisesti identtinen alkuperäisen kappaleen kanssa. Tämä perustuu Johann Radonin 1917 esittelemään tulokseen, jonka mukaan täydellisestä joukosta d-ulotteisen kappaleen $d-1$-ulotteisia projektioita (läpivalaisukuvia) voidaan päätellä yksikäsitteisesti alkuperäinen d-ulotteinen kappale[2] [Gard95]. Teoreemaa on sovellettu laajasti tomografiassa, jossa tilanne on sama kuin virusmallinnuksessa lukuunottamatta sitä, että tomografiassa kaikkien läpivalaisukuvien orientaatiot tunnetaan [Herman99].

Virusmallinnuksen eräs suurimmista haasteista onkin kaksiulotteisten läpivalaisuku-

[1] Nykyisin kuvanmuodostuksessa käytetään myös CCD-kennoja, mutta ne eivät kuitenkaan vielä tarjoa yhtä hyvää kuvanlaatua kuin filmi.

[2] Tuloksen matemaattinen muotoilu johtaa ns. Radon-muunnokseen [Toft96].



Kuva 1: Vasemmalla pala PRD1-viruksen läpivalaisukuvia sisältävästä kryoelektronimikroskooppikuvasta. Oikealla saman viruksen tiheyskartan visualisointi tasaarvopinnan avulla. Elektronimikroskooppikuva ja tiheyskartta S. Butcher ja EMBL

vien orientaatioiden määrittäminen kolmiulotteisessa avaruudessa[3]. Ongelman ratkaisut voidaan jakaa kolmeen osaan: mallipohjainen, läpivalaisukuvien keskinäisten orientaatioiden mukainen ja symmetriaan perustuva orientointi. Menetelmiä sovelletaan eniten reaali- ja Fourier-avaruuksissa.

Mallipohjaisessa etsinnässä kolmiulotteista mallia läpivalaistaan eri suunnista ja näin saatuja mallin kaksiulotteisia projektioita verrataan elektronimikroskoopilla tuotettuihin läpivalaisukuviin.

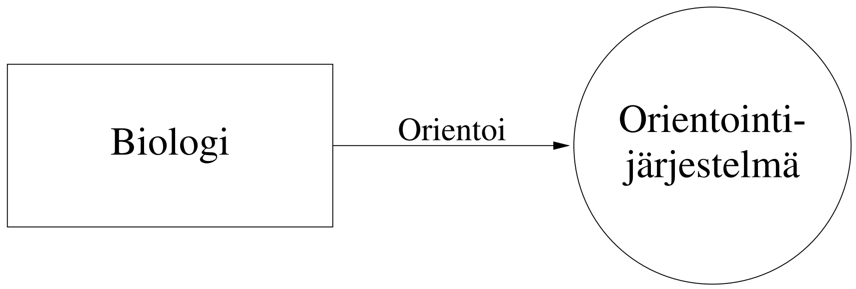
Läpivalaisukuvien keskinäisessä vertailussa kuvien väliltä pyritään löytämään leikkaussuoria[4], joiden avulla kuvien keskinäiset orientaatiot kolmiulotteisessa avaruudessa voidaan yksikäsitteisesti määrittää.

Symmetristen partikkelien tapauksessa läpivalaisukuvan kanssa identtisiä projektioita on symmetrian mukainen määrä. Orientaatio suhteessa symmetria-akseleihin voidaan tällöin määrittää etsimällä kuvasta leikkaussuorat itsensä kanssa.

Ohjelmistotuotantoprojektimme keskittyy mallipohjaiseen orientointiin ja projekt

[3] Toinen perinteisestä tomografiasta puuttuva osa-alue on virusten etsintä elektronimikroskooppikuvista [Kivioja99].

[4] Reaaliavaruudessa leikkaussuoria vastaavat kaksiulotteisen kuvan yksiulotteiset projektiot (eli kaksiulotteisesta kuvasta lasketaan Radon-muunnos) ja Fourier-avaruudessa tarkastellaan pyörähdyskeskipisteen kautta kulkevia suoria.



Kuva 2: Sidosryhmäkaavio

in on tarkoitus tuottaa orientointijärjestelmä, josta enemmän seuraavissa luvuissa.

2 Sidosryhmät

Orientointijärjestelmän käyttäjät ovat virusten rekonstruktioprosessin parissa työskenteleviä biologeja ja tietojenkäsittelytieteilijöitä. Koska järjestelmä tarjoaa käyttäjilleen vain yhden palvelun, kaikki järjestelmän käyttäjät kuuluvat samaan sidosryhmään. Orientointijärjestelmän ja sidosryhmän suhde on esitetty kuvassa 2.

3 Järjestelmän kuvaus

3.1 Tietojärjestelmä

Orientointijärjestelmä saa syötteenä kaksi tietorakennetta, makromolekyylin kolmiulotteisen mallin ja elektronimikroskooppikuvan. Näistä järjestelmä päättelee kaksiulotteisen kuvan orientaation eli mistä kulmasta kuva on otettu.

3.1.1 Makromolekyylin kolmiulotteinen malli

Makromolekyylikompleksien rekonstruktioprosessin tuloksena saatu kolmiulotteinen malli viruksesta. Malli on MRC-formaatissa oleva kolmiulotteinen taulukko, josta lasketaan kaksiulotteisia projektioita eri kulmista.

3.1.2 Elektronimikroskooppikuva

Elektronimikroskoopilla viruksesta otettu kaksiuloitteinen kuva. Kuvat syötetään järjestelmälle kaksiulotteisina taulukoina.

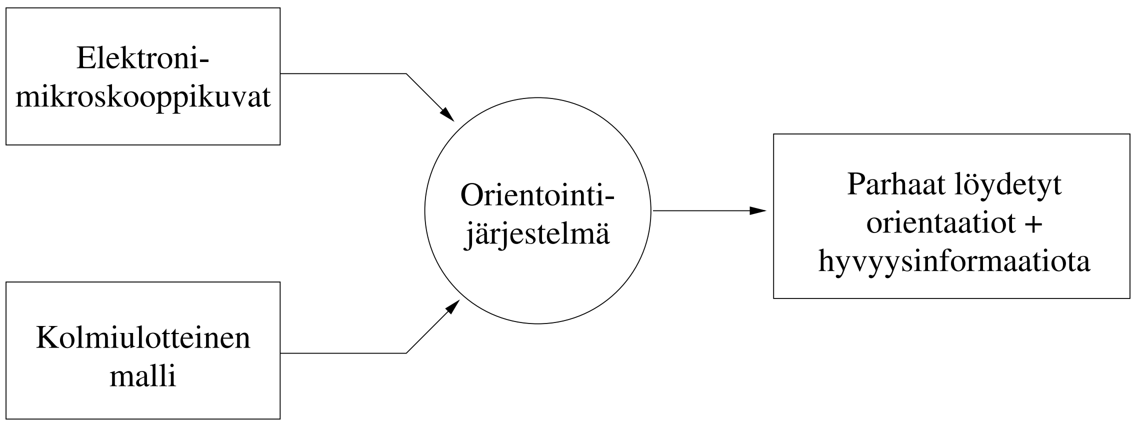
3.1.3 Lasketut orientaatiot

Orientointijärjestelmä tulostaa kutakin syötteenä saamaansa elektronimikroskooppikuvaa kohden listan kulmista, joissa kuva parhaiten sopii annettuun kolmiulotteiseen malliin. Tulostetut kulmat varustetaan informaatiolla, joka kuvaa sitä kuinka hyvin kuva kyseisessä kulmassa sopii annettuun malliin.

3.2 Toimintokuvaus

Orientointijärjestelmän tehtävänä on selvittää syötteenään saamien elektronimikroskooppikuvien orientaatio annettuun kolmiulotteiseen malliin nähden. Kuvien orientaatio selvitetään laskemalla mallista kaksiulotteisia projektioita ja vertaamalla kuvia niihin. Koska kuvissa esiintyy huomattavan paljon kohinaa ja malli ei välttämättä vastaa kovinkaan tarkoin kuvassa olevaa objektia, ei voida olettaa että mikään mallista laskettu kaksiulotteinen projektio sopisi läheskään täydellisesti elektronimikroskooppikuvaan. Tästä syystä järjestelmä tulostaa (epäselvissä tilanteissa) kutakin kuvaa kohden useita orientaatiovaihtoehtoja sekä hyvyysinformaatiota, joka kuvaa sitä kuinka hyvin kuva missäkin orientaatiossa sopii malliin. Tämä on hyvin raskas prosessi, joten se tullaan hajauttamaan luvussa 4 esiteltävään ympäristöön.

Orientointiin käytettävistä algoritmeista päätetään tarkemmin vasta suunnitteluvaiheessa. Tämän dokumentin luvussa **??** on kuitenkin oletettu, että orientointiin käytetään algoritmia, joka ensin laskee kolmiulotteisesta mallista projektioita astein välein (kahden akselin ympäri kierrettynä) ja vertaa sitten saatuja projektioita elektronimikroskooppikuviin (kiertäen vertailun yhteydessä projektioita asteen välein tasossa).



Kuva 3: Tietovuokaavio

4 Toteutusympäristö ja sen rajoitukset

4.1 Toteutusympäristö

4.1.1 Laitteistoympäristö

Orientointijärjestelmä toteutetaan siten, että se toimii Beowulf-klusterissa. Beowulf-klusteri poikkeaa rakenteeltaan sekä perinteisistä supertietokoneista, että mikrotietokoneista. Beowulf-klusterissa on kahdesta jopa useaan sataan mikroprosessoria. Sana Beowulf ei ole tietokoneklustereista puhuttaessa tavaramerkki eikä mikään tietty tuote. Beowulf on yleisnimitys standardiosista kootuille usean prosessorin koneille, joiden vahvuus perustuu yksittäisten prosessorien laskentatehoon ja laskennan sopivaan hajautukseen hienostuneiden ja räätälöityjen tiedonsiirtoarkkitehtuurien sijasta [Beowulf].

Supertietokoneet vs. Beowulf Supertietokoneissa käytetään usein erityisiä kyseistä järjestelmää varten suunniteltuja osia. Beowulf-klusterissa prosessorit ja muut osat ovat tavallisia massamarkkinoille suunniteltuja tuotteita. Tämän vuoksi Beowulf-klusteria sanotaan joskus köyhän miehen supertietokoneeksi. Beowulf-klusterin hajautusperiaate on SPMD tai MIMD. SPMD (Single program, multiple data) tarkoittaa: yksi ohjelma ja eri data jokaisella ohjelmalla. MIMD (multiple instructions, multiple data) tarkoittaa: eri ohjelmat ja eri data jokaisella ohjelmalla. Mikä tahansa MIMD voidaan tehdä SPMD:ksi.

Beowulf-saaga Beowulf-klusterit ovat saaneet nimensä Ajanlaskun alun ajoilta perityvästä Germaanisesta tai Kelttiläsestä runoelmasta, jossa kuningas Beowulf lähtee pelastamaan naapurimaataan Grendel hirviöltä, jota kukaan muu ei kykene voittamaan. Beowulf tappaa hirviön ja palaa kotiinsa sankarina. Myöhemmin Beowulf palaa ottelemaan suuren lohikäärmeen kanssa. Beowulf onnistuu tappamaan lohikäärmeen, mutta kuolee taistelussa saamiinsa vammoihin [BeowulfSaga].

Tiedonsiirto Massamarkkinoille suunniteltu usemman prosessorin väliseen kommunikointiin tarkoitettu tuote tarkoittaa käytännössä lähiverkkoa. Lähiverkkojen kapasiteetit ovat huomattavasti pienempiä kuin mikrotietokoneiden ja supertietokoneiden sisäisistä tiedonsiirtonopeuksista. Tämä on yksi Beowulf-klusterin luonteenomaisista piirteistä ja vaikuttaa olennaisesti siihen, mitä klusterilla voidaan tehdä.

Tiedonsiirtokapasitetti Paras hintalaatusuhde eri lähiverkkoratkaisuissa on kierrettyyn pariin perustuvalla Ethernet 100mb/s lähiverkolla. Tämä on yleinen ratkaisu Beowulf-klustereissa. Muita vaihtoehtoja ovat mm. FDDI-(Fiber DistributeD Interface) tai ATM-tekniikat, joilla päästää huomattavasti parempiin siirtonopeuksiin [ATM].

Nykyiset kotitietokoneet siirtävät tietoa keskusmuistin ja prosessorin välillä hetkittäisellä huippunopeudella 800 megatavua sekunnissa. Kun tätä verrataan klusterin eri prosessorien väliseen tiedonsiirtonopeuteen nähdään, että ero prosessorien välisessä kommunikoinnissa ja prosessorin ja sen oman muistin välisessä kommunikoinnissa on joka tapauksessa vähintään kymmenkertainen.

Verkkoarkkitehtuuri Perinteisesti Beowulf-klustereissa on käytetty väylätyyppistä arkkitehtuuria. Ainakin yksi tutkimus kuitenkin osoittaa, että monimutkaisemmat arkkitehtuurit voivat tehostaa laskentaa, toisaalta väylä on arkkitehtuureista kaikkein luotettavin ja helpoin toteuttaa. Monimutkaisempien rakenteiden hyödyntäminen tehokkaasti vaatii myös enemmän ohjelmistojen kehittäjiltä. Sama tutkimus osoittaa myös, että ohjelmallisesti tehtävä reititys eri prosessoreiden välillä ei ole merkittävästi hitaampi ratkaisu kuin kalliimmat laitteistoppohjaiset ratkaisut [Resch96].

Prosessorin ominaisuudet Yksittäinen prosessori on ominaisuuksiltaan verrattavissa tavalliseen työasemakoneeseen. Kullakin prosessorilla on suurehko määrä keskusmuistia pelkästään omassa käytössään ja yleensä myös satoja megatavuja tilaa hitaammassa magneettisesssa tietovarastossa.

Prosessorin ja syöttö- ja tulostelaitteet Beowulf-klusterin solmuihin ei yleensä ole liitetty tavalliseen työasemakoneeseen kuuluvia syöttölaitteita, kuten näppäimistöä tai hiirtä. Solmuissa on yleensä kuitenkin levykeasema tai muu vastaava, jota käytetään poikkeustilanteissa, kun solmuun ei saada muuten yhtettä. Solmuihin on vain harvoin liitetty kehittyneitä tulostelaitteita kuten näyttöjä tai tulostimia. Solmuun saattaa olla liitettynä joitakin merkkivaloja jotka ilmaisevat solmun teknistä tilaa.

Prosessori ja pysyväismuisti Koska kaupan hyllyltä saatavien kovalevyjen kapasiteetit monikertaistuvat jatkuvasti ja vanhan kovalevyn hinta Beowulf-klusterin yhden prosessorin muihin osiin nähden on vähäinen, on mahdollista olettaa, että jokaisella prosessorilla on kovalevy omassa käytössään.

Yksittäisen prosessorin laskentateho Yksittäisen prosessorin laskentateho on luokkaa 100MFlops (miljoonaa liukulukulaskutoimitusta sekunnissa) riippuen käytettävän prosessorin tyypistä ja sen sopivuudesta liukulukulaskentaan [Terra].

Klusterin laskentateho Klusterin laskentateho on huomattavasti monimutkaisempi asia kuin kaikkien prosessorien laskentatehot laskettuna yhteen. Vain harvoissa sovelluksissa koko järjestelmän laskentateho kasvaa suorassa suhteessa prosessorien koko laskentatehoon. Optimaalisesta Beowulf-klusterin hyötykäytöstä on esimerkkinä Portland Group-nimisen yrityksen yrityksen omalla Fortran kääntäjällä toteutetut kokeilut joissa he raportoivat 90% hyötykäyttösuhteen Beowulf-klusterin kaikilla koneilla. Tässä tapauksessa malli on hyvin määritelty ja algoritmi soveltuu ilmeisen hyvin rinnakkaistamiseen [Terra].

4.1.2 Ohjelmistoympäristö

Prosessorille asennettu ohjelmisto Beowulf-klusterin ylläpidon helpottamiseksi jokaisen koneen kovalevyllä on myös usein valmiiksi asennettuna ainakin käyttöjärjestelmä ja useita apuohjelmia, kuten kääntäjiä ja editoreita. Toinen vaihtoehto on, että jokaisella prosessorilla on vain sen verran pysyvää muistia kuin tarvitaan koneen käynnistämiseen. Tässä tapauksessa sovellukset ja mahdollisesti käyttöjärjestelmä ladataan, joko kokonaan tai osittain verkon yli jonkun toisen tietokoneen pysväismuistista. Prosessorin käynnistämiseen riittää yksi levyke tai prosessorin yhteyteen asennettu ROM-siru.

Käyttöjärjestelmä Käyttöjärjestelmä on usein Linux sen edullisuuden, tehokkuuden ja muokattavuuden ansiosta. Koska monet kaupalliset käyttöjärjestelmät ja ohjelmistot lisensoidaan prosessoria kohti, vapaan ohjelmiston käyttö on usein ainoa taloudellisesti järkevä vaihtoehto. Myös kehitys-Beowulf-klusterissa (4.1.3), käytetään Linux käyttöjärjestelmää [Linux].

Kehitystyökalut Kehitystyökalut pyritään valitsemaan siten, että ne ovat mahdollisimman riippumattomia eri alustojen eroista. Tämä lähestymistapa sulkee pois monia erikoistuneita tai kehitysvaiheessa olevia ratkaisuja.

MPI-kirjasto MPI-kirjasto valittiin hajautuksen sovellusrajapinnaksi, koska se on standardoitu ja siitä on saatavilla useita toimivia toteutuksia. MPI-kirjasto tarjoaa korkean tason palveluita, joita käyttämällä ohjelmasta tehdään skaalautuva ja riippumaton Beowulf-ympäristön rakenteesta [MPI].

C-kieli C-kieli on johdonmukainen valinta järjestelmän toteutukseen, sillä hyvin suunnitellut C-ohjelmat voidaan tunnetusti siirtää hyvin moninaisille laite alustoille ja toisaalta MPI-kirjastossa on valmiiksi ohjelmointirajapinta C kielelle.

Latex Ohjelmisto dokumentoidaan Latex-tekstinasettelukielellä. Latex kielen kääntämiseksi eri muotoihin on olemassa vapaita työkaluja, jotka toimivat miltei kaikissa ympäristöissä. Elektroninen julkaisu voidaan toteuttaa työkalun avulla, joka kääntää Latex kielen HTML-muotoon.

C++-kieli hylättiin C++ kielen käyttö ohjelmiston toteuskielenä hylättiin , koska toistaiseksi sen hienostuneita ominaisuuksia ei ole riittävässä määrin toteutettu eri ympäristöissä. C++ olisi vahvimmillaan joustavan ja nopean luokkahierarkian kehittämisessä. Toisaalta orientointijärjestelmän toteutuksellisten ongelmien ratkaisut vaativat kuitenkin algoritmien ja hajautuksen onnistunutta suunnittelua.

4.1.3 Kehitysklusteri

Ohjelmistotuotantoryhmä Makromolekyylikompleksien rekonstruktionprosessin rinnakkaistaminen Beowulf-ympäristössä (lyhemmmin Beklu) käyttöön on annettu yksitoistaprosessorinen Beowulf klusteri. Klusterin solmut ovat uusia Athlon tyyppisiä prosessoreita. ja klusterin tietoliikennettä hoitaa 100mb/s Ethernet verkko. Kullakin solmulla on 128 megatavua muistia ja noin 10 gigatavua vapaata kovalevytilaa. Yksi klusterin solmuista on niinsanottu isäntäsolmu.

Kehitysklusterin kehitystyökalut Kehitysklusterin solmuille on asennettu suhteellisen uusi GNU projektin C-kääntäjä, debuggeri ja muut tarpeelliset välineet. Myös vapaa toteutus MPI-kirjastosta on käytettävissä.

Käyttöjärjestelmä Klusterin jokaiselle solmulle on asennettu Mandrake käyttöjärjestelmä, joka perustuu linux käyttöjärjestelmäytimelle ja tarjoaa laajan valikoiman vapaata ohjelmistoa. Koska klusterin jokaisella solmulla on käytössä melko suuri määrä kiintolevytilaa, on järkevää, että jokaisella solmulla on monipuolinen valikoima sovelluksia.

4.2 Toteutusympäristön rajoitukset

4.2.1 Alustariippumattomuus

Jos ohjelma toteutetaan mahdollisimman puhtaasti käyttäen vain C-kielen ja MPI-kirjaston ominaisuuksia, on odotettavaa, että ohjelman siirtäminen toisille alustoille on mahdollista tehdä ilman laaja uudelleentoteuttamista. Koska kyseessä on tieteellinen sovellus, käytettävien tietotyyppien arvoalueisiin ja ehkä jopa niiden ominaisuuksiin, kuten pyöristyssääntöihin, on otettava kantaa kutakin alustaa kohti erikseen. Tämä on hyvä ottaa huomioon toteutuksessa siten, että käytettävien tietotyyppien vaihtaminen olisi mahdollisimman helppoa.

4.2.2 Laskentateho

Beowulf-klusterin laskentatehoa on vaikea määrittää täsmällisesti. Yläraja laskentateholle saadaan laskemalla yhteen kaikkien klusterin prosessoreiden erilliset laskentatehot. Orientointijärjestelmän kannalta laskentatehoa rajoittava tekijä on itse järjestelmän suunnittelu, sillä algoritmien valinta ja hajautus ratkaisee sen lisääkö uusien prosessoreiden liittäminen klusteriin orientointijärjestelmän laskentatehoa mielekkäässä suhteessa. Toisaalta satojenkin prosessoreiden Beowulf-klustereita on kokeiltu ja mitään ylärajaa ei ole näköpiirissä. Eli rahalla saa Beowulf-klusteriin lisää tehoa, mutta raha ei korvaa huonosti suunniteltua ohjelmaa.

Liukuluvut vs. kokonaisluvut Koska Beowulf-klusterit rakennetaan usein samoista osista kuin tavalliset mikrotietokoneet, Beowulf-klusteri saattaa kärsiä siitä, että useimmissa prosessoreissa liukulukujen laskenta on prosessorissa erillään kokonaislukujen laskennasta. Monet yleiset prosessoriperheet ovat perinteisesti keskittyneet kokonaislukulaskentaan ja liukulukulaskenta on laahannut suorituskyvyn osalta perässä. Tavalliset sovellusohjelmat vain vähän liukulukulaskentaa, mutta toisaalta tieteellinen laskenta suoritetaan usein likulukuoperaatioilla. Tämä on hyvä ottaa huomioon toteutusta suunniteltaessa.

4.2.3 Pysyväismuistin rajoitukset

Klusterin käytössä oleva pysyväismuisti on jakautunut kahteen pääluokkaan. Ainakin joillain klusterin solmuilla on käytössään omaa pysyväismuistia toisaalta klusteri on yhteydessä johonkin ulkoiseen verkkoon, jossa on varmasti tarjolla erilaisia tiedostopalvelimia.

Beowulf-klusterin sisäinen pysyväismuisti Jos tehdään oletus, että Beowulf-klusterin solmuilla on pysyväismuistia, rajoitetaan mahdollisia alustoja joihin järjestelmä soveltuu. Järkevä oletus klusterin sisäisen pysyväismuistin määrälle yhtä solmua kohti on joitakin satoja megatavuja. Klusterilla on mahdolisesti isäntäsolmuja, joiden tallennuskapasiteetti voi olla huomattavasti suurempi kuin tavallisten solmujen.

Beowulf-klusterin ulkoinen pysyväismuisti Beowulf-klusterista on ainakin yksi solmu (isäntäsolmu) liitetty ulkoisiin verkkoihin ja tieteelliseen laskentaan kesittyvissä laitoksissa on usein tarjolla kymmenien gigatavuja tilaa tiedostopalvelimilla. Voidaan olettaa, että harvoin käsiteltäviä suuriakin tietomassoja voidaan tarpeen vaatiessa säilyttää Beowulf-klusterin ulkopuolella. Jos huomioidaan Beowulf-klusterin ulkopuoliset tallennusmahdollisuudet, yläraja käytettävän pysyväismuistin määrälle on joitakin satoja gigatavuja.

Syötteet Käsiteltävät makromolekyylien kolmiulotteiset mallit ovat useiden kymmenien megatavujen kokoisia. Elektronimikroskooppikuvat ovat pieniä, mutta niitä on paljon. Mallin ja kuvien tallettamiseen tarvitaan pysyväismuistia. Jos klusterissa ei ole omaa pysyväismuistia tai mallia ja kuvia ei tarvitse hakea pysyväismuistista useita kertoja, paras säilytystila mallille ja kuville on klusterin ulkopuolinen tiedostopalvelin tai tietokanta.

Välitulokset Pysyväismuistia saatetaan tarvita myös vällitulosten tallettamiseen. Jos välituloksia käytetään harvakseltaan, mutta ne ovat suuria Beowulf-klusterin ulkopuolinen tiedostopalvelija tai tietokanta on paras ratkaisu niiden tallettamiseen. Jos välitulokset ovat sopivan kokoisia ja niitä käsitellään usein, Beowulf-klusterin prosessoreiden omat kovalevyt ovat tehokkain vaihtoehto.

4.2.4 Beowulf-klusterin rajoitukset

Se, että järjestelmä toimii Beowulf-klusterissa asettaa joitakin rajoituksia yksittäisen ohjelmakomponentin toiminnalle. Näytön ja näppäimistön puute vaikeuttaa vakavista virhetilanteista toipumista ja erityisesti kehitysvaiheessa tarvitta informaatio järjestelmän sisäisistä toiminnoista eri solmuilta on koottava jollakin mekanismilla yhteen, jotta siitä saataisiin täysi hyöty irti. MPI-kirjasto tarjoaa tukea virheilmoitusten käsittelyyn. Koska orientointijärjestelmän osat tulevat toimimaan useissa eri Beowulf-klusterin solmuissa saman aikaisesti on välttämätöntä, että osat ovat mahdollisiman itsenäisiä käyttäjään nähden ja niiden hallinta on mahdollista tehdä keskitetysti. (Kohdassa 4.1.1 on lisää solmun syöte- ja tuloste laitteista.)

4.2.5 Kehitystyökalujen rajoitukset

Koska ohjelma toteutetaan C-kielellä, kehitystyökaluksi kelpaa mikä tahansa tekstieditori ja standardin mukainen C-kääntäjä. Ympäristössä, johon on olemassa toimiva C-kääntäjä, voidaan olettaa, että on myös olemassa muut välttämättömät kehitystyökalut. Koska käytettävät kehitystyökalut eivät ole erikoistuneita tai poikkeuksellisia, kehitystyökalut eivät aseta rajoituksia orientointijärjestelmän alustariippumattomuudelle. Toisaalta kehitystyökalut eivät myöskään tue automaattisesti tapahtuvaa laskennan rinnakkaistamista tai RAD-kehitystä.

4.2.6 MPI-kirjaston rajoitukset

MPI-kirjasto ei aseta rajoituksia ohjelmistolle. MPI-kirjaston käytön aikana kaikki yhden prosessorin tarjoamat ja verkkoon kytkettyjen koneiden tarjoamat palvelut ovat käytössä. MPI-kirjaston käyttö sinällään ei aiheuta merkittäviä lisäyksiä järjestelmän muistitila- , laskentateho- tai tiedonsiirtovaatimuksiin. MPI-kirjasto tukee kaikka C- ja Fortran kielten tietotyyppejä.

C-kielen rajoitukset C-kieli ei aseta merkittäviä rajoituksia orientointijärjestelmälle. Kaikki laitteiston piirteet ovat käytettävissä C-kielestä. C-kielellä toteutetun ohjelmiston muisitila- ja suoritusaika vaatimukset eivät poikkea merkittävästi saman ohjelmiston suoraan konekielellä toteutetusta versiosta. Konekieli taas ei sinänsä aseta mitään rajoituksia koneen optimaaliselle hyödyntämiselle.

5 Suorituskykyvaatimukset

Orientointijärjestelmälle on asetettu vaatimuksia sekä suoritusajan että yksittäisen solmukoneen keskusmuistin koon suhteen.

Järjestelmän tulee suoriutua asiakkaan tarjoamassa hajautetun muistin rinnakkaiskoneessa vähintään tuhannen (1000) kaksiulotteisen elektronimikroskooppikuvan orientoinnista viikon (7 vrk) aikana. Orientointi koostuu makromolekyylin kolmiulotteisen mallin projisoimisesta eri suunnista ja jokaisen saadun projektion vertaamisesta jokaiseen elektronimikroskooppikuvaan. Kun projektioita lasketaan esim. asteen välein täytyy projisointeja siis tehdä vähintään $180 \times 180 = 32400$ kpl/vko ≈ 3.2

kpl/min ja kaksiulotteisten kuvien vertailuja, joissa myös kaikki kiertokulmat käydään läpi, vastaavasti $32400 \times 1000 = 32400000$ kpl/vko ≈ 3214 kpl/min

Järjestelmä ei saa toimiakseen vaatia miltään suoritusympäristön yksittäiseltä solmukoneelta enempää keskusmuistia kuin 128 MB. Suurin keskusmuistin tarve syntyy todennäköisesti kolmiulotteisen $(256 \times 256 \times 256)$ mallin pitämisestä projisointia varten muistissa.

Lähteet

[Lanc98] Cristopher J.Lanczycki, Calvin A. Johnson, Benes L. Trus, James F. Conway, Alasdair C. Steven, Robert L. Martino
US National National Institutes of Health
Parallel Computing Strategies for Determining Viral Capsid Structure by Cryoelectron Microscopy
IEEE Computational Science & Engineering, Huhti-Touko 1998, s. 76-79

[Linux] http://www.linux.org

[Mandrake] http://www.linux-mandrake.com/en/

[BeowulfSaga] http://www.newadvent.org/cathen/09121a.htm

[ATM] http://www.atmforum.com/

[Terra] http://www.pgroup.com/p6_success/Terra.htm

[Resch96] Chance Reschke, Thomas Sterling, Daniel Ridge, Daniel Savarese, Donald Becker, Phillip Merkey

A Design Study of Alternative Network Topologies for the Beowulf Parallel Workstation Proceedings, High Performance and Distributed Computing, 1996

[Beowulf] http://www.beowulf.org/
http://www.hq.nasa.gov/hpcc/insights/vol7/beowulf.htm
Donald J. Becker, Thomas Sterling, Daniel Savarese, John E. Dorband, Udaya A. Ranawak, Charles V. Packer
Beowulf: A Parallel Workstation for Scientific Computation
Proceedings, International Conference on Parallel Processing, 95

Daniel Ridge, Donald Becker, Phillip Merkey, Thomas Sterling, Becker, Phillip Merkey
Beowulf: Harnessing the Power of Parallelism in a Pile-of-PCs
Proceedings, IEEE Aerospace, 1997

[MPI] http://www.mpi-forum.org/index.html
http://www-unix.mcs.anl.gov/mpi/

[Gard95] Richard J. Gardner
Geometric Tomography
Encyclopedia of Mathematics and Its Applications (volume 58),
Cambridge University Press, 1995

[Frank96] Joachim Frank
Three-Dimensional Electron Microscopy of Macromolecular Assemblies
Academic Press, 1996

[Frank92] Joachim Frank
Electronic Tomography: Three-Dimensional Imaging with the Transmission El ectron Microscope
Plenum Press, 1992

[Herman99] Gabor T. Herman and Attila Kuba
Discrete Tomography: Foundations, Algorithms and Applications
Birkhäuser, 1999

[Toft96] Peter Toft
The Radon Transform: Theory and Implementation
PhD thesis, Department of Mathematical Modelling, Technical University of Denmark, 1996

[Kivioja99] Teemu Kivioja
Virusten automaattinen paikantaminen elektronimikroskooppikuvista
Master's thesis, Tietojenkäsittelytieteen laitos, Helsingin yliopisto, 1999