



**TEKNIikka JA LIIKENNE**

**Tietotekniikka**

**Ohjelmistotekniikka**

**INSINÖÖRITYÖ**

**KORKEAN KÄYTETTÄVYYDEN KLUSTERIT LINUX-YMPÄRISTÖSSÄ**

**Työn tekijä: Arto Logrén  
Työn ohjaajat: Matti Luukkainen**

**Työ hyväksytty: \_\_\_\_ . \_\_\_\_ . 2008**

**Matti Luukkainen  
yliopettaja**



## **ALKULAUSE**

Tämä insinöörityö tehtiin Metropolia Ammattikorkeakoululle. Lämpimät kiitokset työn ohjaajalle yliopettaja Matti Luukkaiselle sekä kaikille tämän työn tekemisessä minua tukeneille henkilöille.

Helsingissä 14.11.2008

Arto Logrén

## TIIVISTELMÄ

<b>Työn tekijä:</b> Arto Logrén	
<b>Työn nimi:</b> Korkean käytettävyyden klusterit Linux ympäristössä	
<b>Päivämäärä:</b> 24.11.2008	<b>Sivumäärä:</b> 37 s. + 3 liitettä + cd-rom
<b>Koulutusohjelma:</b> Tietotekniikka	<b>Suuntautumisvaihtoehto:</b> Ohjelmistotekniikka
<b>Työn ohjaaja:</b> Matti Luukkainen	
<p>Tämä insinööri työ tehtiin Metropolia Ammattikorkeakoululle. Työssä perehdyttiin alan kirjallisuuden ja sähköisten lähteiden avulla korkean käytettävyyden klustereihin Linux-ympäristössä.</p> <p>Aluksi työssä selvitettiin, mikä on klusteri ja mitkä ovat niiden hyödyt sekä käyttökohteet. Klustereiden ohella työssä tutustutaan virtuaaliseen testi ympäristöön, jollaisessa klusteri toteutettiin sekä klustereihin liittyviin oheisohjelmistoihin, kuten etähallintaan ja palvelimien valvontaan. Selvityksen jälkeen toteutettiin virtuaaliympäristössä korkean käytettävyyden kuormaa tasaava testiklusteri, jonka avulla teorioita testattiin käytännössä.</p> <p>Käytännön testeissä klusteri toimi odotetulla tavalla ja testien perusteella saatiin hyvä ymmärrys korkean käytettävyyden klustereista sekä niihin liittyvistä rajoituksista ja vaatimuksista. Työssä käytetyt ohjelmistot ovat laajassa käytössä korkeatasoisissa palveluissa, ja työ antaa hyvän kuvan, millaiseen tekniikkaan monet palvelut perustuvat.</p>	
<b>Avainsanat:</b> Linux, korkea käytettävyys, klusteri	



## ABSTRACT

**Name:** Arto Logrén

**Title:** High availability clusters in Linux environment

**Date:** 24.11.2008

**Number of pages:** 37 pages + 3 attachments  
+ cd-rom

**Department:**  
Information technology

**Study Programme:**  
Software technology

**Instructor:** Matti Luukkainen

This study was carried out for Metropolia University of applied sciences. The purpose of this work was to explore high availability clusters in Linux environment through recent research literature as well as electronic sources.

This study describes high availability clusters and discusses their advantages and how clusters are used. In addition to clusters, this study experimented with a virtual testing environment, in which the cluster was implemented, along with utility programs, such as remote control and service monitoring, relating to clusters. The high availability and load balancing test cluster was then implemented in the virtual testing environment and the theories were tested in practice.

In practice the cluster worked according to expectations and the tests provided a good understanding about high availability clusters as well as their limitations and requirements. The software used in this work is widely used in services of a high level and this work presents a good overall picture about the technology used in many services today.

**Keywords:** Linux, high availability, cluster

# SISÄLLYS

## ALKULAUSE

## TIIVISTELMÄ

## ABSTRACT

## LYHENTEITÄ JA MÄÄRITELMIÄ

<b>1</b>	<b>JOHDANTO</b>	<b>1</b>
<b>2</b>	<b>KLUSTERITEORIAA</b>	<b>2</b>
<b>2.1</b>	<b>Laskenta-klusterit</b>	<b>2</b>
<b>2.2</b>	<b>Kuormaa tasaavat klusterit</b>	<b>3</b>
<b>2.3</b>	<b>Korkean käytettävyyden klusterit</b>	<b>4</b>
2.3.1	<i>Korkean käytettävyyden mittaaminen</i>	5
2.3.2	<i>Saavutettavuus</i>	6
2.3.3	<i>Kiintolevyn tiedostojärjestelmä</i>	9
2.3.4	<i>Klusterin toiminta vikatilanteessa eli yliheitto-prosessi</i>	10
2.3.5	<i>Klusterin toiminnan valvonta</i>	11
2.3.6	<i>Klusterin etähallinta</i>	12
<b>3</b>	<b>OHJELMISTOT</b>	<b>13</b>
<b>3.1</b>	<b>Ubuntu</b>	<b>13</b>
<b>3.2</b>	<b>Ldirectord</b>	<b>13</b>
<b>3.3</b>	<b>HA eli High Availability</b>	<b>14</b>
<b>3.4</b>	<b>LVS eli Linux Virtual Server</b>	<b>14</b>
<b>3.5</b>	<b>VMware Player ja virtualisointi</b>	<b>18</b>
<b>3.6</b>	<b>NoMachine</b>	<b>19</b>
<b>4</b>	<b>TESTIKLUSTERIN TOTEUTUS</b>	<b>19</b>
<b>4.1</b>	<b>Testialustan asennus ja asetukset</b>	<b>20</b>
4.1.1	<i>Isäntäkone</i>	20
4.1.2	<i>Ohjelmistojen asennus isäntäkoneeseen</i>	21
<b>4.2</b>	<b>Klusterin pystyttäminen</b>	<b>25</b>
4.2.1	<i>Ytimen kääntäminen ja asentaminen</i>	25
4.2.2	<i>Ohjaajakoneiden asentaminen ja asetukset</i>	30
4.2.3	<i>Palvelimien asentaminen ja asetukset</i>	34
<b>4.3</b>	<b>Klusterin testaus</b>	<b>36</b>

<b>5</b>	<b>LOPPUMIETTEET</b>	<b>38</b>
	<b>VIITELUETTELO</b>	<b>39</b>
	<b>LIITE 1 - UBUNTUN ASENTAMINEN</b>	<b>1</b>
	<b>LIITE 2 - VMWAREN ASENNUSLISTAUS</b>	<b>1</b>
	<b>LIITE 3 - LVSDRRS TIEDOSTON SISÄLTÖ</b>	<b>1</b>

## LYHENTEITÄ JA MÄÄRITELMIÄ

Ext3	<i>Third extended filesystem.</i> Linux-käyttöjärjestelmissä yleinen tiedostojärjestelmä.
IPVS	<i>Internet Protocol Virtual Server.</i> Linuxin ytimessä oleva kokoelma ohjelmistoja.
LVS	<i>Linux Virtual Server.</i> Kuormaa tasaava klusteri.
RAID	<i>Redundant array of independent disks.</i> Fyysisten levyjen joukko, jota käytetään loogisesti suorituskyvyn ja luotettavuuden parantamiseksi
SPOF	<i>Single point of failure.</i> Yksittäinen kohta, jonka rikkoutuessa koneen ajama palvelu keskeytyy.
STONITH	<i>Shoot the other node in the head.</i> Tekniikka, jolla tietokone katkaisee virran toisesta tietokoneesta.
Ubuntu	Suosittu Linux-jakelu.
UPS	<i>Uninterruptible Power Supply.</i> Laite, joka suojaa siihen kytkettyjä laitteita sähköverkon häiriöiltä ja sähkökatkoksilta.

## 1 JOHDANTO

Tietokoneet ja tietotekniikka helpottavat ihmisten työtä monilla tavoin ja niillä on helppo toteuttaa ja automatisoida tehtäviä, jotka muuten olisivat hankalia tai jopa mahdottomia toteuttaa. Osa tehtävistä, joita ne hoitavat ovat kriittisiä turvallisuuden, talouden tai muun tärkeäksi katsotun asian kannalta ja siksi niiltä vaaditaan suurta luotettavuutta ja tavoitettavuutta. Eri viranomaisjärjestelmien tulee olla saavutettavissa aina, jotta esimerkiksi elintärkeä pelastustoiminta ja järjestyksenvalvonta toimii vuoden jokaisena päivänä 24 tuntia vuorokaudessa. Yrityspuolella taas tarjotaan palveluita, joiden toimimattomuus voi tuottaa suuria tappioita rahallisesti ja vahingoittaa pahasti yrityksen julkisuuskuvaa sekä asiakassuhteita.

Yksittäinen tietokone voi kestää käyttöä paljon oletettua käyttöikää pidempään, mutta se ei takaa, että palvelu on tavoitettavissa aina kriittisellä hetkellä. Uuteenkin koneeseen tai luotettavaan yhteyteen, jonka päässä kone on, voi tulla vikaa, tai palvelu voi ylikuormittua odottamattoman suuren kysynnän vuoksi. Tavoitettavuuden koko ketjun tulee toimia luotettavasti, ja tällaisen ratkaisun osaksi Linux-klusterit sopivat todella hyvin kustannustehokkaana ja luotettavana ratkaisuna.

Klustereiden tekemiseen on olemassa niin ilmaisia kuin kaupallisiakin työkaluja monelle eri alustalle. Keskityimme tässä työssä tutkimaan ilmaisia vaihtoehtoja Linux-alustalle, sillä ne ovat kaikkien saavutettavissa hinnan ja saatavilla olevan ohjeistuksen sekä tuen puolesta. Klustereiden lisäksi tutustuimme valvontatyökaluihin ja etähallintaan, jotka ovat välttämättömiä klusterin toiminnan tarkkailuun ja vikatilanteiden hallintaan.

Klusterin toteutus ja testaus tehtiin virtuaaliympäristössä, koska täydellisen korkean käytettävyyden järjestelmän toteuttaminen vain testimielessä olisi vaaditun laitteiston ja taloudellisten syiden takia liian kallista. Tutkittujen ohjelmistojen testaus rajataan niin, että se voidaan suorittaa virtuaalisesti. Tällä tavoin tehdyt testit vastaavat täysin oikealla laitteistolle tehtyjä testejä ja ovat vertailukelpoisia niiden kanssa.



## 2 KLUSTERITEORIAA

Klusteri sanana on tuttu monesta yhteydestä. Sitä käytetään monella tieteen, talouden ja teollisuuden alalla mutta tarkoitus pysyy kaikilla niillä samana. Sanalla tarkoitetaan tekijöiden, asioiden tai osien joukkoa, joka tai jotka ovat riippuvaisia toisistaan tai saavat yhteen toimimalla huomattavaa hyötyä. Tietotekniikassa klusterilla tarkoitetaan tietokoneiden joukkoa, joka pyrkii yhteen toimimalla toteuttamaan niille suunniteltua ja annettua tehtävää. Klusterin tarkka määritelmä vaihtelee hieman, riippuen miltä kannalta asioita katsoo ja minkä tyyppistä klusteria määritellään.

Klustereita on kehitetty moneen eri tarkoitukseen ja tarpeeseen. Alla selitetään lyhyesti kunkin klusterityypin käyttötarkoitus ja toimintaperiaate.

### 2.1 Laskenta-klusterit

Suurteholaskentaa varten on olemassa kolme suosittua toteutustapaa. Ensimmäinen on supertietokone, jossa on paljon prosessoreita, jota hallitaan keskitetysti ja jonka laskentatehoa voidaan ohjata tarpeen mukaan nopeastikin. Supertietokone sopii hyvin laskentatöihin, joissa tulokset riippuvat toisistaan. Haittapuolena on kallis hinta. Toinen toteutustapa on klusteri, joka koostuu suuresta määrästä tietokoneita, jotka on yhdistetty nopealla lähiverkolla. Klusterissa voidaan käyttää vakiokomponentteja, ja tästä syystä kustannukset pysyvät paremmin hallinnassa. Todella suosittu vaihtoehto laskentaan on Beowulf-klusteri [1, s. 635]. Kolmas toteutustapa on grid- eli hajautettu laskenta, joka on ikään kuin klusteri mutta ilman nopeaa lähiverkkoa. Tämän takia grid-laskenta soveltuu vain käyttökohteisiin, joissa laskenta voidaan jakaa toisistaan riippumattomiin osiin. Hyvinä puolina grid-laskenta mahdollistaa yksilöidenkin osallistumisen tarpeelliseksi katsomiinsa projekteihin, sillä pienin vaadittu yksikkö on yksittäinen tietokone. Selkeä huono puoli grid-laskennassa on kulloinkin saatavilla olevan tehon vaihtelu ja epävarmuus. On hyvin vaikeaa ennustaa etukäteen, milloin yksittäiset koneet ovat verkossa ja vapaana käyttöön. Ohjelmisto tulee laskentatyyppistä riippumatta valita parhaan suorituskyvyn saamiseksi aina käyttökohteen mukaan.

Yksittäisten supertietokoneiden hankkiminen on kallista ja siksi on jo pitkään ollut olemassa edullisempia tapoja laskentatehon hankkimiseen. Käyttökohde ratkaisee, tarvitseeko ratkaistava ongelma yksittäisen suureen

laskentatehooon yltävän koneen vai voidaanko laskenta hajauttaa heikomman laskentatehon omaavien koneiden joukolle. Maailman viidensadan tehokkaimman supertietokoneen joukossa on tilastojen mukaan 400 klustereilla toteutettua järjestelmää [2], mikä osoittaa, että teknologia on toimivaksi todettua ja laajasti käytössä.

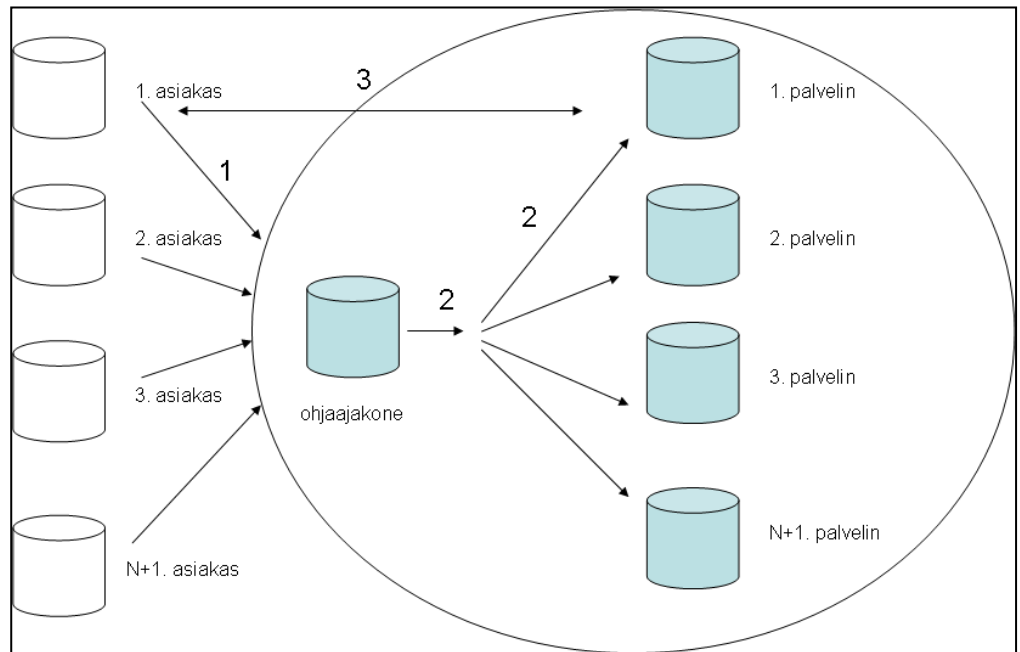
## 2.2 Kuormaa tasaavat klusterit

Internet on hyvin suosittu palvelu ja siellä on paljon palveluja, jotka keräävät vakituisesti suuren määrän käyntejä tai joiden kysyntä kasvaa nopeasti ja ennalta arvaamatta. Yksi palvelin ei tällaisessa tilanteessa millään pysty pitämään palvelun laatua vaaditulla tasolla, vaan siihen vaaditaan useamman palvelimen panos. Samoin palvelut, joiden käyttäjämäärä on jatkuvasti suuri tai palvelut, jotka halutaan lokalisoida vaikkapa maakohtaisesti, käyttävät kuormaa tasaavaa tekniikkaa. Tämän voi käytännössä huomata esimerkiksi siitä, kun kirjoittaa osoiteriville palvelun osoitteen, se ohjautuu maakohtaiseen osoitteeseen. Käyttökohteita on paljon ja edellä mainittiin vain muutama esimerkki.

Kuormaa tasaava klusteri (Load Balancing Cluster) voi jakaa tulevia palvelupyynnöitä useamman palvelimen joukolle, josta voidaan ottaa tarpeen mukaan palvelimia huoltoon tai päivitykseen palvelun silti toimiessa normaalisti. Tällainen ratkaisu sallii myös palvelun kapasiteetin kasvattamisen tarpeen mukaan, kun palvelinten joukkoa voidaan kasvattaa sujuvasti. Kuormaa tasaava klusteri ja korkean käytettävyyden klusteri (kts. luku 2.3) voidaan rakentaa samaan järjestelmään, jolloin saadaan suorituskykyinen ja luotettava palvelu. Tämä onkin yleinen ratkaisu, sillä verkkopalveluiden oletetaan olevan aina tavoitettavissa. Toteutuskin on helppoa, sillä ohjelmisto, joilla korkea käytettävyys ja kuormantasaus saadaan aikaan, on osittain sama. Palvelinten joukkoa voidaan muokata suuremmaksi tai pienemmäksi mutta tällöin pitää muistaa päivittää tehdyt muutokset liikennettä ohjaavalle koneelle suurimman hyödyn ja parhaimman tehon saamiseksi. Ohjaajakone valvoo yleensä automaattisesti mitkä palvelimet ovat kunnossa ja ohjaa yhteyksiä vain näille koneille.

Kuvasta 1 nähdään hyvin kuormaa tasaavan klusterin toimintaperiaate. Ympyrän sisällä olevat palvelimet kuuluvat joukkoon, jolle liikenteenohjaajana toimiva kone ohjaa yhteyksiä. Kaikki tiettyyn

osoitteeseen tulevat yhteyspyynnöt menevät aluksi liikennettä ohjaavalle koneelle (kuva 1, kohta 1). Ohjaaja ohjaa pyynnön palvelimelle, jonka kuorma on vähäisin (kuva 1, kohta 2) ja lopuksi asiakaskone ja palvelin, jolle liikenne ohjattiin vastaa asiakkaan pyyntöön (kuva 1, kohta 3).



Kuva 1. Kuormaa tasaavan klusterin osat ja toiminta.

### 2.3 Korkean käytettävyyden klusterit

Korkean käytettävyyden klusterin tehtävä on luotettavuuden parantaminen ja kasvattaminen. Tarkoituksena on varautua kuviteltavissa oleviin palvelun keskeyttäviin uhuihin ja niiden toteutumiseen. Mikään yksittäinen laitteiston vika (SPOF, Single Point Of Failure) [3, s. 42] ei saa haitata palvelun saatavuutta. Korkean käytettävyyden klusteri rakennetaan tästä syystä tavalla, jossa on päällekkäisyyttä mikä parantaa klusterin vikasietoisuutta. Sadan prosentin käytettävyys on lähes mahdotonta saavuttaa, sillä laitteistot vaativat huoltoa ja ohjelmistot päivityksiä. Palvelussa voidaan varautua uhkien ohella suunniteltuun huoltoikkunaan, jonka aikana tehdään tarvittavat ohjelmistopäivitykset ja korjataan mahdolliset laiteongelmat.

Kun korkean käytettävyyden järjestelmää aletaan suunnitella, tulee miettiä, miten luotettava järjestelmä halutaan. Mitä luotettavampi järjestelmästä halutaan, sitä kalliimmaksi se tulee. Tämän takia pienempään tarpeeseen ei ole järkevää rakentaa mahdollisimman luotettavaa järjestelmää vaan on viisasta tyytyä vaatimattomampaan järjestelmään, missä saavuttamattomuus

voidaan ottaa huomioon ja missä se on hyväksyttävää. Isoissa ja tärkeissä palveluissa saavutettavuus on kuitenkin tärkeintä, ja tällöin suuret kulut voidaan hyväksyä.

Korkean käytettävyyden klusterin voi toteuttaa kahdella eri tavalla ilman, että kytkennät tai kaaviokuvat muuttuvat [1, s. 627]. Ensimmäinen vaihtoehto on passiivinen varakone (Passive Standby), jossa pääkone tekee kaiken työn varakoneen keskittyessä vain ja ainoastaan tarkastelemaan, onko pääkone kunnossa ja hoitaako se palvelua. Pääkoneen rikkoutuessa varakone ottaa sen tehtävät hoitaakseen. Toinen vaihtoehto on aktiivinen varakone (Active Secondary). Tässä vaihtoehdossa kumpikin palvelin tekee töitä ja tarkkailee toisensa tilaa. Vikatilanteessa liikenne ohjataan ehjäksi jäävälle koneelle ja vastuu palvelun tarjoamisesta jää sille. Jälkimmäisessä vaihtoehdossa koneiden kuorma tulee mitoittaa sopivaksi, jotta toisen koneen rikkoutuessa palvelu ei välittömästi ylikuormittuisi yhden koneen ollessa teholtaan riittämätön vastaamaan sille tuleviin pyyntöihin. Jos kuormitus on laskettu turhan tarkalle ja esimerkiksi kahden koneen järjestelmästä häviää toinen kone, eli 50 %, niin jäljelle jäävälle koneelle tulee todennäköisesti ylikuormitustilanne. Jos palvelu ei kaadu, niin sen taso laskee.

### 2.3.1 Korkean käytettävyyden mittaaminen

Korkeaa käytettävyyttä mitataan saavutettavuuden prosenteilla [4, s. 90 - 91]. Tällä tarkoitetaan aikaa, jonka palvelu on saavutettavissa siitä ajasta, mikä oikeasti kuluu. Taulukosta 1 käy hyvin ilmi, kuinka kauan palvelu on saavuttamattomissa kullakin yhdeksikköjen määrällä.

99 % käytettävyys on vielä saavutettavissa yhdellä koneella ja hyvällä tuurilla mutta jos vaatimuksena on sitä korkeampi käytettävyys, pitää koneita olla enemmän [4, s. 9].

Saavutettavuus yhdeksikköinä	Palvelu saavuttamattomissa (%)	Aika, jona palvelu saavuttamattomissa
Yksi (98%)		2 7,3 päivää
Kaksi (99%)		1 3,65 päivää
Kolme (99,9%)		0,1 8 tuntia 45 minuuttia
Neljä (99,99%)		0,01 52,5 minuuttia
Viisi (99,999%)		0,001 5,25 minuuttia

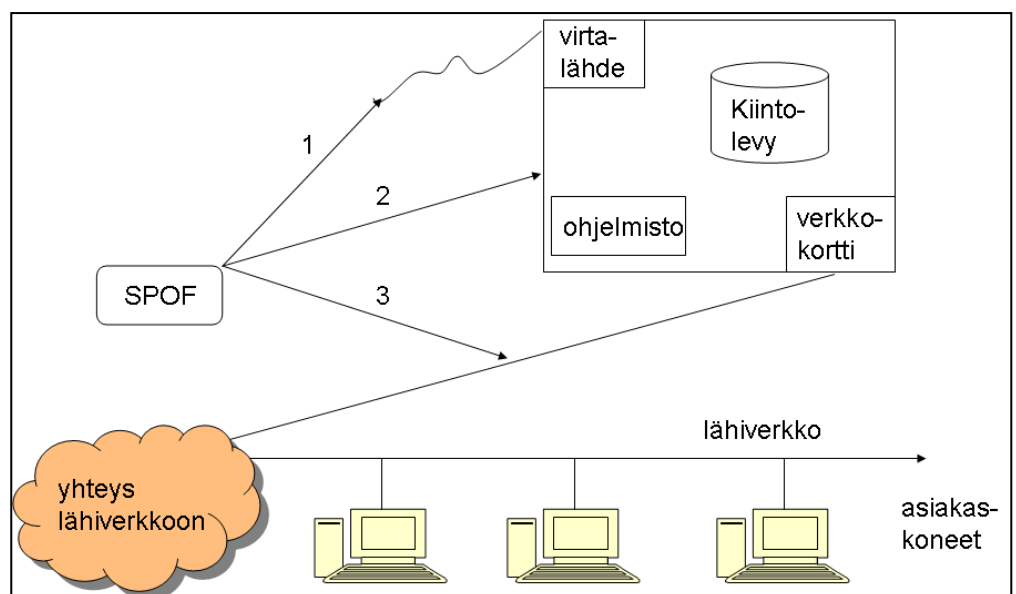
Taulukko 1. Saavutettavuuden yhdeksiköt.

### 2.3.2 Saavutettavuus

On hyvä erottaa eri termejä, joita usein käytetään, kun puhutaan järjestelmän luotettavuudesta. Usein puhutaan uptime-ajasta, joka tarkoittaa sitä aikaa, minkä kone on ollut päällä yhtä mittaa. Siinä ei kuitenkaan oteta huomioon, onko palvelu oikeasti ollut saatavilla vai ei. Korkean käytettävyyden järjestelmistä puhuttaessa saavutettavuus on sana, jolla tarkoitetaan oikeaa asiaa. Kun palvelu on saavutettavissa, niin se on käytettävissä.

Tässä työssä mainitaan usein korkean käytettävyyden klusteri kahden koneen toisiaan tarkkailevana järjestelmänä, koska testialusta on suunniteltu siten. Koneita voi olla kuitenkin enemmän, jopa useita kymmeniä [5]. Yhdellä koneella tarkoitetaan normaaleista komponenteista kasattua konetta, joiden rinnakkaista käyttöä tässä työssä tutkitaan.

Tilannekuva on hieman erilainen tarjottaessa palvelua lähiverkossa ja julkisessa verkossa. Lähiverkon tilannetta selkeyttävää kuva 2, jossa kuvataan helposti vahingoittuva järjestelmä, ja kuva 3, jossa kuvataan varmistettu järjestelmä. Kun palvelua tarjotaan ulkoisessa verkossa, tulee ottaa huomioon itse laitteiston lisäksi myös mm. yhteys ulkoiseen verkkoon ja siihen liittyvät riskit. Aina järjestelmää rakennettaessa ja muutettaessa tulee tekniikasta riippumatta ottaa huomioon, että mukana on päällekkäisyyttä ja että joku toinen laite voi korvata rikkimenevän.



Kuva 2. Mahdolliset palvelun pysäyttävät viat (SPOF) yhden koneen järjestelmässä.

Kuvassa 2 on osoitettu kaikki kohdat, joiden vioittuessa palvelun tarjoaminen estyy. Mahdolliset ongelmat voi jakaa kolmeen osaan, jotka on osoitettu kuvassa: virransyöttö (1), koneen toiminta (2) ja verkkoyhteys (3).

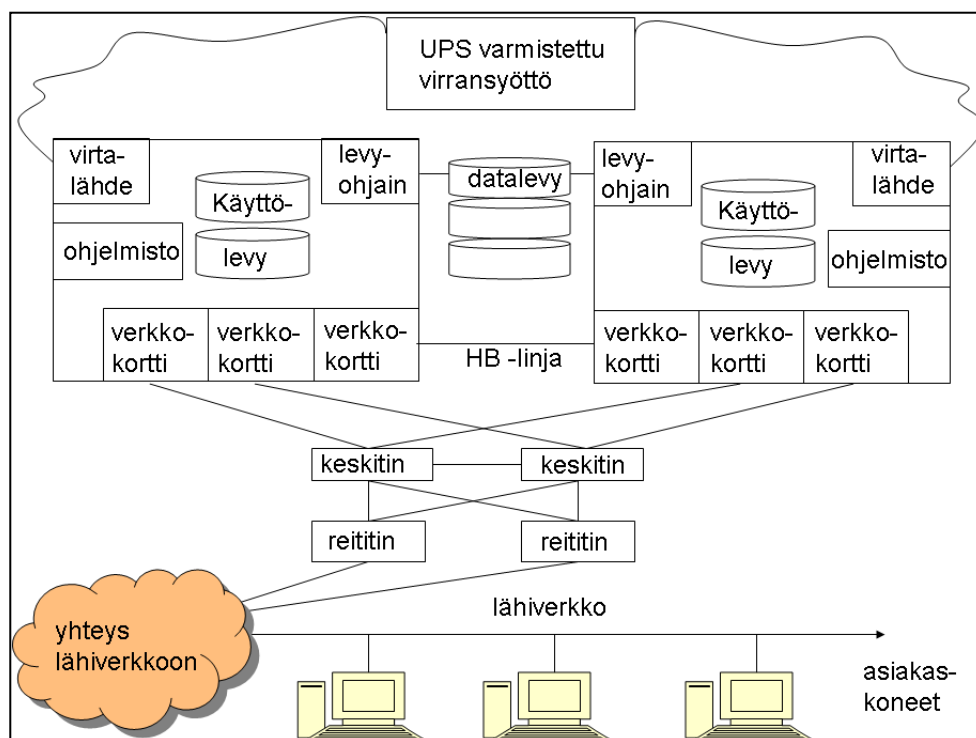
Sähköverkon toiminta on Suomessa vakaata, mutta se ei silti riitä takaamaan tarpeeksi tasaista virransyöttöä, ja hetkellinenkin heilahtus virransyötön laadussa johtaa usein koneen toiminnan häiriintymiseen. Virransyötön varmuutta voidaan parantaa kahdella tavalla. Ensimmäisenä kone tulee kytkeä sähköverkkoon UPS-laitteen (Uninterruptible Power Supply) kautta. UPS on laite, joka tasaa sähkön laatua ja sisältää akuston, jonka avulla kone voi jatkaa toimintaansa lyhyiden sähkötoimituskatkosten yli häiriintymättä. UPS suodattaa jännitepiikit ja alijännite- tai virtakatkotilanteissa käyttää omia akkujaan koneen virransyöttöön. UPS-laitteita on kotikoneille tarkoitetuista malleista aina konesalienkin kattaviin ratkaisuihin. Haluttu varmuustaso ratkaisee, kuinka järeä laitteisto kulloinkin tarvitaan. Toinen tapa parantaa varmuutta on oma varageneraattori. Kun katko on tarpeeksi pitkä ja kun UPS-laitteiston akuston varaus laskee kriittisen rajan alapuolelle, voidaan ottaa käyttöön varageneraattorit. Niiden avulla selvittää käytännössä kaikenlaisten katkosten yli niin pitkään, kuin polttoainetta on saatavilla.

Toisena kuvassa 2 olevassa järjestelmässä on itse koneen toiminta. Osat ovat yleensä yksittäisiä ja niiden rikkoutuessa kone on vietävä huoltoon. Vikaherkimpiä osia ovat osat, joissa on liikkuvia osia. Tavallisilla komponenteilla voi parantaa yksittäisen koneen varmuutta monilta osin ja edullisin kustannuksin. Verkkokortteja voi olla kaksi, jolloin toisen rikkoutuminen ei estä koneen verkkoliikennettä. Yksittäinen kiintolevy voidaan korvata RAID (Redundant Array of Independent Disks) -levypakalla, joka parhaimmillaan (RAID6) kestää kahden levyn rikkoutumisen ilman käyttökatkoksia. RAID on järjestelmä, joka mahdollistaa fyysisten kiintolevyjen joukon käyttämisen loogisesti yhtenä levynä parantaen luotettavuutta. Laitteistopohjainen RAID on suorituskyvyltään parempi ratkaisu kuin ohjelmistopohjainen. Tietokoneen ohjelmisto tulee valita testatuista ja hyväksi todetuista vaihtoehdoista. Ohjelmiston laatu ja luotettavuus ovat tärkeämpiä kuin laaja ominaisuuskirjo.

Kolmas kuvassa 2 osoitettu mahdollinen ongelma on verkkoyhteys. Yhden kaapelin varassa toimiva kriittinen järjestelmä ei ole varma ratkaisu.

Koneeseen lisätty toinen verkkokortti parantaa tilannetta, sillä silloin kone voidaan kytkeä verkkoon tarvittaessa useampaa eri reittiä.

Kuten kuvasta 2 havaitaan, minkä vain komponentin rikkoontuminen pysäyttää palvelun tarjoamisen, sillä päällekkäisyyttä ei ole. Palvelu voi toimia ilman ongelmia, mutta mitään kriittistä palvelua tällaisella koneella ei pitäisi ajaa, koska se voi yllättäen olla tavoittamattomissa.



Kuva 3. Kahdesta koneesta koostuva klusteri, jossa ei ole yksittäisiä palvelun estäviä kohtia.

Kuvassa 3 on kuvattu järjestelmä, jossa on otettu huomioon saavutettavuus rakentamalla päällekkäisyyttä järjestelmän joka kohtaan.

Koneita on kaksi kappaletta ja niiden virransyöttö on varmistettu UPS-laitteella. Koneiden käyttölevyt on muutettu RAID-levypakaksi, ja datalevy on ulkoistettu erilliseksi osaksi, jotta koneen rikkoutuminen ei vie mennessään dataa, johon jäljelle jäävän koneen pitäisi päästä käsiksi. Käyttölevyllä on vain koneen itsensä toimintaan liittyvä data. Verkkokortteja on verkkoliikennettä varten kaksi kappaletta kummassakin koneessa ja kolmansilla verkkokorteilla on toteutettu heartbeat-linja (kuvassa 3 HB-linja) koneiden välillä. Heartbeat-linja mahdollistaa yhteydenpidon koneiden välillä ja sen avulla koneet valvovat toistensa tilaa. Kun toinen kone rikkoutuu, niin toinen huomaa rikkoutumisen heartbeat-linjan avulla.

Yhteys verkkoon on varmistettu kytkemällä koneet eri verkkokomponentteihin ja kytkemällä nämä vielä ristiin. On hyvä huomata, että myös verkkolaitteet tulisi olla kytkettynä varmistetussa virransyötössä niiden toiminnan takaamiseksi. Palvelinhuoneessa päällä olevat palvelimet eivät ole kenenkään käytössä, jos niihin sähkökatkon aikana ei pääse mitään kautta käsiksi.

### 2.3.3 Kiintolevyn tiedostojärjestelmä

Linux-käyttöjärjestelmissä hyvin yleinen ext2-tiedostojärjestelmä on vakaa ja hyväksi todettu. Siitä puuttuu kuitenkin oleellinen ominaisuus, joka on uudemmissa tiedostojärjestelmissä. Tämä on lokin pitäminen (Journaling) [4, s. 67].

Tiedostoja käsitellessä tiedostojärjestelmä tekee levyille koko ajan muutoksia. Kun koneesta katoaa yllättäen virta, voi tiedostojärjestelmän tekemä merkintä jäädä kesken. Koska ext2-tiedostojärjestelmässä ei ole loki-ominaisuutta, se pitää tämän takia tarkistaa aina yllättävien sammutusten jälkeen siihen tarkoitukseen erikseen tehdyllä ohjelmalla. Tämä ei sinänsä ole ongelma mutta suurien levyjen ollessa kyseessä aika, joka tarkistuksiin kuluu kasvaa huomattavaksi.

Jos käytössä on lokia pitävä tiedostojärjestelmä, toipuminen käy nopeammin, koska koko levyä ei tarvitse käydä läpi. Riittää kun lokista tarkistetaan, mitä oltiin kirjoittamassa, niin tiedetään, mistä kohtaa jatkaa.

#### *Ext3*

Ext3 on ext2:een perustuva tiedostojärjestelmä, johon on lisätty joitain ominaisuuksia kuten juuri aiemmin mainittu loki, joka nopeuttaa koneen toipumista yllättävistä sammumistilanteista. Koska ext2 ja ext3 ovat sukulaisia, siirtyminen niiden välillä käy suhteellisen helposti. Ext3 tulee vakiona muun muassa tässä työssä käytetyssä Linux-jakelussa, joten siirtymää ei tarvitse erikseen tehdä.

#### *ReiserFS*

ReiserFS on toinen lokiominaisuudella varustettu tiedostojärjestelmä, jossa on myös muita edistyneitä ominaisuuksia, joita ext2:sta ja ext3:stakin



puuttuu. ReiserFS osaa esimerkiksi varata lohkoja muuttuvalla koolla eikä kiinteällä, mikä säästää levytilaa.

ReiserFS:n käyttöönotto ext2-järjestelmässä ei kuitenkaan ole yksinkertainen suorittaa, ja se tuo mukanaan monia ongelmia eri ohjelmien kanssa. Uudessa järjestelmässä käyttöönotto on helpompi mutta huomioon kannattaa ottaa, että pääkehittäjä Hans Reiser on tuomittu vankilaan 25 vuodeksi, mikä saattaa vaikuttaa tiedostojärjestelmän kehitystyöhön.

#### *2.3.4 Klusterin toiminta vikatilanteessa eli yliheitto-prosessi*

Prosessi, jossa pääkoneen toiminta siirtyy varalla olleen koneen hoidettavaksi, on nimeltään yliheitto eli failover-prosessi.

##### *Heartbeat-linja*

Koneet käyttävät luvussa 3.2.3 esiteltyä heartbeat-linjaa toistensa tilan tarkkailuun. Linjan toteutus on mahdollista toteuttaa ainakin nollamodeemi-kaapelilla sarjaporttien välille sekä lähiverkkotekniikalla [4, s. 94 - 85].

Jos päädytään käyttämään lähiverkkotekniikkaa, tulisi heartbeat-liikenteen kulkea omassa verkossaan. Palveluverkon kuorma voi olla suuri ja jos koneiden toistensa tarkkailuun käyttämä liikenne ohjataan samaan verkkoon se voi häiriintyä ikävin seurauksin. Käytetystä ratkaisusta riippuen aika, jonka koneet odottavat toisen koneen vastausta heartbeat-viestiin on parhaimmillaan alle sekunti. Kun palveluverkon liikenne ja heartbeat-linjan liikenne kulkevat omissa verkoissaan, törmäysten vaaraa ei ole.

##### *Yliheitto-prosessin kulku*

Yliheitto alkaa palvelusta vastaavan koneen toiminnan häiriintyessä. Varalla ollut kone huomaa heartbeat-linjan kautta, että pääkone ei enää vastaa. Tällöin varakone ottaa haltuunsa pääkoneen verkko-osoitteen, ulkoisen levyjärjestelmän käyttöoikeuden ja alkaa hoitaa palvelua. Prosessi kuulostaa yksinkertaiselle, mutta siinä on monta ongelmaa, joihin pitää kiinnittää huomiota.

Ongelmia tulee, kun varapalvelin ottaa verkko-osoitetta haltuunsa [4, s. 100 - 101]. Varapalvelin ottaa virtuaalisen IP-osoitteen haltuunsa mutta kytkimellä

on edelleen tieto, että kyseiseen IP-osoitteeseen menevä liikenne tulisi ohjata koneeseen, jonka MAC-osoite kytkimellä on muistissaan. Kytkimen muisti päivittyy aikanaan, mutta siihen voi mennä liikaa aikaa ja siksi tarvitaan nopeampia tapoja osoitteen siirtymiseen ja liikenteen jatkumiseen. Nopein tapa ongelman ratkaisemiseen on palvelinkoneen MAC-osoitteen siirtäminen varapalvelimen kortin osoitteeksi. Vaikka MAC-osoite on periaatteessa laitteistopohjainen tunnus, se voidaan asettaa myös ohjelmallisesti, kunhan sekä laitteisto että käytettävät ohjelmistot molemmat sitä tukevat.

Toinen suuri ongelma ratkaistaan käytännöllä, joka kulkee nimellä stonith. Se tulee sanoista Shoot The Other Node In The Head [6, s. 163-164]. Myös termiä stonith käytetään ja tässä m tarkoittaa englannin kielen sanaa machine. Kummallakin tarkoitetaan samaa asiaa, vaikka pilkuntarkka tulkitseminen ei aivan sama olekaan. Termillä tarkoitetaan tilannetta, jossa vastuun palvelusta ottava kone sammuttaa toisen koneen.

Kuvitellaan tilanne, jossa palvelu toimii normaalisti pääkoneella toisen koneen vahtiessa pääkoneen tilaa. Koneiden välillä olevaan heartbeat -linjaan tulee vika mikä katkaisee koneiden välisen yhteyden. Tästä seuraa, että varalla ollut kone luulee pääkoneen rikkoutuneen ja yrittää ottaa haltuunsa osoitteet ja levyjärjestelmän. Koska pääkone ei kuitenkaan oikeasti ole rikki vaan toimii normaalisti ja yrittää jatkaa palvelun tarjoamista, tuloksena on vakava ongelma palvelun kannalta. Kun useampi laite taistelee samoista resursseista, tuloksena on varmasti jonkinlainen virhetilanne. Tästä tilanteesta käytetään nimitystä split brain.

Stonith on ulkoinen laite, jolla poistetaan tällaisen tilanteen mahdollisuus. Kun varapalvelin huomaa pääkoneen kadonneen, se lähettää stonith-laitteelle sammutussignaalin, ja laite katkaisee virran pääkoneelta. Näin varakone voi ottaa resurssit haltuunsa ilman, että mahdollisesti päällä oleva pääkone alkaa sitä häiritä.

### 2.3.5 Klusterin toiminnan valvonta

Kun klusterissa tapahtuu vikatilanne, ei riitä, että tehtävät siirtyvät klusterin seuraavan koneen hoidettaviksi. Vioista on mentävä tietä eteenpäin, jotta ne voidaan korjata ja klusterin varmuus sekä täysi toimintakyky palauttaa.

Korkean käytettävyyden klusterit ovat luotettava ratkaisu, mutta missään tekniikassa ei ole pelkkiä hyviä puolia vaan pitää ymmärtää myös ratkaisujen heikkoudet. Korkean käytettävyyden klusteri rakentuu päällekkäisistä osista, jotka korvaavat toisia häiriöiden sattuessa. Kun jokin osa on mennyt rikki tai ohjelmisto on jumiutunut, niin palvelun toiminta voi olla enää yksittäisen osan varassa ja tällaisesta tilanteesta pitää saada tieto ylläpidolle tilanteen korjaamiseksi. Valvonta on oleellinen osa klusterin toiminnan varmistamista ja huoltotilanteita varten pitää olla testatut suunnitelmat, jotta ne sujuisivat ilman ongelmia ja yllättäen vastaantulevia tilanteita.

Valvontatyökalut tarjoavat myös paljon tietoa klusterin kuormasta ja sen hetkisestä tilasta, mikä voi auttaa ongelmissa, jotka eivät välttämättä estä palvelua mutta hidastavat sitä. Valvontatyökalujen tallentamaa käyttöhistoriatietoa voi käyttää hyödyksi monella tapaa esimerkiksi huoltokatkosten ajoittamisessa hiljaiselle hetkelle.

### 2.3.6 Klusterin etähallinta

Klusterissa voi olla paljon koneita tai koneet voivat sijaita paikassa, jossa kulkeminen ei ole esimerkiksi turvallisuussyistä tavallista. Koneissa ei yleensä ole edes näppäimistöä tai näyttöä ja tästä syystä niitä on parempi hallita verkon yli. Ylläpito voi olla keskitetty jonnekin muualle kuin klusterin välittömään läheisyyteen ja vain laiterikkojen tapauksessa vaaditaan paikalla käyntiä.

Etähallinta on mahdollista tehdä niin päätteen kautta tai graafisesti. Kumpikin tapa käyttää SSH-protokollaa, joten etähallinnan käyttö on turvallinen tapa hallita testiympäristöä ja tuotantokäytössä olevia klustereita. Tekstipohjainen etähallinta on hyvä ja varma vaihtoehto nykypäivänäkin ja etenkin hitaiden yhteyksien yli mutta nykypäivän nopeilla yhteyksillä myös graafinen etähallinta on hyvin toimiva ratkaisu. Käyttökohde ratkaisee, kumpaa tapaa on parempi käyttää. Klusterien koneilla ei välttämättä edes ole graafista ympäristöä asennettuna, joten tekstipohjainen hallinta on niille parempi. Tilanne on toinen, jos esimerkiksi valvontakoneeseen tai muuhun koneeseen, jossa on käytössä graafinen ympäristö, tarvitsee saada yhteys paikasta riippumatta.

### 3 OHJELMISTOT

Tässä työssä pyrittiin huomioimaan helppo toistettavuus ja testattavuus. Tästä syystä käytettiin ilmaisia ohjelmistoja aina, kun se oli mahdollista. Tässä luvussa esitellyt ohjelmat eivät ole ainoita tai aina käyttötarkoitukseensa parhaita mutta ne ovat jokaisen helposti saatavilla ja niitä käytetään tärkeissäkin palveluissa - osaa hyvin laajasti ja hyvin kriittisissä palveluissa.

Muita olemassa olevia ja käyttökelpoisia vaihtoehtoja löytyy helposti muun muassa tämän työn viitteinä olevista linkeistä sekä kirjallisuuden lähteistä.

#### 3.1 Ubuntu

Ubuntu on Debianiin pohjautuva ja suosiota saavuttanut Linux-jakelu. Se on suunnattu työpöytäkäyttöön ja päivittyy kaksi kertaa vuodessa ja jokaista versiota tuetaan 18 kuukautta julkaisuhetkestä. Siitä on olemassa myös palvelimille suunnattu versio, jossa ei ole lainkaan graafista käyttöliittymää ja myös pitkän tuen eli long term support (LTS) -versio, jota tuetaan palvelimissa viisi vuotta julkaisuhetkestä eteenpäin. Työpöytäversion tuki on LTS-versiolla kolme vuotta. Tässä työssä käytetään pidempään tuettua työpöydälle suunnattua LTS-versiota.

Käyttäjöpohjan ollessa laaja tukea löytyy laajalti ja jakelulla on toimivat käyttäjäfoorumit niin englanniksi kuin suomeksi.

#### 3.2 Ldirectord

Ldirectord on oikeiden palvelimien tarkkailuun ja hallintaan tehty ohjelmisto. Klusteri ympäristössä se vastaa taulukosta, jossa on listattuna toimivat oikeat palvelimet. Tämän taulukon perusteella ohjaajakonepari reitittää asiakkailta tulevia palvelupyynnöitä oikeille palvelimille. Ohjelma kysyy tietyn väliajoin oikeilta palvelimilta tiettyä tiedostoa ja odottaa tietynlaista vastausta. Mikäli vastaus on oikea, oikea palvelin pysyy tai lisätään listalle. Mikäli vastaus on väärä tai sitä ei tule ollenkaan, oikea palvelin poistetaan listalta. Ohjelmalla ei ole graafista käyttöliittymää ja sen asetukset ovat yhdessä tiedostossa. Ldirectord tarvitsee toimiakseen ipvsadm-nimisen apuohjelman.

### 3.3 HA eli High Availability

The Linux High Availability Projectin tarkoituksena on sivuston [5] omien sanojen mukaan vapaasti suomennettuna seuraava:

”Tarjota Linuxille korkean käytettävyyden klusterointiratkaisua, joka korostaa luotettavuutta, saatavuutta ja huollettavuutta yhteisön kehityssponsorien avulla.”

Vaikka projektin nimi viittaa Linuxiin, niin ohjelman kerrotaan toimivan monilla muillakin järjestelmillä kuten FreeBSD, Solaris, OpenBSD ja MacOS/X. Tällä ohjelmistolla voidaan toteuttaa heartbeat-linja, jolla koneet valvovat toistensa kuntoa.

Ominaisuuslista on kattava. Ohjelmiston toisesta versiosta lähtien tuetaan jopa 16 konetta kun ensimmäisessä versiossa tuki oli vain kahdelle koneelle. Suurempi tuettujen koneiden määrä mahdollistaa ohjelmiston käytön laajemmissa ratkaisuissa. Sivuston mukaan ohjelmiston käyttöönotto parantaa klusterin luotettavuutta noin yhdellä yhdeksiköllä, mikä on hyvä parannus. Ohjelmiston luvataan tunnistavan toisen koneen vioittuminen parhaimmillaan alle sekunnissa ja myös erillisiä stonith-laitteita tuetaan. Ohjelmiston laitteistolle aiheuttama kuorma on pieni mutta kasvaa jonkin verran, kun tavoitellaan alle sekunnin vasteaikoja vian tunnistamisessa. Hallintaan, valvontaan ja asetusten tekemiseen ohjelmiston mukana tulevat graafiset työkalut.

### 3.4 LVS eli Linux Virtual Server

LVS-klusteri koostuu kuormantasaajasta, jossa on käytössä IPVS ja klusterin jäsenkoneista. IPVS eli IP Virtual Server on kokoelma ohjelmistoja, jotka ovat mukana Linuxin ytimessä mutta ne eivät ole oletuksena päällä ja ne pitää erikseen ottaa käyttöön. Projektin kotisivuilla [9] projektin käyttökohteiksi mainitaan muun muassa hyvin skaalautuvat ja saavutettavat palvelut, jotka tarvitsevat kuormantasaamista.

Kuorman tasaus toimii siten, että ohjaajakone jakaa klusterin jäsenille asiakasyhteyksiä esimerkiksi sen mukaan, miten kuormitettuja ne jo ovat. Tällöin kuorma jakautuu tasaisesti, eikä yksittäinen palvelin pääse ylikuormittumaan. Kuorma voidaan jakaa monella eri tavalla. Yleisiä tapoja jakaa liikenne jäsenkoneiden välillä on yksinkertainen vuorottelu tai palvelinten mahdolliseen suorituskykyeroon perustuvalla kaavalla.

Skaalautuvuus saavutetaan palvelinkoneiden sujuvan lisäämisen ja poistamisen kautta, sillä tällaiset toimenpiteet voi tehdä järjestelmän ollessa päällä ja tuotantokäytössä. Tarvittaessa se onnistuu automaattisesti.

Palvelun korkea käytettävyys voidaan toteuttaa varmistamalla ohjaajakone varakoneella ja ottamalla käyttöön HA-ohjelmisto. Tällöin laiterikko ei pysäytä koko järjestelmää, ja klusterin jäsenkoneet ovat saavutettavissa.

LVS-klusterin toiminnan kannalta on oleellista erottaa koneiden käyttämät IP-osoitteet.

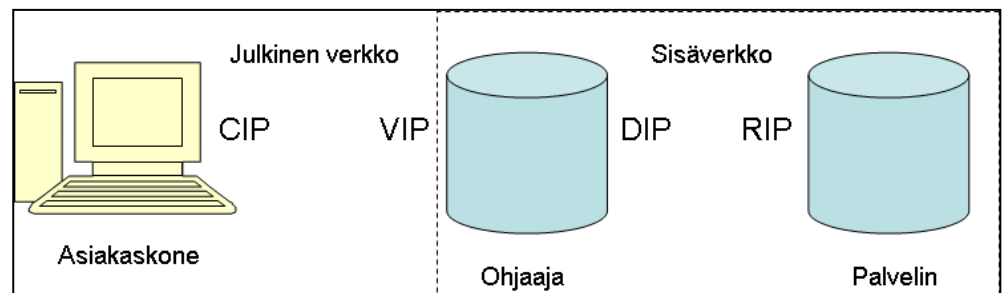
**CIP** eli Client IP on asiakaskoneen IP-osoite.

**VIP** eli Virtual IP on ohjaajakoneen ulospäin näkyvä osoite ja oleellinen, koska se on osoite mihin palvelimelle tulevat yhteydet ohjautuvat.

**DIP** eli Director IP on ohjaajakoneen oikea IP-osoite, jota käytetään reitittämiseen sisäisessä verkossa.

**RIP** eli Real IP on palvelimen IP-osoite sisäisessä verkossa.

Kuvassa 4 on selvennetty, mikä osoite on sisäisessä ja mikä julkisessa verkossa.

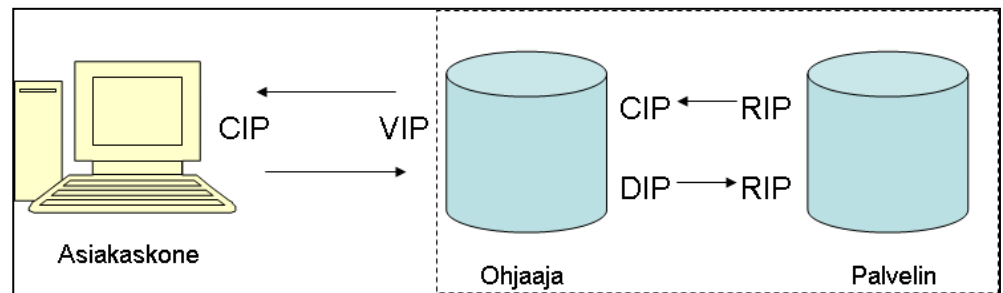


*Kuva 4. LVS-järjestelmän osat ja niiden osoitteiden nimet.*

LVS-klusterissa liikenne on mahdollista reitittää kolmella eri tavalla. Eri reititystapoja voidaan käyttää samaan aikaan saman klusterin sisällä [3, s. 196] mutta selkeyden vuoksi alla on kerrottu jokaisesta reititystyyppistä erikseen omana kokonaisuutenaan.

*LVS - Network Address Translation (LVS - NAT)*

Kun LVS asetetaan NAT-tilaan, käytetään hyväksi verkko-osoitteen muunnosta. Kuvasta 5 nähdään mitä tämä tarkoittaa, kun seurataan paketin kulkua.



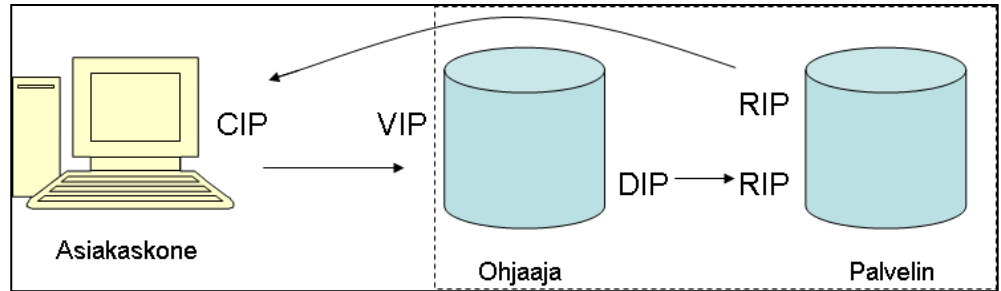
Kuva 5. LVS - NAT -viestikaavio.

Pyynnön lähtiessä asiakkaalta lähettäjänä on oma osoite (CIP) ja vastaanottajana on klusterin julkinen osoite (VIP). Ohjaajakone ohjaa pyynnön palvelimen sisäiseen osoitteeseen (RIP) omalla osoitteellaan (DIP), ja palvelin lähettää vastauksen omalla osoitteellaan asiakaskoneen osoitteeseen (CIP) mutta ohjaajakoneen kautta. Ohjaajakone muuttaa vastauksen osoitetta siten että lähettäjäksi tulee palvelimen julkinen osoite (VIP). Koska osoite, mihin pyyntö lähtee, ja osoite, mistä vastaus tulee, on sama, klusteri näkyy asiakaskoneelle yhtenä koneena.

NAT-tilassa tulee ottaa huomioon, että ohjaajan ja palvelimien tulee olla samassa verkossa. Palvelimilla voi olla käytössä mikä vain käyttöjärjestelmä. LVS-NAT-klusteri on helpoin rakentaa mutta haittapuolena kaikki liikenne kulkee ohjaajakoneen läpi ja ohjaajakone saattaa jossain vaiheessa muodostua klusterin pullonkaulaksi [6, s. 196 - 197].

#### *LVS - Direct Routing (LVS - DR)*

DR-tilassa klusteri ottaa pyynnöt vastaan edelleen ohjaajan kautta, ja pyynnöt välitetään palvelimille käyttäen osoitteena VIP-osoitetta. Vastaus palvelimelta lähtee kuitenkin suoraan asiakkaalle, ja lähettäjän osoitteen on VIP-osoite. Tässäkin tilanteessa asiakkaalle palvelu näkyy yhtenä koneena, koska lähetys ja vastausosoite ovat samat.

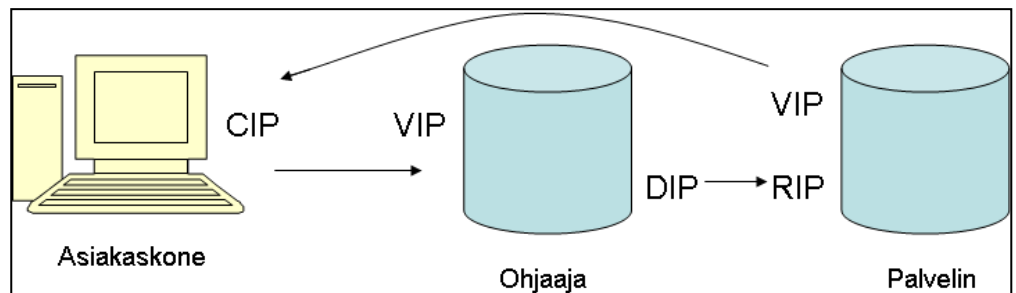


Kuva 6. LVS - DR -viestikaavio.

DR-tilassa ohjaajakoneen teho ei muodostu niin helposti pullonkaulaksi, koska palvelimet vastaavat suoraan asiakkaalle keventäen näin ohjaajakoneen kuormaa. DR-tilassa palvelimia voi myös olla enemmän käytössä kuin NAT-tilassa.

#### LVS - Tunneling (LVS - TUN)

Tunneling eli tunnelointitila eroaa NAT- ja DR-tiloista siten, että ohjaajakoneen ja palvelimen ei tarvitse olla samassa verkossa toimiakseen yhteen, vaan ne voivat olla vaikka maapallon vastakkaisilla puolilla eri verkoissa. Sekä palvelimella että ohjaajalla tulee olla julkinen osoite. Kuvasta 7 nähdään, miten pyyntö lähtee asiakkaalta ja menee ohjaajakoneelle. Ohjaaja pakotoi asiakkaalta tulleen pyynnön uuteen pakettiin ja lähettää sen palvelimelle sen julkiseen osoitteeseen. Palvelin vastaa suoraan asiakkaalle. Kuvasta huomataan, että sisäistä ja ulkoista verkkoa ei tässä tapauksessa ole, vaan sekä ohjaaja että palvelin toimivat julkisessa verkossa.



Kuva 7. LVS - TUN -viestikaavio.

Tunnelointitilaa voi käyttää korkean käytettävyyden klustereissa mutta sellaista nähdään harvoin. Pitkät yhteydet ja mahdolliset reititysongelmat ohjaajakoneen ja jäsenkoneiden välillä ovat hankalia ja laskevat



luotettavuutta. Koska reititys ei ole niin hyvin klusterin ylläpidon käsissä, yllättävien tilanteiden hallinta on vaikeampaa.

### 3.5 VMware Player ja virtualisointi

VMware tarjoaa monipuolisia työkaluja virtualisoinnin eri tarpeisiin aina yksityiskäyttäjistä suurtietokoneisiin [10]. Pienempään tarpeeseen sopiva ohjelmisto on VMware Player, joka on ilmainen ja tukee useita eri isäntä- ja vieraskäyttöjärjestelmiä. Isäntäkäyttöjärjestelmänä eli käyttöjärjestelmänä, johon Player asennetaan, voi olla niin Windows- kuin Linux-konekin. Virtuaalikoneeseen voi asentaa käyttöjärjestelmän vielä laajemmasta valikoimasta. Kotisivujen mukaan tuetaan ainakin Windowsin ja Linuxin eri versioita sekä Solarista niin 32- kuin 64 -bittisinä versioina, joten Player on todella kätevä vaihtoehto eri käyttöjärjestelmien ja asetusten turvalliseen testaamiseen.

Playerin käyttö on yksinkertaista mutta sillä ei voi tehdä itse virtuaalikoneita, vaan ne täytyy tehdä muilla keinoin. VMwaren tuotteista mm. Server versiolla voi tehdä virtuaalikoneita mutta toista virtualisointituotetta ei haluttu asentaa samaan koneeseen. Virtuaalikoneita voi tehdä myös Internetissä ja yksi hyvä palvelu tähän tarkoitukseen on Easyvmx [11]. Palvelu tarjoaa kolme vaihtoehtoista ja yksinkertaista tapaa tehdä virtuaalikoneen. Eroina eri tavoilla on lähinnä se, miten paljon käyttäjä haluaa puuttua koneen kokoonpanoon ja laitteiston asetuksiin.

#### *Virtualisointi*

Virtualisointi on nopeasti suosiota kasvattava teknologia. Sillä saavutetaan monia etuja eri käyttötarkoituksissa niin palvelinhuoneessa kuin käyttäjien työpöydilläkin.

Palvelinten virtualisoinnissa saavutettavia etuja on muun muassa laitteiston koko kapasiteetin saaminen käyttöön, kun palvelut, jotka aiemmin ajettiin omassa koneessa mutta eivät kuitenkaan käyttäneet kaikkia koneen resursseja hyväksi, voidaan ajaa yhdessä fyysisessä koneessa. Tällainen järjestely voi tuoda huomattavia säästöjä laitteisto- ja käyttökuluissa.

Työpöydällä virtualisointi tarjoaa turvallisen ympäristön uusien käyttöjärjestelmien testaamiseen ja uusien asioiden testaamiseen vanhoilla

käyttöjärjestelmillä ilman vaaraa laiterikoista tai koneen sekaisin menemisestä. Näin vain toimiviksi todetut uudet asiat on turvallista ottaa käyttöön käytössä olevassa käyttöjärjestelmässä. Virtuaalinen kone on myös kätevä tarvittaessa siirtää toiselle fyysiselle koneelle, sillä riittää, kun tarvittavat tiedostot siirretään uuteen koneeseen, jossa on yhteensopiva alusta valmiiksi asennettuna.

Virtualisointi toteutettiin alkuaikoina kokonaan ohjelmistopohjaisesti mutta tekniikan saadessa suosiota laitteistovalmistajat ovat alkaneet tuoda myös laitteistopohjaista tukea sen toteuttamiselle. Esimerkiksi prosessorivalmistajilta niin AMD:ltä [7] kuin Inteliltäkin [8] löytyy jo tuki virtualisoinnille.

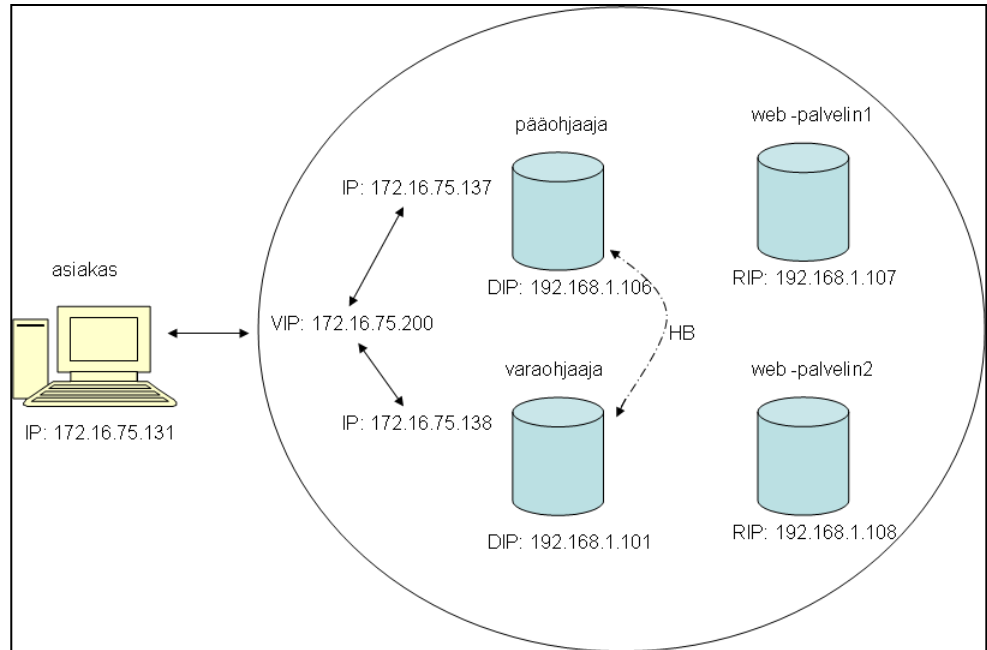
Virtualisoinnin voi tehdä vain jollekin järjestelmän osalle tai koko laitteistolle, ja käyttötarkoitus ratkaisee, mikä on paras vaihtoehto. Tässä työssä käytetään virtualisointivaihtoehtoa, jossa virtualisoidaan kokonaisia x86-koneita.

### **3.6 NoMachine**

NoMachine tarjoaa ilmaisen palvelimen ja asiakasohjelmiston etähallinnan toteuttamiseen Linux-ympäristössä. Windowsille on saatavilla ilmaiseksi vain asiakasohjelmisto. Ohjelman kotisivuilta [14] on saatavilla asennuspaketit, ohjeita ja tukea tuotteen käyttämiseen. Palvelinosa on ilmainen Linuxille, mutta sitä on hieman rajoitettu ja siihen voi olla yhteydessä vain kaksi käyttäjää kerrallaan.

## **4 TESTIKLUSTERIN TOTEUTUS**

Testiympäristö päädyttiin pystyttämään Linux-alustalle, koska alustan käyttö oli tuttua, alusta on vakaaksi todettu, hyväksi todettu virtualisointiohjelmisto ja myös etäkäyttötyökalu löytyy samalle alustalle. Virtuaalikoneita tehtiin kuusi kappaletta. Kahdesta koneesta tehtiin testattavan klusterin ohjaajapari ja kahdesta koneesta tehtiin palvelinpari ohjaajaparin taakse. Yhdestä koneesta tehtiin asiakas ja yksi kone jäi varastoon odottamaan testien aikana ilmitulevaa käyttötarvetta varten.



Kuva 8. Testiklusterin kaavio.

Kuvassa 8 on testiklusterin kaavio. Eroavaisuuksia aiemmin esitettyyn klusterin kuvaan on käytännön syistä. Tässä oli tarkoituksena testata klusterin ohjelmisto-osaa testiympäristön sallimissa puitteissa, joten verkkolaitteita tai virransyötön varmistuksia ei kaaviossa ole.

Web-palvelimilla on vain sisäisen verkon osoitteet, joilla ne ovat yhteydessä ohjaajakoneiden sisäisessä verkossa olevaan osoitteeseen. Ohjaajakoneilla on ulospäin näkyvä oma verkko, jonka kautta ne näkyvät asiakkaalle. Heartbeat-linja toteutettiin kolmannen verkkokortin puutteen takia palvelimien sisäisessä verkossa kulkevalla signaalilla. Liikenne verkossa ei ole niin suurta, että heartbeat-signaalin oletettiin häiriintyvän. Oikeassa käytössä tällainen ratkaisu ei ole mitenkään perusteltua.

#### 4.1 Testialustan asennus ja asetukset

Tässä luvussa tarkastellaan virtuaalisen testialustan pystyttämistä ja etähallinnan asentamista. Näillä kahdella komponentilla alustaa pystyi käyttämään kätevästi mistä vain, missä oli verkkoyhteys saatavilla.

##### 4.1.1 Isäntäkone

Isäntäkoneen käyttöjärjestelmäksi valittiin Ubuntu Linux 8.04 LTS ja virtualisointiin Vmware Player. Myös itse testiklusteri päätettiin pystyttää Ubuntu Linuxin päälle. Ubuntu LTS -versiota tuetaan normaalia versiota

pidempään, ja se on sen takia parempi valinta ympäristöihin, joiden vaatimukset eivät muutu niin nopeasti ja joiden tehtävät pysyvät samana pidemmän aikaa.

Isäntäkoneen laitteisto oli suhteelliseen uusi ja sen arveltiin suoriutuvan tehtävästään hyvin. Koneen kokoonpano oli seuraava:

- suoritin: Athlon 64 X2 5200+ (2,70 GHz, kaksi ydintä)
- keskusmuisti: 3 Gt
- kiintolevy: 500 Gt

Kriittisin isäntäkoneen resurssi on muisti, jota testikoneessa oli 3 Gt. Jokainen virtuaalikone käytti 320 Mt. Mikäli kaikki kuusi tehtyä konetta otettaisiin käyttöön, muistintarpeeksi tulisi pelkästään virtuaalijärjestelmien osalta 1,9 Gt. Mikäli isäntäkone joutuu käyttämään levyvälimuistia, se vaikuttaa negatiivisesti niin isäntäkoneen kuin virtuaalikoneidenkin suorituskyvyn. Toiseksi kriittisin isäntäkoneen ominaisuus on suoritinteho, jota tässä tapauksessa oli kaksisyttimisessä suorittimessa 2,7 GHz. Tämän arveltiin riittävän hyvin testikäyttöön mutta rooliin, jossa järjestelmä olisi tuotantokäytössä, koneen arvioitiin olevan alitehoinen. Levytilaa oli testitarkoituksiin enemmän kuin tarvittiin, joten sen ei katsottu muodostuvan esteeksi missään vaiheessa.

#### 4.1.2 Ohjelmistojen asennus isäntäkoneeseen

Isäntäkoneeseen asennettiin ohjelmistot, joiden avulla itse klusteri voitiin toteuttaa virtuaalisesti ja joiden avulla isäntäkoneetta pystyttiin ohjaamaan etänä.

##### *VMware Player*

VMware Playerista ei ole olemassa valmista pakettia Ubuntulle, joten se piti asentaa käsin. Valmistaja on tehnyt asiaa helpottavan komentojonon, joka hoitaa asentamisen, kunhan sille antaa sen tarvitsemat tiedot. VMware tarjoaa rpm-pohjaisille järjestelmille valmiiksi asennettavan paketin.

Playerin asennuspaketin voi ladata valmistajan kotisivuilta [15]. Ennen kuin latauksen voi tehdä, pitää vastata pakolliseen kyselyyn, jossa esitetään tarkentavia kysymyksiä ohjelmiston käyttötarkoituksesta ja -ympäristöstä.

Itse asennus käynnistyy, kun käynnistetään komentojono kansioista, johon ladattu paketti purettiin. Paketin voi purkaa minne tahansa. Komento jona tulee ajaa järjestelmänvalvojana, jotta asennus voi kopioida tarvittavat tiedostot oikeille paikoilleen. Tarvittava komento on:

```
sudo ./vmware-install.pl
```

Yllä oleva komento on kolmiosainen: "sudo" ottaa käyttöön järjestelmänvalvojan oikeudet ja "." käynnistää komentojonon "vmware-install.pl". Tämän jälkeen ohjelma alkaa kysellä tarvitsemiensa tiedostojen ja kansioiden sijaintia. Asennuksen vaiheet komennon suorittamisesta loppuun asti ovat liitteessä 2. Komentojonon kysymät kysymykset on lihavoitu, jotta ne erottuvat paremmin muun tekstin seasta.

Komento jona ehdottaa oletusarvoksi yleensä oikeaa vaihtoehtoa ja asennuksen aikana piti korjata kahta kohtaa, joista toinen oli oikea verkkoliitäntä ja toinen kääntäjän väärä versio. Testikoneessa oli useampi verkkoliitin ja tämän takia oletusarvo eth0 piti korjata eth1:ksi. Toinen kohta, jossa piti poiketa oletusarvoista, oli, kun komento jona havaitsi asennuksen mukana tulleiden moduuleiden olevan epäyhteensopivia asennettuna Linuxin ytimen kanssa ja alkoi kääntää uusia. Asennus käänsi uudet, moduulit mutta antoi varoituksen väärästä kääntäjän versiosta ytimen ja nykyisen kääntäjän suhteen. Asennus suositteli, että väärää kääntäjää ei käytettäisi, mutta asennus suoritettiin loppuun sillä hetkellä asennettuna olleella kääntäjällä.

Vmware Player tutki hetken verkkoasetuksia ja päätti ottaa käyttöön virtuaalisen verkon osoitteesta 172.16.20.0 maskilla 255.255.255.0, koska se ei ollut käytössä asennushetkellä. Kaikki virtuaalikoneet, jotka käynnistetään, saavat osoitteen tästä verkosta ja näkevät täten toisensa. Kaikissa luoduissa virtuaalikoneissa on kaksi verkkoliitäntää, mutta kahta tarvitaan periaatteessa vain ohjaajakoneissa, jotta klusterin tietoliikenneliitännöiden lisäksi voidaan toteuttaa klusterin toiminnalle oleellinen heartbeat-signaali koneiden välille käyttäen lähiverkkotekniikkaa.

### *Virtuaalikoneiden luominen*

Kun ympäristö on asennettu, niin on aika tehdä virtuaalikoneet, joihin testattavat järjestelmät voi asentaa. EasyVMX:n kotisivuilla [11] on kolme eri vaihtoehtoa, joilla virtuaalikoneita voi tehdä. Vaihtoehtoina ovat alkuperäinen

monipuolisen virtuaalikoneen teko, helpon virtuaalikoneen teko ja uusi versio monipuolisen virtuaalikoneen tekemisestä. Viimeisin vaihtoehto on päivitetty versio ensimmäisestä vaihtoehdosta ja sen tekemisessä on otettu huomioon muun muassa Windows Vista -yhteensopivuus ja VMware Playerin uuden 2. version ominaisuuksia. Tekemiseen käytettiin viimeisintä vaihtoehtoa. Jokaisen kohdan vieressä on ohjelaatikko, mikä selkeyttää hyvin, mitä kohdan valinnat tarkoittavat.

Virtuaalikoneiden ominaisuuksista muutettiin seuraavia kohtia:

- Virtual hardware
  - Nimi: vapaavalintainen (ei vaikuta koneen toimintaan)
  - Muisti: 320Mt
  - CPU: 1kpl
- Network Interface Card
  - Ethernet0 ja ethernet1 otettiin kumpikin käyttöön
- Sound and I/O -ports
  - Poistettiin kaikki valinnat, koska niitä ei testeissä tarvita

Seuraavien kohtien annettiin olla oletusasetuksissaan:

- Power ON / Power OFF
- Virtual Machine Description
- Floppy Disk Drive (ei valittuna)
- CDROM Drives (1 valittuna)
- Hard Disk Drives (1kpl, 4.7Gt eli DVD:lle mahtuva)

Klusterin ohjaajakoneille valittiin kiintolevyn kooksi 15Gt, sillä ne tarvitsevat enemmän tilaa, jotta niillä voi kääntää ytimen. Kun asetukset oli valittu, painettiin "Create virtual machine" -nappia ja päästiin lataussivulle, jossa oli latauslinkki alle 5 kt:n kokoiseen zip-pakattuun tiedostoon ja linkin alla näytettiin virtuaalikoneen asetustiedoston sisältö.

Tuntemattomasta syystä monipuoliset virtuaalikoneen tekovaihtoehdot eivät antaneet valita, millä käyttöjärjestelmällä virtuaalikonetta ajateltiin käyttää. Tämä kävi ilmi vasta virheilmoituksesta, kun jo luotuja virtuaalikoneita käynnistettiin. Ongelma ratkesi, kun tehtiin virtuaalikone yksinkertaisella tavalla ja kopioitiin sen .vmx-tiedostosta jokaisen jo luodun virtuaalikoneen .vmx-tiedoston "guest os" kohdan arvoksi "ubuntu". Guest os -kohta oli

kokonaan tyhjä virheellisissä virtuaalikoneissa. Kun korjaukset oli tehty kaikkiin virtuaalikoneisiin, niin asennukset käynnistyivät normaalisti.

### *NoMachine*

NoMachine tarjoaa kotisivuillaan [14] valmiit paketit asennettavaksi sekä deb- että rpm-pohjaisiin järjestelmiin. Sivuilta pitää noutaa kolme eri pakettia; nxclient\_3.2.0-14\_i386.deb, nxnode\_3.2.0-13\_i386.deb ja nxserver\_3.2.0-16\_i386.deb. Palvelinkoneelle vaaditaan kaikki kolme pakettia, mutta jos asennusta tehdään koneelle, miltä on tarkoitus vain ottaa yhteys koneeseen, missä palvelin jo on, voidaan asentaa vain pelkkä client- eli asiakasohjelmisto. Tiedostot voi hakea seuraavalla komennolla vaihtaen yhden tiedoston nimen kerrallaan komennon loppuun:

```
wget http://64.34.161.181/download/3.2.0/Linux/
```

```
nxclient_3.2.0-14_i386.deb
```

```
nxnode_3.2.0-13_i386.deb
```

```
nxserver_3.2.0-16_i386.deb
```

Asennus tulee tehdä järjestyksessä client, node ja viimeiseksi server. Paketit voi asentaa kirjoittamalla yhdelle riville seuraava komento:

```
sudo apt-get install nxclient_3.2.0-14_i386.deb  
nxnode_3.2.0-13_i386.deb nxserver_3.2.0-16_i386.deb
```

Ohjelman käynnistävä kuvake löytyy Sovellukset-valikon Internet-kohdasta. Palvelin on asennuksen jälkeen heti käyttövalmis eikä sitä tarvitse säätää. Kun siihen otetaan yhteyttä, olisi hyvä tietää koneen osoite, käyttöjärjestelmä, ikkunointi järjestelmä ja yhteyden nopeus. Näitä kysytään ennen kuin yhteyden voi muodostaa.

### *Ubuntun asentaminen*

Seuraava tehtävä oli käyttöjärjestelmien asentaminen virtuaalikoneisiin. Ubuntun asentaminen levykuvan noutamisesta toimivaan järjestelmään on ohjeistettu liitteessä 1. Asennus on hyvin selkeä prosessi, ja koska Ubuntu asennetaan ainoaksi käyttöjärjestelmäksi, asennus menee läpi lähes kokonaan painamalla aina seuraava-nappia.

## 4.2 Klusterin pystyttäminen

Klusteri, johon asennetaan LVS, vaatii Linuxin ytimeistä valittavaksi asetuksia, joita ei ole valmiiksi valittuna vakio ytimissä. Tämän takia Linuxin ydin eli kernel pitää kääntää erikseen tätä tehtävää varten. Ydin tarvitsee kääntää vain klusterin ohjaajakoneisiin, joille tulee virtuaalinen IP-osoite. Muilla koneilla voidaan käyttää asennuksen mukana tulevaa vakioydintä.

Klusteri päädyttiin pystyttämään LVS-DR-mallin mukaiseksi. Koska LV-DR-klusteri reitittää liikenteen palvelimelta suoraan asiakkaalle, ilman että liikenteen täytyy kulkea ohjaajan kautta, saavutetaan parempi kuormansietokyky verrattuna LVS-NAT-malliin.

### 4.2.1 Ytimen kääntäminen ja asentaminen

Ytimen versiossa 2.4.28 ja sitä aiemmissa ei ole tukea IPVS-järjestelmälle. Mikäli ollaan käyttämässä edellä mainittua tai vanhempaa ydintä, ydin tulee ensin paikata ja kääntää vasta sitten. 2.6-sarjan ytimissä tuki löytyy valmiiksi ja niissä riittää, kun ominaisuus valitaan asennettavaksi ennen kuin ydin käännetään.

Ytimen kääntämiseen tarvitaan kääntäjä, joka Ubuntussa on gcc eli The GNU Compiler Collection sekä ytimen lähdekoodit [16]. Kääntäjä tulee vakioasennuksen mukana ja eikä sitä tarvitse erikseen asentaa. Lähdekoodipaketin voi hakea kätevästi wget-ohjelmalla.

Ytimeistä on olemassa kaksi eri haaraa, jotka kannattaa tunnistaa. Käyttöön tarkoitetuissa ytimissä toinen numero on aina parillinen (2.6.x) ja testikäyttöön tarkoitetuissa ytimissä pariton (2.5.x). Testikäytössä olevissa ytimissä testataan eri asioita ja niitä ei suositella muuhun käyttöön. Käyttöön tarkoitetuissakin ytimissä on vakaita versioita ja kehitysversioita, jotka erottaa rcx-liitteestä nimen perässä. Lyhenne tulee sanoista Release Candidate #, missä # on numero, mikä kertoo, monesko testijulkaisu on menossa. Vakaan version nimessä ei ole mitään liitteitä. Ytimen versio 2.6.27.5 oli kirjoitushetkellä uusin ja suositeltavaa on, että käytetään aina uusinta vakaata versiota.

Ytimen lähdekoodit haetaan /tmp -kansioon, kun annetaan seuraavat komennot:



```
cd /tmp
```

```
wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.27.5.tar.bz2
```

Ensimmäinen komento "cd" vaihtaa hakemistoon "/tmp" ja toinen komento "wget" noutaa tiedoston 2.6.27.5.tar.bz2 verkkokansioista "http://www.kernel.org/pub/linux/kernel/v2.6/". Wget noutaa tiedoston siihen kansioon missä ollaan komennon antamishetkellä. Haettu paketti puretaan seuraavalla komennolla "/usr/src" -hakemistoon:

```
sudo tar -xjvf linux-2.6.27.5.tar.bz2 -C /usr/src
```

Edellinen komento tulee suorittaa järjestelmänvalvojan oikeuksin (sudo), sillä kohdekansioon ei ole tavallisella käyttäjällä oikeuksia. Kohdekansiossa pitäisi nyt olla "linux-2.6.27.5" -niminen kansio ja sinne siirryttiin komennolla:

```
cd /usr/src/linux-2.6.27.5
```

Tässä vaiheessa tuli tehdä ytimen kääntämisasetukset, jotka voi tehdä graafisesti tai päätteessä. Ennen kuin kumpikaan toimi, piti asentaa kehityskirjastoja, joita ei tullut mukana vakio asennuksessa. Päätteessä komennolla

```
sudo make menuconfig
```

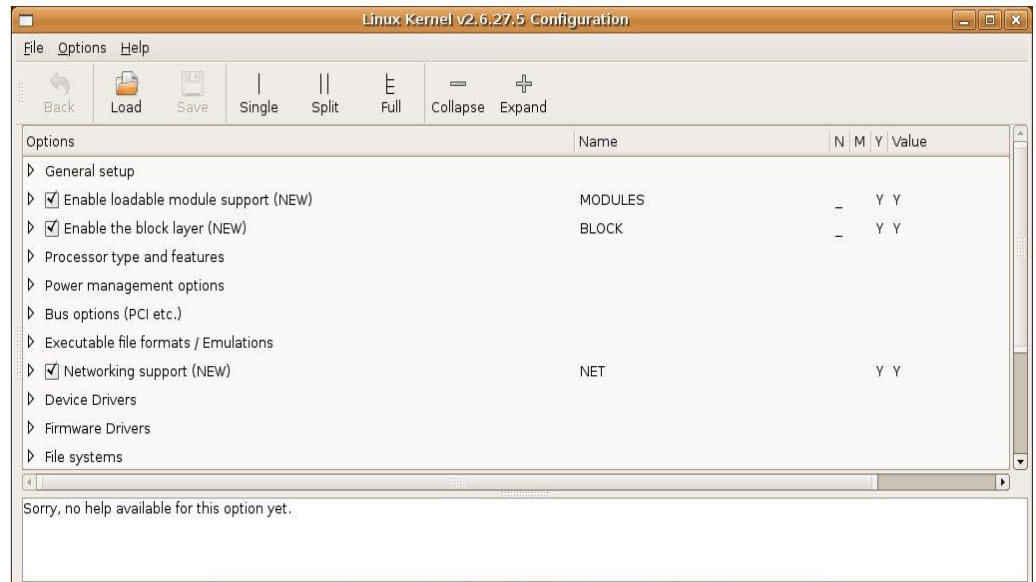
ajettava asetusohjelma ei toiminut ilman libncurses5-dev-kirjastoa ja päätteessä käynnistettävä mutta graafisesti toimiva asetusohjelma vaati libglade2.0-dev-kirjaston asentamista riippuvuuksineen. Kirjastot voi asentaa päätteeltä komentamalla:

```
sudo apt-get install libncurses5-dev libglade2.0-dev
```

Asentaminen onnistuu myös Synaptic-ohjelman kautta graafisesti. Jos kirjastot eivät ole asennettuna ja asetusohjelmaa yritetään käynnistää, tuloksena on sekava virheilmoitus, jossa ei suoraan sanota, mitä puuttuu. Kun kirjastot ovat asennettuina, asetusohjelmat toimivat ja asennuksessa päästiin asetusten tekoon. Graafinen ja selkeämpi asetusohjelma käynnistettiin komennolla:

```
sudo make gconfig
```

Komennon tuloksena avautui kuvassa 9 näkyvä ikkuna, jossa näkyy gconfig-ohjelman aloitusikkuna. Käyttöliittymä on yksinkertainen. Collapse- ja Expand-napit pienentävät tai laajentavat valitun puun haaroja. Single-, Split- ja Full-napit muuttavat tiedon esitystapaa.

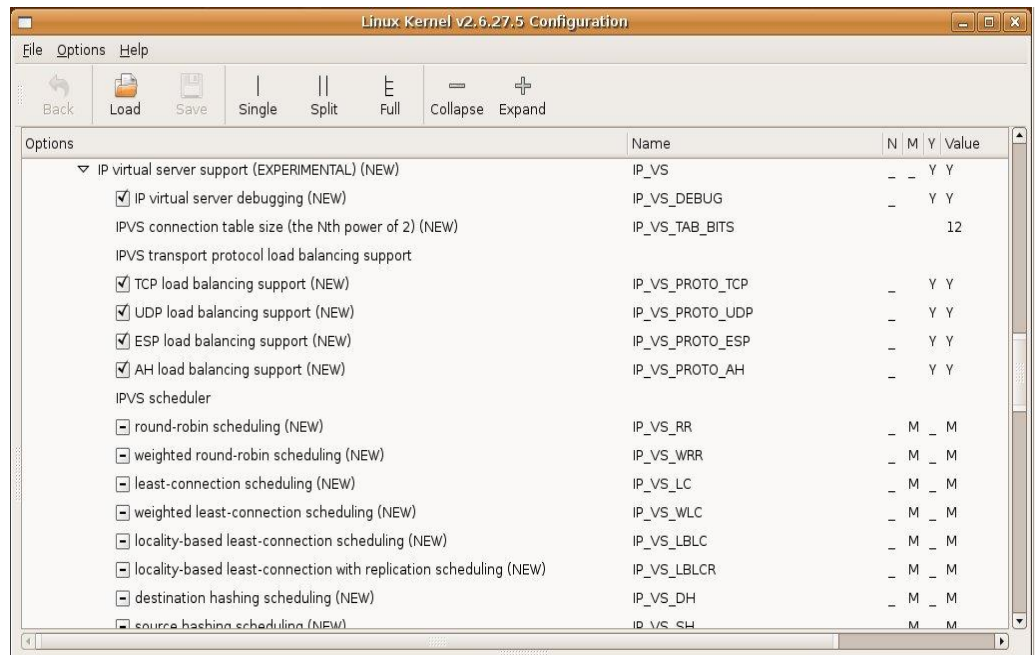


Kuva 9. Gconfigin aloitusikkuna.

Vaihtoehto, joka pitäisi saada kääntymään ytimeen, löytyy seuraavan polun takaa:

```
Networking support -> Network options -> TCP/IP network-
ing -> IP Virtual Server Support
```

Kuvassa 10 näkyy kohta etsittynä ja avattuna. IPVS valitaan kääntymään painamalla sen riviltä Y:n kohdalta kerran hiirellä. Tällöin se kääntyy suoraan ytimeen.



Kuva 10. IPVS valittuna ja tallennettuna.

Tämän jälkeen tulee painaa Save-nappia, jolloin asetustiedosto tallentuu levyllä muutoksineen. Ydin on nyt valmis käännettäväksi. Kääntäminen aloitetaan komennolla:

```
make
```

Kääntäminen kestää aika pitkään. Käännöksen aikana ei tullut yhtään virhettä mutta 6 varoitusta. Varoitukset eivät suoraan viitanneet minkään menevän pieleen, joten asennusta päätettiin virheistä huolimatta kokeilla nyt käännetyllä ytimellä. Kun kääntäminen on valmis, voidaan kääntää moduulit komennolla:

```
make modules
```

Moduulit kääntyvät huomattavasti nopeammin kuin itse ydin, eikä niiden kääntymisen aikana tullut varoituksia. Kun ytimen moduulit ovat kääntyneet, voidaan asentaa ensin moduulit ja sitten itse ydin antamalla komennot:

```
sudo make modules_install
```

ja

```
sudo make install
```

Komento asentaa kolme tiedostoa /boot-hakemistoon ja muokkaa Grubin, eli Linuxin käynnistyslataajan asetustiedostoa. Asennetut tiedostot ovat:

- config-2.6.27.5
- System.map-2.6.27.5
- vmlinuz-2.6.27.5

Viimeisenä on jäljellä käynnistyslataajan asetusten tarkistaminen. Mikäli asennus ei valmiiksi ollut lisännyt uusia asetuksia käynnistyslataajaan, asetusten päivittäminen onnistuu automaattisesti komennolla:

```
sudo update-grub
```

Komento etsii asennetut ytimet ja asettaa ne asetustiedostoon, uusin oletusvaihtoehdoksi. Jos kuitenkin tarvitsee käsitellä asetustiedostoa käsin tai haluaa tutkia sen sisältöä, sen voi avata editoriin komennolla:

```
sudo gedit /boot/grub/menu.lst
```

Grubin käynnistysvalikko on oletuksena pois päältä, kun käytössä on vain yksi käyttöjärjestelmä. Grubin asetuksia päätettiin muuttaa siten, että valikko näkyy kymmenen sekuntia ennen kuin Grub käynnistää oletuskäyttöjärjestelmän. Jotta toivottu käytös saadaan aikaiseksi, tulee asetustiedostosta muuttaa kahta riviä. Ensimmäinen rivi on "timeout 3" ja toinen on "hiddenmenu". Timeout-muuttujan arvo asetetaan arvoon 10 ja hiddenmenu-rivin eteen laitetaan #-merkki. Tämä tarkoittaa että rivi on kommentoitu eikä sitä oteta enää huomioon tiedostoa luettaessa. Kun valikko on näkyvillä, siitä on ongelmatilanteissa helppo käynnistää vanhempi toimiva kokoonpano ongelman korjaamiseksi. Tiedosto pitää muistaa tallentaa kun se suljetaan, jotta asetukset tallentuvat.

Uusi ydin oli nyt valmiina käyttöön otettavaksi ja se tehtiin käynnistämällä järjestelmä uudelleen komennolla:

```
sudo reboot
```

Järjestelmä käynnistyi käynnistyslataajaan asti normaalisti, mutta uutta ydintä käynnistettäessä tuli seuraava virheilmoitus:

```
"[ 1.176113] Kernel Panic - not syncing: VFS: Unable to mount root fs on unknown-block(0,0)"
```

Hetken tutkimisen jälkeen selvisi, että ytimestä puuttuu tarvittava ajuri tiedostojärjestelmän lataamiseen. Ytimen kääntämisen yhteydessä oli jäänyt tekemättä initrd-tiedosto, joka sisältää Linuxin käynnistymiseen välttämättömiä ajureita. Näihin kuuluvat muun muassa tiedostojärjestelmän ajurit.

Ytimen versiosta 2.6 eteenpäin initrd-tiedoston voi tehdä update-initramfs nimisellä ohjelmalla. Komento tulee ajaa /boot-hakemistossa järjestelmänvalvojan oikeuksin. Komento määreineen on seuraavanlainen:

```
sudo update-initramfs -ck 2.6.27.5
```

Määreellä c luodaan uusi initrd-tiedosto, ja määreellä k määritellään ytimen versio. Tuloksena /boot-hakemistoon on tullut tiedosto nimeltä: initrd.img-2.6.27.5.

Koska tiedosto luotiin vasta nyt, se tulee erikseen lisätä vielä käynnistyslataajaan asetuksiin. Kuvassa 11 näkyvä initrd-alkuinen rivi tulee lisätä kuvassa näkyvälle paikalle. Asetustiedosto tulee tallentaa, jotta asetukset ovat voimassa seuraavan käynnistytksen aikana.

```
title          Ubuntu 8.04.1, kernel 2.6.27.5 Default
root          (hd0,0)
kernel       /boot/vmlinuz root=UUID=3f274bbb-a5f8-49bb-9b23-93f5ce342732 ro quiet splash
initrd       /boot/initrd.img-2.6.27.5
quiet
```

*Kuva 11. Käynnistyslataajaan lisättävä rivi.*

Kun kone nyt käynnistettiin uudelleen, se latasi käyttöjärjestelmän virheetä loppuun asti ja toimi normaalisti.

#### 4.2.2 Ohjaajakoneiden asentaminen ja asetukset

##### *Linux Virtual System*

LVS:n eli IPVS:n asentaminen tapahtui ytimen kääntämisen yhteydessä. Tässä vaiheessa se otetaan käyttöön tekemällä tarvittavat muutokset asetustiedostoihin ja asentamalla ldirectord- ja High Availability -ohjelmat.

##### *High Availability:n asentaminen ja asetukset*

Linux HA -ohjelmisto löytyi Ubuntun pakettienhallinnasta, joten se päätettiin asentaa sieltä. Kun ohjelma asennetaan pakettienhallinnasta, sen hallinta on helpompaa sillä se päivittyy automaattisesti sitä kautta.

Tarvittavat paketit ovat heartbeat-2 ja heartbeat-2-gui. Ensimmäinen on itse ohjelma ja toinen on graafinen käyttöliittymä, joka helpottaa hallintaa ja asetusten tekoa. Paketit asennettiin komennolla:

```
sudo apt-get install heartbeat-2 heartbeat-2-gui
```

Asennus asentaa tässäkin tapauksessa tarvittavat riippuvuudet ja kysyy siihen luvan ennen asentamista.

Jotta ohjelmisto saadaan toimimaan halutulla tavalla, tulee muokata kolme asetustiedostoa. Nämä ovat ha.cf, haresources ja authkeys. Tiedostojen tulee sijaita hakemistossa /etc/ha.d/. Tiedostoista on olemassa esimerkit kansiossa /usr/share/doc/heartbeat/. Tiedostot ha.cf ja haresources on pakattu ja ne tulee purkaa komennoilla

```
sudo gunzip ha.cf.gz ja
sudo gunzip haresources.gz
```

Komennon tuloksena syntyvät tiedostot ha.cf ja haresources. Tämän jälkeen nämä kolme asetustiedostoa pitää kopioida kansioon /etc/ha.d/. Tämä tapahtuu komennolla

```
sudo cp authkeys haresources ha.cf /etc/ha.d/
```

Ensimmäiseksi muokataan tiedostoa ha.cf. Tiedosto avataan muokattavaksi komennolla

```
sudo gedit /etc/ha.d/ha.cf
```

Alla on ne rivit, jotka mallitiedostosta otettiin käyttöön poistamalla kommentit rivien alusta [6, s. 135]:

```
logfile /var/log/ha-log
keepalive 2
deadtime 30
initdead 120
udpport 694
bcast eth1
node palvelin1
node palvelin2
```

Ja lisäksi kommentoidaan alla oleva rivi, koska ylempänä otettiin oma loki käyttöön.

```
logfacility local0
```

Keepalive-arvo kertoo montako sekuntia heartbeat-viestien välillä on. Deadttime-arvo kertoo, kauanko odotetaan, ennen kuin oletetaan toisen koneen olevan epäkunnossa. Initedad-arvo on sekunneissa se käynnistyksen jälkeinen aika mikä odotetaan, ennen kuin oletetaan varakoneen olevan epäkunnossa. Udppport kertoo portin numeron, jossa ohjelmisto toimii. Bcast-rivi kertoo missä verkossa heartbeat-signaali kulkee ja node-rivit kertovat palvelimien nimet, joiden välillä heartbeat-signaali kulkee. Näiden nimien tulee olla juuri ne nimet, jotka tulevat kun koneessa ajetaan komento `uname -n`.

Seuraavaksi muokataan `authkeys`-tiedostoa. Se avataan muokattavaksi komennolla

```
sudo gedit /etc/ha.d/authkeys
```

Tiedoston kaikki rivit on kommentoitu. Kommentit otettiin pois riveiltä `"auth 1"` ja `"2 sha1 HI!"`. Jälkimmäinen rivi muokattiin muotoon `"1 sha1 testisana"`. Rivit määrittelevät käytetyn avaimen ja sen koodaustyyppin. Tämän jälkeen tiedosto tallennettiin. Tiedoston tulee olla vain pääkäyttäjän luettavissa. Tämä saadaan aikaan komennolla

```
sudo chmod 600 /etc/ha.d/authkeys
```

Seuraavaksi avattiin muokattavaksi `haresources`-tiedosto komennolla

```
sudo gedit /etc/ha.d/haresources
```

Tiedoston kaikki rivit ovat kommentoituja. Loppuun lisättiin rivi `"palvelin1 httpd"`. Rivi kertoo, mikä ohjaajakone omistaa normaalisti minkäkin palvelun. Tässä tapauksessa palvelin1-koneella on web-palvelin. Kun rivi on lisätty, tiedosto voidaan tallentaa.

Tiedostojen `ha.cf`, `haresources` ja `authkeys` tulee olla tarkalleen samoja sekä pääkoneella, että varakoneella. Hyvä keino varmistua asiasta, on tehdä

muokkaukset jommallakummalla koneella ja tämän jälkeen kopioida tiedostot toiselle koneelle.

#### *Ldirectord:n asentaminen*

Ldirectord löytyy pakettienhallinnasta nimellä ldirectord, ja se asentuu komennolla

```
sudo apt-get install ldirectord
```

Koska se on tehty perl-kielellä, riippuvuuksina asentuu myös tarvittavat paketit perl-kielisten ohjelmien ajamiseen, joita vakioasennuksen mukana ei tullut.

Ldirectord ohjelma valvoo oikeiden palvelinkoneiden tilaa käyttäen http-protokollaa. Ohjelma tarkkailee palvelimien tilaa ja mikäli tulee uusia koneita, ne otetaan käyttöön ja jos jokin palvelin vioittuu, sille ei enää ohjata asiakasyhteyksiä. Ldirectord käyttää palvelimien lisäämiseen ja poistamiseen ohjelmaa nimeltä ipvsadm. Ohjelman käyttö on automaattista eikä vaadi käyttäjän toimenpiteitä. Ohjelman asennetaan pakettienhallinnasta komennolla

```
sudo apt-get install ipvsadm
```

#### *Ldirectord:n asetukset*

Ldirectord:n asetustiedoston tulee sijaita kansiossa /etc/ha.d/conf. Tiedostoa ei oletuksena ole olemassa, vaan se pitää sinne luoda. Tämä tapahtuu komennolla

```
sudo gedit /etc/ha.d/conf/ld.conf
```

Tiedoston nimi voi olla mikä vain, mutta sijainnin tulee olla yllä määrätty. Kun tiedostoa on muokattu, se tulee tallentaa. Alla on esimerkki asetustiedoston sisällöstä [6, s. 267]:

```
checktimeout=20
checkinterval=5
autoreload=yes
quiescent=no
logfile="info"
virtual=172.16.75.200:80
```



```

    real=192.168.1.107:80 gate 1
".healthcheck.html", "OK"
    real=192.168.1.108:80 gate 1
".healthcheck.html", "OK"
    service=http
    checkport=80
    protocol=tcp
    scheduler=wrr
    checktype=negotiate
    fallback=127.0.0.1

```

Asetustiedoston ensimmäinen rivi kertoo sekunneissa, kuinka kauan vastausta odotetaan. Toinen rivi kertoo kuinka tiheään tarkistusviestejä lähetetään. Autoreload-arvolla määrätään ldirectord tarkistamaan, onko asetustiedosto muuttunut. Jos muutoksia on, ne otetaan automaattisesti käyttöön. Quiescentin arvolla "no", ldirectord määrätään poistamaan vastaamatta jättänyt palvelin IPVS-taulukosta. Logfile-arvolla asetetaan ohjelma kirjoittamaan lokia "info" tasolla. Virtual-rivi kertoo virtuaaliosoitteen ja käytettävän portin. Real rivit kertovat tarvittavat tiedot jokaisesta oikeasta palvelimesta. Rivin rakenne on seuraavanlainen: ip-osoite ja portti, LVS-tila, painotus, pyynnön kohde ja odotettu vastaus. LVS-DR-tilan vaihtoehto on "gate". Painotuksella voidaan jakaa yhteyksiä koneille niiden tehokkuuden mukaan. Mitä enemmän painoa annetaan, sitä enemmän yhteyksiä koneelle ohjataan. Service-kohta määrää mitä palvelua käytetään ja port määrää mitä porttia käytetään palvelun tavoitteluun. Protocol-asetuksella määrätään ohjelma käyttämään tcp-protokollaa. Scheduler-asetus kertoo, millaista aikataulutajaa käytetään. Wrr tarkoittaa "weighted round robin" aikataulutusta, jossa palvelimille jaetaan yhteyksiä vuoronperään, ottaen samalla huomioon niiden suorituskyky. Scheduler asetuksia on useanlaisia [6, s. 202 - 207]. Checktype-arvolla määrätään tapa, jolla ldirectord tarkistaa palvelimien kunnon. Negotiate-arvolla palvelimelle lähetetään pyyntö, johon tulee vastata tietyllä vastauksella. Jos vastaus on se mitä pitää kaikki on hyvin, mutta mikäli se eroaa, niin palvelimen katsotaan olevan epäkunnossa. Fallback-osoite on osoite, jonne yhteydet ohjataan kun niitä ei voida enää ohjata oikeille palvelimille.

#### 4.2.3 Palvelimien asentaminen ja asetukset

Palvelinkoneisiin tulee aika yksinkertainen ohjelmisto, sillä riittää kun niissä on web-palvelin ja "lvsdrns" niminen komentojono-tiedosto.

### *Apache 2:n asentaminen*

Apache2 löytyy pakettienhallinnasta ja se on kätevinä asentaa sitä kautta.

Paketin nimi on apache2 ja se asentuu komennolla:

```
sudo apt-get install apache2
```

Asennus asentaa samalla useita muitakin paketteja, joita apache2:n asennus vaatii. Näiden asentamiseen kysytään lupa ennen asentamista. Asennuksen lopuksi palvelin käynnistetään automaattisesti.

Palvelin vastaa pyyntöihin portissa 80. Tarvittava IP-osoite löytyy kuvasta 8.

Palvelimien tarjoamiksi tiedostoiksi riittää kaksi yksinkertaista html-tiedostoa. Toinen on palvelimen tilan tarkkailuun tarkoitettu .healthcheck.html-tiedosto ja toinen on asiakkaalle näkyvä index.html-tiedosto. Tiedostot tulee sijoittaa sen kansioon juureen, missä web-palvelimen jaettavaksi tarkoitettut tiedostot ovat. Ubuntussa tämä kansio on /var/www/. Tiedostot voi avata muokattaviksi komennoilla:

```
sudo gedit /var/www/.healthcheck.html
```

ja

```
sudo gedit /var/www/index.html
```

.healthcheck-tiedoston sisällöksi riittää "OK" ja kun se on lisätty, tiedosto voidaan tallentaa. Index.html-tiedostossa on valmiina otsikko "It works!" ja ainoa muutos on otsikon perään lisättävä numero. Tällöin voidaan tarvittaessa helposti tunnistaa, miltä fyysiseltä palvelimelta sivu todellisuudessa tulee.

### *Lvsdrss komentojonotiedoston asentaminen*

Lvsdrss-mallitiedosto [6, s. 261 - 263] kopioitiin kansioon /etc/init.d. Tiedoston sisältö on liitteessä 3. Tiedostossa on VIP-alkuinen rivi, jolla määritellään virtuaalinen ip-osoite, mikä järjestelmässä on käytössä. Tämä tulee vaihtaa arvoon 172.16.75.200. Tiedosto muuten kelpaa sellaisenaan ja se voidaan seuraavaksi tallentaa.

Tämän jälkeen se asetettiin käynnistymään aina koneen käynnistyessä. Jotta symbolisten linkkien hallinta olisi helpompaa, asennettiin apuohjelma nimeltä `sysv-rc-conf` komennolla

```
sudo apt-get install sysv-rc-conf
```

Paketti vaatii asennettavaksi riippuvia paketteja ja nämä tulee hyväksyä asennettavaksi. Tämän jälkeen tarvittavat symboliset linkit voitiin tehdä komennolla

```
sudo sysv-rc-conf --level 123456 lvsdrrs on
```

Komennolla lisätään `lvsdrrs` komentojonotiedosto ajettavaksi jokaisella ajotasolla. Komento tulee antaa `/etc/init.d` kansiossa.

Tämän jälkeen palvelinkoneet olivat valmiina testeihin.

### 4.3 Klusterin testaus

Testeissä keskityttiin korkean käytettävyyden kannalta oleellisiin osiin. High Availability -ohjelmistoa testattiin tekemällä ohjaajakonepari, jossa pääohjaajan vioittuessa varalla olevan koneen tulisi ottaa palvelu haltuun. Toinen oleellinen osa on ohjaajakoneen takana olevien palvelimien automaattinen poisto vikatilanteissa ja palautus, kun ongelma on korjattu. Jos toinen edellä mainituista osista ei toimi kunnolla, tuloksena on ongelmia, jotka vaikuttavat palvelun laatuun.

Testivaihe on suhteellisen yksinkertainen, kun järjestelmä on saatu toimintakuntoon. Toiminnallisuus voidaan testata sammuttamalla jokin laite, tai palvelu ja palvelun siirtyminen tai palautuminen koneiden välillä voidaan todeta tarkastelemalla lokitiedostoja. Samalla havaitaan asioita ohjelmistojen toiminnasta.

#### *Ohjaajakoneen vioittuminen*

Ohjaajakoneeseen aiheutettiin vika sammuttamalla High Availability -ohjelmisto komennolla

```
sudo /etc/init.d/heartbeat stop
```

Lokista `/var/log/ha-log` voi seurata High Availability -ohjelmiston tapahtumia. Sen loppupään saa auki komenolla

```
sudo tail -f /var/log/ha-log
```

Ohjaajakoneen ja varaohjaajakoneen välisen yhteyden katkettua, alkaa tapahtumasarja, minkä aikana resurssien hallinta siirretään koneelta toiselle. Loki on hyvin selkeä, ja sieltä erottaa hyvin vaiheet siitä, kun huomataan toisen koneen kuolleen sinne saakka, kunnes resurssit on siirretty toimivan koneen hoidettaviksi [6, s. 141 - 144].

Ohjelmisto käynnistettiin uudelleen komennolla

```
sudo /etc/init.d/heartbeat start
```

Koneiden yhteistyö sujui tässäkin tilanteessa hyvin ja resurssien siirto takaisin onnistui ilman ongelmia.

### *Palvelimen vioittuminen ja palautuminen*

Palvelinkoneiden vioittuminen toteutettiin ottamalla verkkokortti pois käytöstä. Toinen hyvä tapa aiheuttaa keinotekoinen ongelma on sammuttaa web-palvelin, jolla tarkistettava .healthcheck-tiedosto sijaitsee. Kun sitä ei voi lukea, koneen oletetaan olevan rikki.

Kun palvelinkoneet ovat päällä, ja niiden tiedot ovat ldirectord:llä tiedossa, koneiden päivittäminen tapahtuu automaattisesti. Ohjaajakoneella tulee olla ldirectord oikein asetettuna. Tällöin ldirectord suorittaa jokaiselle palvelimelle kyselyn tietyin väliajoin ja siihen oikein vastaamalla palvelin pysyy listalla. Lista, jolla palvelimet ovat, on nimeltään IPVS-taulukko ja sen sisällön näkee komennolla

```
sudo watch ipvsadm -L -n
```

Lista päivittyy automaattisesti ja antaa hyvän yleiskuvan palvelimien tilasta. Kun palvelimelta otetaan verkkokortti pois käytöstä, kuluu aiemmin asetuksissa määritetty aika, ennen kuin muutos näkyy listalla. Kun palvelin tulee kuntoon, se lisätään ldirectord:n toimesta listalle. Kun ldirectord seuraavankerran lähettää palvelimelle testipaketin ja saa siihen oikeanlaisen vastauksen, kone päivittyy automaattisesti taulukkoon. Kuvassa 12 on taulukon rakenne ja esimerkkisisältönä yksi palvelin. TCP-alkuinen rivi kertoo virtuaalisen ip-osoitteen ja aikataulutuksen menetelmän.

```

Every 2,0s: ipvsadm -L -n
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port          Forward Weight ActiveConn InActConn
TCP  172.16.75.200:80 wrr
  -> 192.168.1.107:80             Route   1         0         0

```

*Kuva12. IPVS-taulukko.*

Palvelimien osalta lennossa tapahtuva lisääminen ja poistaminen toimivat todella hyvin. Palvelimen vioittuessa se häviää listalta ja kun se saadaan taas toimintaan, se ilmestyy itsestään listalle.

## 5 LOPPUMIETTEET

Työn aihe osoittautui haastavaksi, yllättävän laajaksi ja sen aikana eteen tuli tilanteita, joissa piti rajoittaa tutkittavaksi otettavia kohteita. Työn parissa tuli perehdyttyä korkean käytettävyyden käsitteeseen ja sitä toteuttaviin käytännön ratkaisuihin. Työn tekeminen oli myös yleishyödyllistä ja sitä tehdessä oppi asioita, mitkä helpottavat työskentelyä myös muun tyyppisten ongelmien parissa.

Korkean käytettävyyden ratkaisuja on hyvin tarjolla ja tässä työssä käytetyille ohjelmille löytyy myös vaihtoehtoja. Työssä käytetyt ohjelmat ovat korkealaatuisten käyttäjien suosimia, ja niistä löytyy runsaasti tietoa niin kirjallisena kuin sähköisessä muodossa. Ohjelmistojen säätämisen vara ja erityyppisten kokonaisuuksien tekeminen on joustavaa. Tässä työssä erikseen käytettyjä ohjelmistoja löytyy myös valmiiksi paketoituna ja osin asennettuna, jolloin tutustumisen vaiva ei ole niin suuri ja lopullista tuotetta pääsee testaamaan nopeammin.

Aihe oli hyödyllinen, sillä palvelinpuoli on Linuxin vahvuuksia. Vaikka jakeluksi valittiin harvemmin palvelinkäyttöön päätyvä Ubuntu, sille löytyi lähes kaikki valitut ohjelmistot suoraan pakettienhallinnasta, mikä auttoi keskittymistä itse korkean käytettävyyden klustereihin.

**VIITELUETTELO**

- [1] Stallings, William, *Operating Systems – Internals and Design Principles*. New Jersey: Pearson Prentice Hall. 2005.
- [2] Top 500 Supercomputing sites [Verkkodokumentti, viitattu 23.10.2008]. Saatavissa: <http://www.top500.org/stats/list/31/archtype>.
- [3] Weygant, Peter S., *Clusters for High Availability – A Primer of HP Solutions*. Second Edition. New Jersey: Prentice Hall. 2001.
- [4] Bookman, Charles, *Linux Clustering – Building and Maintaining Linux Clusters*. Indiana: New Riders Publishing. 2003.
- [5] HomePage: Linux HA [Verkkodokumentti, viitattu 6.11.2008]. Saatavissa: <http://linux-ha.org/>.
- [6] Kopper, Karl, *The Linux Enterprise Cluster*. San Francisco: No Starch Press. 2005.
- [7] The AMD-V™ Story [Verkkodokumentti, viitattu 9.11.2008]. Saatavilla: [http://www.amd.com/us-en/0,,3715\\_15781\\_15785,00.html](http://www.amd.com/us-en/0,,3715_15781_15785,00.html).
- [8] Intel® Virtualization Technology (Intel® VT) in Computing [Verkkodokumentti, viitattu 9.11.2008]. Saatavilla: <http://www.intel.com/technology/virtualization/>.
- [9] The Linux Virtual Server Project - Linux Server Cluster Project for Load Balancing [Verkkodokumentti, viitattu 9.11.2008]. Saatavilla: <http://www.linuxvirtualserver.org/index.html>.
- [10] VMWare: Virtualization via Hypervisor, Virtual Machine & Server Consolidation [Verkkodokumentti, viitattu 9.11.2008]. Saatavilla: <http://www.vmware.com/>.
- [11] EasyVMX!: Virtual Machine Creator [Verkkodokumentti, viitattu 9.11.2008]. Saatavilla: <http://www.easyvmx.com/>.
- [12] Ganglia Monitoring System [Verkkodokumentti, viitattu 4.11.2008]. Saatavissa: <http://ganglia.info/>.
- [13] NoMachine NX - Desktop Virtualization and Remote Access Management Software [Verkkodokumentti, viitattu 6.11.2008]. Saatavissa: <http://www.nomachine.com/index.php>.
- [14] Download VMware Player, Free VMware - VMware [Verkkodokumentti, viitattu 9.11.2008]. Saatavilla: <http://www.vmware.com/download/player/>.
- [15] The Linux Kernel Archives [Verkkolähde, viitattu 10.11.2008]. Saatavilla: <http://www.kernel.org/>.

## LIITE 1 - UBUNTUN ASENTAMINEN

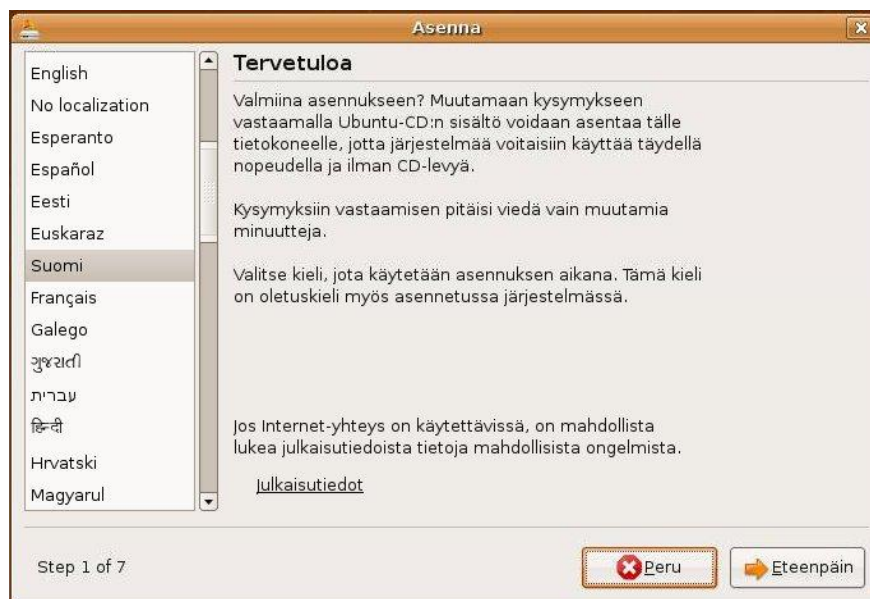
Ubuntu on suosittu sekä ilmainen Linux -jakelu, joka on saatavilla kymmenille eri kielille ja jolle on tarjolla paljon tukea suuren käyttäjäkuntansa vuoksi. Sen asentamiseen tarvitaan cd:lle poltettu levykuva, jonka voi hankkia osoitteesta [www.ubuntu.com](http://www.ubuntu.com) tai suomesta noudettaessa sen saa parhaiten Funetin ftp -palvelimelta osoitteesta <http://ftp.funet.fi/pub/mirrors/releases.ubuntu.com/8.04.1/>. Jos et tiedä minkä version tarvitset, valintasi on ubuntu-8.04.1-desktop-i386.iso -niminen tiedosto. Se on 32-bittinen työpöytäversio PC-koneille. Kun tiedosto on latautunut se pitää polttaa cd:lle levykuvana, jotta siitä tulee levy jolta koneen voi käynnistää. Ladattua tiedostoa ei tule purkaa ja sitten polttaa levyille, sillä silloin se ei toimi.

Kun cd on poltettu, levy asetetaan tietokoneen cd -asemaan ja sama asema valitaan asemaksi jolta kone etsii käynnistyslevyä. Seuraavaksi kone käynnistetään uudelleen ja asennus lähtee käyntiin. Näyttöön ilmestyy valikko, josta voidaan valita joko suora asennus tai käynnistys Live -tilaan, jolloin käyttöjärjestelmä ladataan cd -levyltä ja sen jälkeen sitä voidaan käyttää tai asentaa se graafisesti sen jälkeen.

Alkuvalikko on alla olevan kuvan näköinen. Siitä valittiin vaihtoehto ”Asenna Ubuntu” ja tämän jälkeen järjestelmä käynnistyi graafiseen tilaan.



Graafisesessa tilassa tulee valita kieli, joka tässä tapauksessa oli Suomi. Eteenpäin painikkeella siirryttiin seuraavaan vaiheeseen.

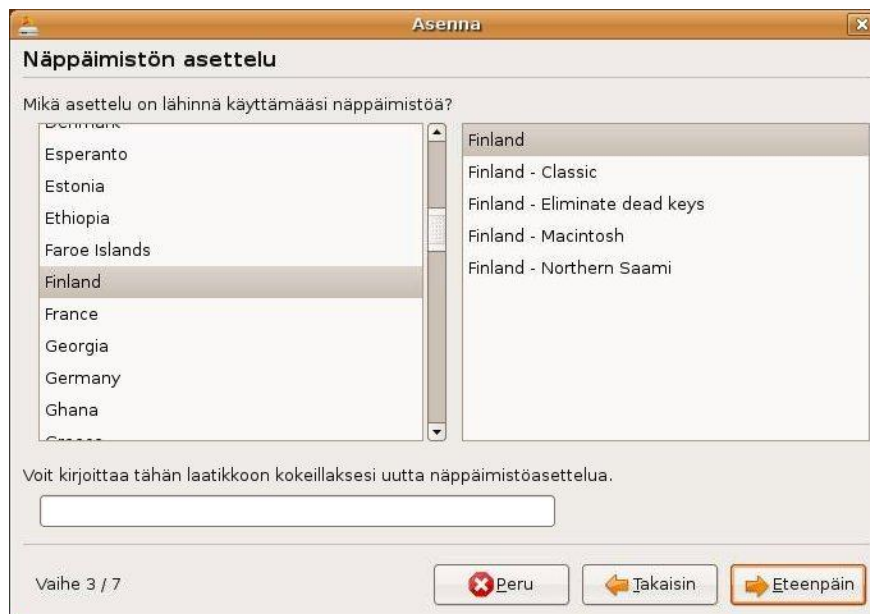


Seuraavaksi valittiin asuinpaikka. Valinta tehdään osoittamalla hiirellä sopivaa kaupunkia. Tässä tapauksessa löytyi juuri oikea paikka eli Helsinki. Jos ei juuri omaa kaupunkia löydy, valitaan lähin samalla aikavyöhykkeellä oleva paikka.

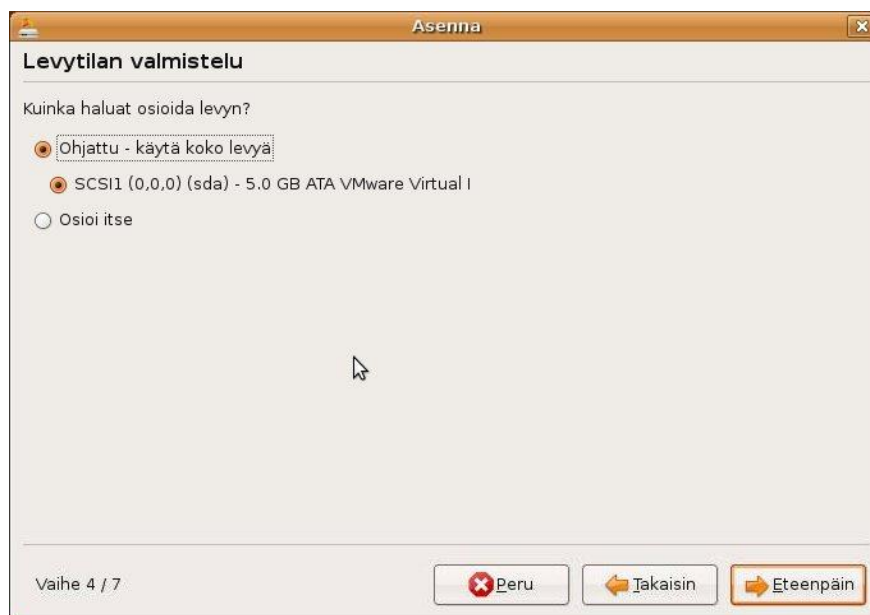




Seuraavaksi valittiin näppäimistön asettelu, oletusasetus on valittu oikein ja jos tarvitset erikoisemman näppäinasetteluun voit valita toisin. Alhaalla olevaan laatikkoon voit testata eri asetteluja.



Seuraavaksi valitaan haluttu osiojako. Oletuksena asennus tekee sivutus- ja käyttöosion, joka on alustettu ext3-muotoon. Mikäli kiintolevyllä on osioita, jotka täytyy säilyttää tai jos haluat tehdä erikoisemman osiojaon, valitse osioi itse. Tässä tapauksessa käytettiin koko käytettävissä oleva levy vakio-osioinnilla.



Seuraavassa vaiheessa kysytään käyttäjän nimi, haluttu tunnus, salasana ja koneen nimi. Salasanassa ei ole erikoisuusvaatimuksia kuten erikoismerkit tai numerot ja esimerkiksi käyttäjätunnuskin kelpaa.

**Asenna**

**Kuka olet?**

Mikä on nimesi?

Mitä nimeä haluat käyttää kirjautuaksesi koneelle?

*Jos useampi kuin yksi ihminen käyttää tätä tietokonetta, voit luoda lisää käyttäjätilejä asennuksen jälkeen.*

Valitse salasana pitääksesi käyttäjätilesi turvallisena.

*Syötä sama salasana kahdesti, jotta mahdolliset kirjoitusvirheet huomataan.*

Mikä on tämän tietokoneen nimi?

*Nimeä käytetään jos asetat tietokoneesi muiden näkyville verkossa.*

Vaihe 5 / 7

Lopuksi eteen tulee yhteenveto, josta järjestelmä lähtee asentumaan "Asenna" -nappia painamalla. Käynnistyslataajan asetuksia pääsee tarvittaessa muuttamaan "Lisäasetukset..." -nappia painamalla.

**Asenna**

**Valmiina asennukseen**

Uusi käyttöjärjestelmä asennetaan seuraavilla asetuksilla:

Kieli: Finnish  
 Näppäimistön asettelu: Finland  
 Nimi: nappipappi  
 Kirjautumisnimi: nappipappi  
 Sijainti: Europe/Helsinki  
 Siirtoapulainen:

Jos jatkat, alla luetellut muutokset kirjoitetaan levyille. Muussa tapauksessa voit tehdä itse lisää muutoksia.

**VAROITUS:** Tämä tuhoaa kaiken tiedon poistamistasi levyosioista sekä niistä levyosioista, joihin olet määrittänyt luotavaksi uuden tiedostojärjestelmän.

Vaihe 7 / 7

Käyttöjärjestelmän asennuttua, se on hyvä päivittää heti uusimpaan versioonsa parhaan tuvallisuuden, käytettävyyden ja yhteensopivuuden saavuttamiseksi.

**LIITE 2 - VMWAREN ASENNUSLISTAUS**

artsi@Ubuntu-Desktop:~/Työpöytä/vmware-player-distrib\$ **sudo ./vmware-install.pl**  
 [sudo] password for Ubuntu-Desktop:  
 Creating a new VMware Player installer database using the tar4 format.

Installing VMware Player.

**In which directory do you want to install the binary files?**

[usr/bin]

**What is the directory that contains the init directories (rc0.d/ to rc6.d)?**

[etc]

**What is the directory that contains the init scripts?**

[etc/init.d]

**In which directory do you want to install the daemon files?**

[usr/sbin]

**In which directory do you want to install the library files?**

[usr/lib/vmware]

**The path "/usr/lib/vmware" does not exist currently. This program is going to create it, including needed parent directories. Is this what you want?**

[yes]

**In which directory do you want to install the documentation files?**

[usr/share/doc/vmware]

**The path "/usr/share/doc/vmware" does not exist currently. This program is going to create it, including needed parent directories. Is this what you want?**

[yes]

The installation of VMware Player 2.0.5 build-109488 for Linux completed successfully. You can decide to remove this software from your system at any time by invoking the following command: "/usr/bin/vmware-uninstall.pl".

**Before running VMware Player for the first time, you need to configure it by invoking the following command: "/usr/bin/vmware-config.pl". Do you want this program to invoke the command for you now? [yes]**

Making sure services for VMware Player are stopped.

Stopping VMware services:

Virtual machine monitor done

Configuring fallback GTK+ 2.4 libraries.

**In which directory do you want to install the theme icons?**

[usr/share/icons]

**What directory contains your desktop menu entry files? These files have a**

**.desktop file extension.** [/usr/share/applications]

**In which directory do you want to install the application's icon?**

[/usr/share/pixmaps]

Trying to find a suitable vmmon module for your running kernel.

**None of the pre-built vmmon modules for VMware Player is suitable for your running kernel. Do you want this program to try to build the vmmon module for your system (you need to have a C compiler installed on your system)?** [yes]

Using compiler "/usr/bin/gcc". Use environment variable CC to override.

**Your kernel was built with "gcc" version "4.2.3", while you are trying to use "/usr/bin/gcc" version "4.2.4". This configuration is not recommended and VMware Player may crash if you'll continue. Please try to use exactly same compiler as one used for building your kernel. Do you want to go with compiler "/usr/bin/gcc" version "4.2.4" anyway?** [no] yes

**What is the location of the directory of C header files that match your running kernel?** [/lib/modules/2.6.24-21-generic/build/include]

Extracting the sources of the vmmon module.

Building the vmmon module.

Using 2.6.x kernel build system.

make: Siirytään hakemistoon "/tmp/vmware-config0/vmmon-only"

make -C /lib/modules/2.6.24-21-generic/build/include/.. SUBDIRS=\$PWD

SRCROOT=\$PWD/. modules

make[1]: Siirytään hakemistoon "/usr/src/linux-headers-2.6.24-21-generic"

CC [M] /tmp/vmware-config0/vmmon-only/linux/driver.o

CC [M] /tmp/vmware-config0/vmmon-only/linux/hostif.o

CC [M] /tmp/vmware-config0/vmmon-only/common/comport.o

CC [M] /tmp/vmware-config0/vmmon-only/common/cpuid.o

CC [M] /tmp/vmware-config0/vmmon-only/common/hash.o

CC [M] /tmp/vmware-config0/vmmon-only/common/memtrack.o

CC [M] /tmp/vmware-config0/vmmon-only/common/phystrack.o

CC [M] /tmp/vmware-config0/vmmon-only/common/task.o

CC [M] /tmp/vmware-config0/vmmon-only/common/vmciContext.o

CC [M] /tmp/vmware-config0/vmmon-only/common/vmciDatagram.o

CC [M] /tmp/vmware-config0/vmmon-only/common/vmciDriver.o

CC [M] /tmp/vmware-config0/vmmon-only/common/vmciDs.o

CC [M] /tmp/vmware-config0/vmmon-only/common/vmciGroup.o

CC [M] /tmp/vmware-config0/vmmon-only/common/vmciHashtable.o

CC [M] /tmp/vmware-config0/vmmon-only/common/vmciProcess.o

CC [M] /tmp/vmware-config0/vmmon-only/common/vmciResource.o

CC [M] /tmp/vmware-config0/vmmon-only/common/vmciSharedMem.o

CC [M] /tmp/vmware-config0/vmmon-only/common/vmx86.o

CC [M] /tmp/vmware-config0/vmmon-only/vmcore/moduleloop.o

LD [M] /tmp/vmware-config0/vmmon-only/vmmon.o

Building modules, stage 2.

MODPOST 1 modules

CC /tmp/vmware-config0/vmmon-only/vmmon.mod.o

LD [M] /tmp/vmware-config0/vmmon-only/vmmon.ko

```
make[1]: Poistutaan hakemistosta "/usr/src/linux-headers-2.6.24-21-generic"
cp -f vmmon.ko ../vmmon.o
make: Poistutaan hakemistosta "/tmp/vmware-config0/vmmon-only"
The module loads perfectly in the running kernel.
```

/dev is dynamic:

Trying to find a suitable vmblock module for your running kernel.

**None of the pre-built vmblock modules for VMware Player is suitable for your running kernel. Do you want this program to try to build the vmblock module for your system (you need to have a C compiler installed on your system)?**  
[yes]

Extracting the sources of the vmblock module.

Building the vmblock module.

Using 2.6.x kernel build system.

```
make: Siirytään hakemistoon "/tmp/vmware-config0/vmblock-only"
make -C /lib/modules/2.6.24-21-generic/build/include/.. SUBDIRS=$PWD
SRCROOT=$PWD/. modules
make[1]: Siirytään hakemistoon "/usr/src/linux-headers-2.6.24-21-generic"
CC [M] /tmp/vmware-config0/vmblock-only/linux/block.o
CC [M] /tmp/vmware-config0/vmblock-only/linux/control.o
CC [M] /tmp/vmware-config0/vmblock-only/linux/dblInkIst.o
CC [M] /tmp/vmware-config0/vmblock-only/linux/dentry.o
CC [M] /tmp/vmware-config0/vmblock-only/linux/file.o
CC [M] /tmp/vmware-config0/vmblock-only/linux/filesystem.o
CC [M] /tmp/vmware-config0/vmblock-only/linux/inode.o
CC [M] /tmp/vmware-config0/vmblock-only/linux/module.o
CC [M] /tmp/vmware-config0/vmblock-only/linux/stubs.o
CC [M] /tmp/vmware-config0/vmblock-only/linux/super.o
LD [M] /tmp/vmware-config0/vmblock-only/vmblock.o
Building modules, stage 2.
MODPOST 1 modules
CC /tmp/vmware-config0/vmblock-only/vmblock.mod.o
LD [M] /tmp/vmware-config0/vmblock-only/vmblock.ko
make[1]: Poistutaan hakemistosta "/usr/src/linux-headers-2.6.24-21-generic"
cp -f vmblock.ko ../vmblock.o
make: Poistutaan hakemistosta "/tmp/vmware-config0/vmblock-only"
The module loads perfectly in the running kernel.
```

/dev is dynamic:

**Do you want networking for your virtual machines?** (yes/no/help) [yes]

Configuring a bridged network for vmnet0.

**Your computer has multiple ethernet network interfaces available: eth0, eth1. Which one do you want to bridge to vmnet0?** [eth1]

The following bridged networks have been defined:

. vmnet0 is bridged to eth1

**Do you wish to configure another bridged network?** (yes/no) [no]

**Do you want to be able to use NAT networking in your virtual machines?** (yes/no)  
[yes]

Configuring a NAT network for vmnet8.

**Do you want this program to probe for an unused private subnet?** (yes/no/help)  
[yes]

Probing for an unused private subnet (this can take some time)...

The subnet [172.16.20.0/255.255.255.0](#) appears to be unused.

The following NAT networks have been defined:

. vmnet8 is a NAT network on private subnet [172.16.20.0](#).

**Do you wish to configure another NAT network?** (yes/no) [no]

**Do you want to be able to use host-only networking in your virtual machines?**  
[yes]

Configuring a host-only network for vmnet1.

**Do you want this program to probe for an unused private subnet?** (yes/no/help)  
[yes]

Probing for an unused private subnet (this can take some time)...

The subnet [172.16.21.0/255.255.255.0](#) appears to be unused.

The following host-only networks have been defined:

. vmnet1 is a host-only network on private subnet [172.16.21.0](#).

**Do you wish to configure another host-only network?** (yes/no) [no]

Trying to find a suitable vmnet module for your running kernel.

**None of the pre-built vmnet modules for VMware Player is suitable for your running kernel. Do you want this program to try to build the vmnet module for your system (you need to have a C compiler installed on your system)?** [yes]

Extracting the sources of the vmnet module.

Building the vmnet module.

Using 2.6.x kernel build system.

```
make: Siirytään hakemistoon "/tmp/vmware-config0/vmnet-only"
```

```
make -C /lib/modules/2.6.24-21-generic/build/include/.. SUBDIRS=$PWD
```

```
SRCROOT=$PWD/. modules
```

```
make[1]: Siirytään hakemistoon "/usr/src/linux-headers-2.6.24-21-generic"
```

```
CC [M] /tmp/vmware-config0/vmnet-only/driver.o
```

```
CC [M] /tmp/vmware-config0/vmnet-only/hub.o
```

```
CC [M] /tmp/vmware-config0/vmnet-only/userif.o
```

```

CC [M] /tmp/vmware-config0/vmnet-only/netif.o
CC [M] /tmp/vmware-config0/vmnet-only/bridge.o
CC [M] /tmp/vmware-config0/vmnet-only/filter.o
CC [M] /tmp/vmware-config0/vmnet-only/procfs.o
CC [M] /tmp/vmware-config0/vmnet-only/smac_compat.o
SHIPPED /tmp/vmware-config0/vmnet-only/smac_linux.x386.o
LD [M] /tmp/vmware-config0/vmnet-only/vmnet.o
Building modules, stage 2.
MODPOST 1 modules
WARNING: could not find
/tmp/vmware-config0/vmnet-only/.smac_linux.x386.o.cmd for
/tmp/vmware-config0/vmnet-only/smac_linux.x386.o
CC /tmp/vmware-config0/vmnet-only/vmnet.mod.o
LD [M] /tmp/vmware-config0/vmnet-only/vmnet.ko
make[1]: Poistutaan hakemistosta "/usr/src/linux-headers-2.6.24-21-generic"
cp -f vmnet.ko ../vmnet.o
make: Poistutaan hakemistosta "/tmp/vmware-config0/vmnet-only"
The module loads perfectly in the running kernel.

```

Starting VMware services:

```

Virtual machine monitor                done
Blocking file system:                  done
Virtual ethernet                        done
Bridged networking on /dev/vmnet0       done
Host network detection                  done
Host-only networking on /dev/vmnet1 (background) done
DHCP server on /dev/vmnet1              done
Host-only networking on /dev/vmnet8 (background) done
DHCP server on /dev/vmnet8              done
NAT service on /dev/vmnet8              done

```

The configuration of VMware Player 2.0.5 build-109488 for Linux for this running kernel completed successfully.

You can now run VMware Player by invoking the following command:  
"/usr/bin/vmplayer".

Enjoy,

--the VMware team

## LIITE 3 - LVSDRRS TIEDOSTON SISÄLTÖ

```

#!/bin/bash
#
# lvsdrdrs init script to hide loopback interfaces on LVS-DR
# Real servers. Modify this script to suit
# your needs. You at least need to set the correct VIP address(es).
#
# Script to start LVS DR real server.
#
# chkconfig: 2345 20 80
# description: LVS DR real server
#
# You must set the VIP address to use here:
VIP=172.16.75.200
host=`/bin/hostname`
case "$1" in
start)
    # Start LVS-DR real server on this machine.
    /sbin/ifconfig lo down
    /sbin/ifconfig lo up
    echo 1 > /proc/sys/net/ipv4/ip_forward
    echo 1 > /proc/sys/net/ipv4/conf/all/hidden
    /sbin/ifconfig lo:0 $VIP netmask 255.255.255.255 up
    /sbin/route add -host $VIP dev lo:0
    echo 1 > /proc/sys/net/ipv4/conf/lo/hidden
;;
stop)
    # Stop LVS-DR real server loopback device(s).
    /sbin/ifconfig lo:0 down
;;
status)
    # Status of LVS-DR real server.
    islothere=`/sbin/ifconfig lo:0 | grep $VIP`
    isrothere=`netstat -rn | grep "lo:0" | grep $VIP`
    if [ ! "$islothere" -o ! "isrothere" ];then
        # Either the route or the lo:0 device
        # not found.
        echo "LVS-DR real server Stopped."
    else
        echo "LVS-DR Running."
    fi
;;
*)
    # Invalid entry.
    echo "$0: Usage: $0 {start|status|stop}"
    exit 1
;;
esac

```