



VIRTUALISOINTIYMPÄRISTÖ OPETUSKÄYTTÖÖN

Jari Lohtari

Opinnäytetyö
Marraskuu 2011
Tietotekniikka
Tietoliikennetekniikka ja tietoverkot
Tampereen ammattikorkeakoulu

TAMPEREEN AMMATTIKORKEAKOULU
Tampere University of Applied Sciences

TIIVISTELMÄ

Tampereen Ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Tietoliikennetekniikka ja tietoverkot

LOHTARI, JARI: Virtualisointiympäristö opetuskäyttöön

Opinnäytetyö 27 sivua, liite 7 sivua

Marraskuu 2011

Tässä työssä käsitellään virtualisointiympäristön rakentamista Tampereen ammattikorkeakoulun tarpeisiin. Työn tarkoituksena oli luoda ja toteuttaa virtualisointijärjestelmä, jonka avulla erilaisten ohjelmistoympäristöjä ja ohjelmia olisi mahdollista kokeilla paljon nopeammin kuin perinteisesti.

Työn valmistumisen jälkeen suoritetuissa kokeiluissa todettiin, että käytetyn laitteiston resurssit riittävät hyvin raskaaseen laskentaan, järjestelmä tarjoaa myös mahdollisuuden skaalata resurssien käyttöä virtuaalikoneiden tarpeita vastaaviksi. Järjestelmä mahdollistaa esimerkiksi virtuaalikoneiden luomisen opiskelijoiden ohjelmistoprojekteja varten, sillä palvelimen virhetilanne ei uhkaa isäntäkoneen järjestelmää.

Työtä tehdessä kerättiin paljon tietoa erilaisten Linux-käyttöjärjestelmälle tehtyjen virtualisointiohjelmistojen toiminnoista ja yhteensopivuuksista. Järjestelmä luotiin käyttäen avoimen lähdekoodin sovelluksia ja hallintaa varten luotiin skripti, jonka avulla yksinkertaistettiin perustoimintoja kuten laitteiston uudelleenkäynnistystä.

ABSTRACT

Tampere University of Applied Sciences

ICT Engineering

Telecommunications Engineering and Networks

LOHTARI, JARI: Virtualization environment for Teaching Uses

Bachelor's thesis 27 pages, appendix 7 pages

November 2011

This thesis work handles both designing and implementing a virtualization environment. The end result of will be used as a system in which students and teachers can try out and test new software far faster than methods commonly used.

This system can be used for a variety of uses, such as lending computing power to software projects made by students of Tampere University of Applied Sciences. Since everything is done virtually, there is no risk regarding the physical hardware or base operating system. System can also support different user needs, as it is possible to scale the amount of processing power for a virtual instance.

During this work a lot of knowledge was gathered on virtualization on Linux-systems, compatibility of different hypervisors and orchestrating software and networking. System was created using Open Sourced software and some coding was done using PERL.

SISÄLLYS

1 JOHDANTO	6
2 VIRTUALISOINTI.....	7
3 LAITTEET JA OHJELMISTOT	9
3.1 Tietokoneet.....	9
3.2 Verkkolaitteet.....	10
3.3 Verkkotekniikat.....	11
3.4 Käyttöjärjestelmät	12
3.5 Virtualisointiohjelmisto	14
4 PALVELUN RAKENTAMINEN	17
4.1 Tietokoneet.....	17
4.2 Virtualisointiohjelmisto	18
4.3 Tallennusratkaisu	19
4.4 Verkko.....	20
4.4.1 Perusrakenne	20
4.4.2 PRIVNET-verkko	21
4.4.2 PUBNET-verkko.....	22
5 KÄYTTÖÖNOTTO JA JATKOKEHITYS.....	23
5.1 Käyttöönotto.....	23
5.2 Jatkokehitys.....	24
6 POHDINTA	25
LÄHTEET.....	26
LIITE.....	28

LYHENNELUETTELO

DHCP – Dynamic Host Configuration Protocol, verkkoasetusten automaattiseen jakoon käytetty protokolla.

HDD – Hard Disk Drive, perinteinen magneettilevyihin perustuva kiintolevy.

Hypervisor – Fyysisen ja virtuaalisen laitteiston rajapinta

IANA – Internet Assigned Numbers Authority, Internetin IP-osoitteita hallinnoiva organisaatio.

IP – Internet Protocol, tietoverkkojen loogisista osoitteista vastaava protokolla

LTS – Long Term Support, normaalia pidemmällä valmistajan tuella julkaistu ohjelmisto.

NAS – Network-attached Storage, verkkoon liitetty levypalvelin

OSI – Open Systems Interconnection, verkkostandardi.

RAID – Redundant Array of Independent Disks, levyjärjestelmien muodostamiseen käytetty tekniikka.

SSD – Solid State Drive, puolijohteisiin perustuva kiintolevy.

SSH – Secure Shell, tietokoneen etähallintaan käytetty protokolla.

VLAN – Virtual Local Area Network, mahdollistaa loogisen verkon luomisen suurempaan verkkoon.

VNC – Virtual Network Computing, graafiseen etähallintaan käytetty sovellus.

1 JOHDANTO

Rajattuihin käyttöoikeuksiin perustuvassa hyvän tietoturvatason tietojärjestelmässä ongelmaksi muodostuu kokeilumahdollisuuksien rajallisuus: ”kuinka kokeilla uutta ohjelmaa tilanteessa, jossa ohjelman asentaminen vaatii laajempia käyttäjätunnuksia kuin käytössä olevat?”. Edellä kuvattuun ongelmaan pyritään etsimään ratkaisua tässä dokumentissa käsitellyllä projektilla, jonka tavoitteena oli luoda helppokäyttöinen ja skaalattava virtualisointiympäristö.

Palvelun tarkoituksena on luoda mahdollisuus luoda virtuaalisia tietokoneita ennaltamäärittelyin asetuksin ja sisällöin mahdollisimman käyttäjäystävällisellä tavalla. Työssä ei käsitellä järjestelmän ohjaamiseen käytettyjen sovelluksien luontia vaan keskitytään Linux-käyttöjärjestelmän käyttöön ja palvelimien välisen tietoverkon rakentamiseen. Työssä ei myöskään käsitellä erityisesti palvelinkäyttöön tarkoitettuja ohjelmistoja tai laitteita.

Työn toisessa luvussa käsitellään käytettyjä laitteita ja ohjelmistoja sekä syitä, miksi juuri kyseistä laitetta tai ohjelmaa päädyttiin käyttämään.

Työn kolmas luku käsittelee palvelun tietokoneiden, tallennusratkaisun ja tietoverkon rakentamista. Luvussa käsitellään myös ohjelmistojen asennuksia ja konfigurointeja.

Työn neljäs luku esittelee pääpiirteisesti palvelun käyttäjärajapintaa sekä jatkokehitysmahdollisuuksia. Luvussa esitellään myös vaihtoehtoisia tapoja hallita järjestelmää.

2 VIRTUALISOINTI

Virtualisoinnilla tarkoitetaan jonkin virtuaalisen resurssin luomista fyysisen resurssin päälle. Resurssi voi olla joko yksittäinen sovellus, käyttöjärjestelmä tai kokonainen järjestelmäalusta. (Wikimedia Foundation, Virtualization, 2011)

Alustavirtualisoinnin historia ulottuu 1960-luvulle, jolloin IBM julkaisi oman CP/CMS-käyttöjärjestelmänsä, jonka tarkoituksena oli jakaa tehokkaiden tieteellisessä käytössä olleiden tietokoneiden laskentakapasiteettia käyttäjille. Virtualisointia onkin käytetty lähinnä tehokkaiden keskustietokoneiden yhteydessä, joissa oli riittävästi laskentatehoa jaettavaksi. Tavallisissa tietokoneissa ei yksinkertaisesti ollut riittävästi laskentatehoa emuloimaan kaikkia tarvittavia komponentteja. (Wikimedia Foundation, Timeline of virtualization development, 2011)

Vaikka erilaisia pienkäyttöön tarkoitettuja virtualisointiohjelmistoja olikin julkaistu pitkin 90- ja 2000-lukuja, vuosien 2006 ja 2007 aikana julkaistut Microsoft Virtual PC, Virtual-Box ja Parallors Desktop For Mac toivat alustavirtualisoinnin suuren yleisön tietoon. Lisäksi prosessorivalmistajat AMD ja Intel julkaisivat AMD-V - ja VT-x -teknologiat, joiden avulla virtualisointiohjelmisto saattoi käyttää suoraan prosessorin tiettyjä käskyjä, näin tehostaen virtuaalitietokoneiden käyttöä. (Intel Corporation, VT-x, 2011)

Pilvipalveluiden kehittyminen 2000-luvun toisella puoliskolla vahvasti kehittyneet pilvipalvelut ("Cloud Computing") ovat myös tuoneet virtualisoinnin suuren yleisön keskuuteen. Parhaimpana esimerkkinä tästä voidaan pitää Amazonin vuonna 2006 esittelemää Amazon Elastic Computing Cloud ("EC2"), jonka käyttäjä voi tilata tarpeidensa mukaisen virtuaalikoneen ja maksaa sen käytöstä tunneittain tai kiinteällä kuukausimaksulla.

EC2-palvelun käyttäminen on hyödyllistä esimerkiksi pienelle yritykselle, joka tarvitsee laskenta- ja tiedonsiirtokapasiteettia esimerkiksi ohjelmistotuotannon tai verkkosivuston tarpeisiin. Yrityksen ei tarvitse sijoittaa palvelimia omiin tiloihinsa, jolloin säästöä syntyy niin tilakustannuksissa kuin laitteisto- ja tietoliikennekustannuksissa. Koska uuden virtuaalitietokoneen luominen tapahtuu minuuteissa, kapasiteetin laajentaminen ja supistaminen kysynnän mukaan on mahdollista toteuttaa seuraamaan todellista kysyntää, poistaen "tyhjiä" resursseja. Palvelu tarjoaa myöskin paremman luotettavuuden, sillä virtuaalinen tieto-

kone on mahdollista siirtää isäntäpalvelimelta toiselle, jopa ilman keskusmuistissa olevien tietojen menetystä. (Amazon, EC2 Faqs, 2011)

Joskus tarpeeseen tulee myös yksityinen pilvipalvelu, jonka resurssit ovat vain tietyn käyttäjäyhteisön saatavilla ja hallittavissa. Tällaisten palveluiden luomista varten on ilmestynyt monia pääasiassa linuxia käyttäviä järjestelmiä, joista esimerkkinä UCP (“Ubuntu Cloud Platform”). UCP on Canonical Ltd:n julkaisema Ubuntu Linuxiin perustuva paketti, jonka avulla on mahdollista käyttää monia eri virtualisointiohjelmistoja ja hallita niitä Eucalyptus-sovelluksen avulla. Yksityinen virtualisointipalvelu vaatii luonnollisesti omat laskentayksikkönsä ja tallennusratkaisunsa, mutta mikäli käyttäjämäärät jäävät pieniksi, laitteistona voidaan käyttää tavallista PC-laitteistoa.

3 LAITTEET JA OHJELMISTOT

Projektin käytettävissä ollut budjetti ei sallinut varsinaisen palvelinkäyttöön tarkoitettun laitteiston käyttöönottoa, joten valitsimme koti- ja pientoimistokäyttöön tarkoitettua laitteistoa. (Kuvio 1) Ohjelmistoja valitessa kandidaateiksi otettiin vain avoimen lähdekoodin sovelluksia.



Kuvio 1: Projektin varattu laitteisto

3.1 Tietokoneet

Projektin laitteistovaatimuksia tutkittaessa tultiin siihen tulokseen, että tietokoneita tullaan tarvittamaan kahteen käyttötarkoitukseen: käyttäjärajapinta ja laskenta. Koska käyttäjärajapinnalla ei ole suuria resurssivaatimuksia, voitiin käyttöön ottaa varastosta löytynyt käytöstä poistettu tietokone.

Laskentakäyttöön tarkoitetuille tietokoneille asetettiin vaatimuksiksi vähintään neliytiminen suoritin, 16 gigatavua muistia sekä 2 verkkokorttia.

Tietokoneet vastaanotettiin taulukon 1 mukaisilla komponenteilla. Asennettuna olleet teratavun kiintolevyt poistettiin käytössä havaittujen ongelmien vuoksi.

Neliytimisen suorittimen vaatiminen johtuu suoraan virtualisoinnin luonteesta, jossa jokaiselle virtuaalikoneelle dedikoidaan oma suoritinydin. Käytetty Intel Core i7 –suoritin tukee myös HyperThreading-tekniikkaa, jonka avulla jokainen suoritinydin kykenee suoritta-

maan kahta prosessia samanaikaisesti. Suoritin tukee myös virtualisointia tehostavaa Intel VT-x –tekniikkaa, jonka ansiosta virtualisointiohjelmiston ei tarvitse emuloida prosessoria.

Keskusmuistin kokovaatimus liittyy suoraan prosessorin kykyyn isännöidä kahdeksaa virtuaalikonetta, sillä jokaisen virtuaalikoneen oletusarvo keskusmuistin kooksi on yksi gigatavu. Tällöin laskentayksikön keskusmuistia jää myös itse järjestelmän käytettäväksi, joten koneeseen valittua SSD-kiintolevyä ei tarvitse käyttää sivutusmuistin tarpeisiin.

Vaatus kahdesta verkkokortista selittyy halulla yksinkertaistaa ja selventää laskentayksikön konfigurointia dedikoimalla kortit omiin verkkoihinsa. Tällöin myöskin levyjärjestelmän käyttö nopeutuu kaistan ollessa vapaa virtuaalikoneiden liikenteestä.

Proessori	Intel Core I7 2600 @ 3.4 GHz
Muisti	16GB DDR3
Kiintolevyt	Intel 510 128GB SSD, Seagate 1TB HDD
Verkkosovittimet	Intel ja TP-Link 1Gbps

Taulukko 1: Peruskomponentit

3.2 Verkkolaitteet

Kuten jo kohdassa 3.1 todettiin, laskentayksiköiden käyttöön vaadittiin kaksi erillistä verkkosovittinta. Koneiden mukana toimitettiin sekä emolevyyn integroitu Intelin valmistama ja Intelin piiriä käyttävä verkkosovitin että TP-linkin valmistama Realtekin R8169-piiriä käyttävä verkkosovitin, molempien maksiminopeudeksi ilmoitettiin 1Gb/s.

TP-Linkin valmistaman verkkosovittimen toiminta osoittautui epävarmaksi käytettäessä 1Gb/s siirtonopeutta, jolloin sovitin ei saanut aikaan vakaata yhteyttä. Ongelma jäljitettiin kernelimoduulin 'r8169' aiheuttamaksi ja voitiin korjata käyttämällä uudempaa modulia.

Verkkoa ohjaa Hewlett-Packardin valmistama ProCurve E2510-24G –hallittava kytkin, joka tarjoaa 24 1Gb/s siirtonopeuteen kykenevää verkkoliitintä ja kattavan hallinnan. Verkkoon liitettiin myös Buffalo Technologyn valmistama NAS-laite, joka sisälsi yhteensä 4 teratavua tallennustilaa ja 1Gb/s verkkosovittimen.

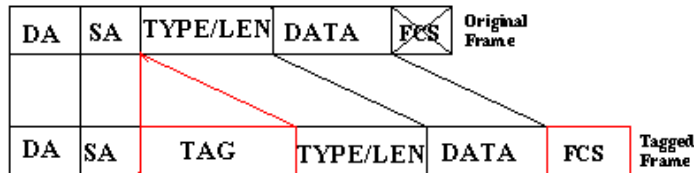
3.3 Verkkotekniikat

Verkko-osoitteistoksi valittiin IPv4-protokolla, joka harmaista osoitealueista otettiin käyttöön 10.0.0.0/8-osoiteavaruus, joka tarjoaa mahdollisuuden noin 16.7 miljoonaan yksityiseen osoitteeseen, uudemman IPv6-protokollan mahdollistamaa isompaa osoiteavaruutta ei katsottu tarpeelliseksi koneiden pienen määrän vuoksi. Verkossa käytetään sekä kiinteitä IP-osoitteita että DHCP:n kautta jaettavia dynaamisia osoitteita, jotka voivat olla joko julkisia tai harmaan alueen osoitteita. (IANA, Address space registry, 2011)

Projektin kannalta verkon yhdeksi tärkeimmistä osista muodostui VLAN-tekniikka (IEEE 802.1Q), jonka avulla samassa fyysisessä verkossa olevia verkkoja voitiin eriyttää toisistaan. Käytännössä VLAN mahdollistaa kahden fyysisesti erillään sijaitsevan verkon yhdistämisen ohjelmallisesti Ethernet-kehukseen lisättävän VLAN-tunnuksen avulla. (Cisco Systems, 802.1Q Frame, 2011)

VLAN-tunnuksen lisäys Ethernet-kehukseen voi tapahtua käytännössä missä tahansa verkkolaitteessa, projektin tapauksessa tunnuksellista kehystä tarvittiin vain terminaalikoneen kommunikointiin sekä virtuaalikoneiden että palvelimien kanssa, sillä muuten kytkin määriteltiin lähettämään tunnuksettomia kehyksiä. VLAN-tunnus lisätään Ethernet-kehyksessä lähettävän laiteosoitteen ja kehyksen pituustiedon väliin ja on kooltaan 4 tavua, sisältäen 16-bittisen indikaation tunnuksen tyypistä, 3 bittiä prioriteetti-informaatiota, bitin verran tietoa laiteosoitteiden tyypistä sekä 12 bittisen VLAN-tunnusnumeron. (Kuvio 2) Tämä neljän tavun lisäys kasvattaa Ethernet-kehysten maksimikoon yli tavallisen 1518 tavun, joten käytettävien laitteiden on joko tuettava VLAN-ominaisuutta tai niille lähetettävästä datasta tulee poistaa VLAN-tunnus. Tunnuksen lisääminen ja poistaminen aiheuttaa myös tarkistussumman FCS uudelleenlaskemisen. (Cisco Systems, 802.1Q Frame, 2011)

Sekaannuksien välttämiseksi on todettava, että tunnuksen eteen lisättävät osoitteet eivät ole IPv4-osoitteita vaan OSI-mallin tason 2 kuusitavuisia laiteosoitteita, jotka ovat uniikkeja kullekin verkkolaitteelle ja joiden käyttöalue rajoittuu aina yhteisen lähiverkon alueelle.



Kuvio 2: VLAN-tunnuksettomien ja tunnuksellisten Ethernet-kehysten ero (Cisco Systems, 802.1Q Frame, 2011)

3.4 Käyttöjärjestelmät

Projektin vaatimuksia tutkittaessa päätettiin käyttöjärjestelmäksi valita GNU/Linux-jakelu ja jakeluksi joko Centos 5.6, Debian 6 tai Ubuntu 10.04 LTS. Valittuja jakeluita käytetään yleisesti erilaisissa palvelimissa ja dokumentaatio on laajaa, helpottaen käyttöönottoa ja käyttöä. Eri jakeluita verratessa huomioon otettiin erityisesti tuki käytettyjen virtualisointiohjelmistojen kanssa, joista enemmän kohdassa 3.5.

3.4.1 CentOS 5.6

CentOS 5.6 (Community ENTERprise Operating System) on Centos Projectin julkaisema GNU/Linux-jakelu, joka perustuu Red Hat Incorporatedin RHEL-jakelun kanssa. Centos Project luo Red Hatin julkaisemasta lähdekoodipaketista omat binääripakettinsa ja julkaisee ne, tavoitteenaan 100% yhteensopivuus RHELin kanssa. (CentOS Project, Purpose, 2011)

CentOS-jakelun positiivisiin puoliin on luettava loistava yhteisön tuki ja dokumentaatio, joka on keskittynyt erityisesti palvelinkäyttöön. Jakelun tuki erilaisille virtualisointiratkaisuille on myös hyvä, sillä se tarjoaa järjestelmän RPM-pakettienhallinnan kautta suoraan sekä XEN- että KVM-järjestelmien tarvitsemat paketit.

Kokeiluissa CentOS-jakelun heikkoudeksi osoittautui heikko tuki TP-Linkin verkkosovittimen käyttämälle Realtek R8169-piirille, jonka vuoksi 1Gb/s siirtonopeus ei ollut mahdollinen. Ongelma johti CentOS-jakelun pudottamiseen käytettävien jakeluiden listalta.

3.4.2 Debian

Debian on Debian Projectin GNU/Linux-jakelu, jonka ensimmäinen julkaisu tapahtui elokuussa 1993 ja täten vertailuun otetuista jakeluista vanhin. Toisin kuin Ubuntu ja Centos, Debian ei perustu mihinkään valmiiseen GNU/Linux-jakeluun, vaan sitä on pidetty omana jakelunaan alusta alkaen ja Debian on toiminut perustana monelle muulle jakelulle, kuten tässäkin vertailussa mukana olleelle Ubuntu Linuxille. (Debian, FAQ, 2011)

Debian Projectin tavoitteena on ollut luoda täysin avoimeen koodin pohjautuva käyttöjärjestelmä, jonka mukana ei tulisi yhtäkään suljetun koodin ohjelmaa. Debian tukee myös montaa eri arkkitehtuurimallia ja sitä voidaankin käyttää aina tietokoneista pieniin sulautettuihin järjestelmiin. Jakelun ohjelmistojen jakeluun käyttämät repositoryt sisältävät yli 20 000 ohjelmistopakettia ja myös useat ohjelmistovalmistajat ovat päättäneet tukea Debiania. (Debian, FAQ, 2011)

Debianin positiiviseksi puoleksi voidaan pitää yksinkertaisuudessa pitäytyminen: jakelu saa suuria päivityksiä harvoin joten päivitysten takia tulevia käyttökatkoja tulee harvoin. Mukana ei myöskään ole apuohjelmia verkko- ja järjestelmähallinnointiin vaan järjestelmä konfiguroidaan asetustiedostoja käyttäen samalla yksinkertaistaen ylläpitoa.

Debian Linuxin versio 6 ei toiminut käytetyn ohjaussovelluksen kanssa, koska kyseinen sovellus ei tunnistanut käytettyä järjestelmää ja käytettyä virtualisointisovellusta. Version 5 kanssa sovelluksen asentaminen onnistui mutta järjestelmä ei ollut yhteensopiva käytetyn TP-linkin verkkosovittimen kanssa, johtaen samanlaisiin ongelmiin kuin CentOS-jakelua käytettäessä.

3.4.3 Ubuntu Linux

Ubuntu linux on Canonical Limitedin vuonna 2004 julkaisema GNU/Linux-jakelu joka käyttää perustanaan Debian Linuxia. Ubuntu ja sen johdannaiset kuten Kubuntu ovat GNU/Linux-julkaisuista käytetyimpiä ja tästä johtuen dokumentointi on käytetyistä jakeluista laajinta. (Ubuntu Documentation, Common, 2011)

Ubuntu Linuxista julkaistaan uusi versio kuuden kuukauden välein ja jokaisen ei-LTS-version tuki jatkuu yleisesti noin vuoden. Tämä ei kuitenkaan johda pakolliseen päivityk-

seen, sillä Ubuntusta julkaistaan kahden vuoden välein LTS-versio, jonka tuen pituus on aina vähintään kaksinkertainen normaaliin verrattuna. (Ubuntu, Releases, 2011)

Ubuntu Linuxin ensimmäinen versio 4.10 pohjautui pitkälti Debian Linuxiin, näkyvimpänä erona graafinen ilme. Ubuntu julkaisuaikataulua noudattaen uusi versio on julkaistu aina kuuden kuukauden välein, jokaisen lisätessä järjestelmään jotain uutta, mukaan lukien päivitys- ja ajurityökalut.

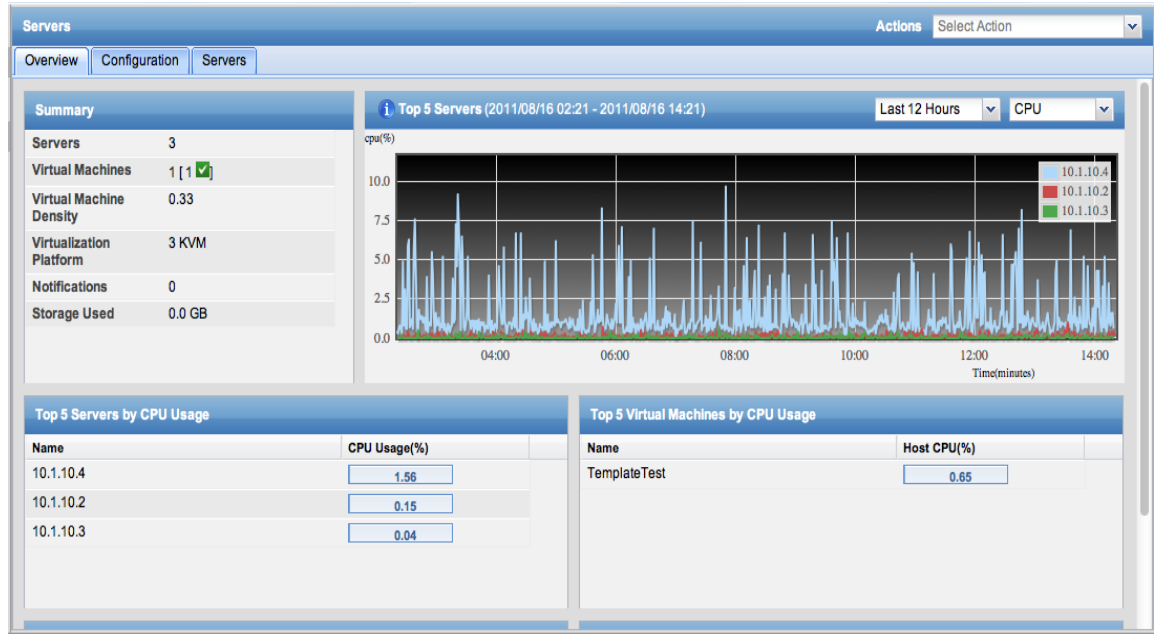
Ubuntu server-versio on versiosta 9.10 alkaen sisältänyt Eucalyptus-pilvipalveluohjelmiston, helpottaen paljon yksityisen pilven rakentamista. Server-version mukana tulee myös mukana Canonicalin Landscape-palvelu jonka avulla on mahdollista hallita palvelinta mistä tahansa internet-portaalin yli.

Ubuntu käyttöön päädyttiin lopulta kahdesta syystä: käyttöönotossa ei esiintynyt suuria kompastuskiviä ja virtualisoinnin ohjaamiseen käytettyä sovellusta suositeltiin käytettäväksi 10.04 LTS –version kanssa.

3.5 Virtualisointiohjelmisto

Projektin virtualisointiohjelmistovaihtoehdoksi tutkittiin Xen- ja KVM-Qemu –ohjelmistoja, hallinnan tapahtuessa erillisellä sovelluksella. Xenin käytöstä luovuttiin CentOS 5.6n kanssa esiintyneiden ongelmien vuoksi. KVM-Qemu –yhdistelmä tukee VT-x –teknologiaa ja sen kanssa on mahdollista käyttää monia eri käyttöjärjestelmiä. Lisäksi se mahdollistaa tilannekaappausten (”snapshot”) ottamisen, jolloin koneen tietty tila voidaan tallentaa ja uudelleenkäyttää myöhemmin. (KVM, Faq, 2011)

KVM-Qemu –yhdistelmä on niin sanottu tason 2 hypervisor, eli se vaatii alustakseen käyttöjärjestelmän, tässä tapauksessa linuxin. Yhdistelmän luomat virtuaalikoneet näkyvät isäntäkoneella tavallisina prosesseina, joiden resurssien käyttöä ohjaa käyttöjärjestelmän ydin eli kernel. Tämä rakenne ei ole tehokkain mahdollinen, sillä resursseja syövät sekä alustana käytetty käyttöjärjestelmä että virtuaalikone, mutta laitteistokapasiteetin riittävyyden vuoksi tätä ei koettu ongelmaksi. (KVM, Faq, 2011)



Kuvio 3: Palvelimen hallintanäkymä ConVirt-ohjelmistossa

Virtualisointia ohjaamaan valittiin Convirturen luoma ConVirt Open Source –ohjelmisto, joka on kehitetty käyttäen Python- ja Ajax-ohjelmointikieliä. Palvelimille ConVirt-ohjelmistosta asennetaan joko palvelinohjelma tai agenttiohjelma joista palvelin tarjoilee käyttäjälle käyttöliittymän sekä ohjaa agenttisovelluksia, jotka ohjaavat virtualisointia itse palvelinkoneella. Palvelin- ja agenttisovellukset käyttävät keskinäiseen tiedonsiirtoon SSH-protokollaa ja tarvittaessa siirtävät tiedostoja SCPn ylitse. Palvelinsovellukseen kuuluu myös Pythonilla toteutettu web-palvelin, jonka kautta koko järjestelmää hallitaan. (Convirture wiki, Faq, 2011)

ConVirtä käytettäessä jää käyttäjän vastuulle kuormittaa palvelimia tasaisesti, sillä ohjelmistoon ei ole toteutettu kuormaa tasaavaa logiikkaa. Tämä ei kuitenkaan aiheuta suurempia ongelmia ja jopa auttaa palvelun skaalaamisessa, sillä tämän vuoksi palvelimien raudan ei tarvitse olla identtistä. Käyttäjää avustaa myös ohjelmiston palvelinkohtainen tilanäkymä, jonka avulla on mahdollista tutkia palvelimen eri resurssien käyttöä eri aikaväleillä. (Kuva 3)

ConVirt mahdollistaa myös käyttäjäryhmien luomisen, perusasetuksilla käyttäjä voi olla joko ylläpitäjä tai tavallinen käyttäjä, joista ylläpitäjä voi muun muassa luoda uusia käyttäjiä. Myös levyhallinta tapahtuu ConVirtin kautta, palvelinsovellukselle syötetään NAS-palvelimen liitospiste paikallisissa järjestelmissä jonka jälkeen kaikki järjestelmän tietokoneet käyttävät yhteistä paikkaa työtiedostojen tallentamiseen.

Virtuaalitietokoneen hallintaa helpottamaan KVM-Qemu –yhdistelmä tarjoaa Console-näkymän, jonka käyttö on mahdollista VNC:n avulla. Edelleen ConVirtia käyttämällä tämä näkymä voidaan tunneloida niin, että käyttö onnistuu vaikka käyttäjä olisikin yhteydessä hallintapaneeliin palvelinverkon ulkopuolelta. ConVirt mahdollistaa myös migrate-toiminnon, jonka avulla virtuaalikone on mahdollista siirtää palvelimelta toiselle ilman että virtuaalikonetta on tarvetta sammuttaa.

4 PALVELUN RAKENTAMINEN

Palvelun rakentaminen suoritettiin heinäkuun 2011 aikana ja käyttöönotto tapahtui elokuussa 2011. Seuraavat kappaleet kertovat palvelun osien rakentamisesta sekä käyttöönotossa esiintyneistä ongelmista.

4.1 Tietokoneet

Virtualisointipalvelimet valmisteltiin asentamalla niihin Ubuntu Linux 10.04 LTS – käyttöjärjestelmä. Järjestelmiin lisättiin asennusvaiheessa peruskomponenttien lisäksi vain SSH-palvelin etäkäyttöä varten ja päivitettiin moduuli ‘e100’ Intelin integroidun verkkokortin käyttöönottamiseksi. Palvelimille annettiin nimet “Picard”, “Info” ja “Data”.

Terminaalikone valmisteltiin asentamalla Debian Linuxin perusjärjestelmä ilman lisäkomponentteja ja sille annettiin nimi “Scrapheap”. Terminaalikoneen graafiseksi käyttöliittymäksi asennettiin Fluxbox, joka käynnistetään xinit-ohjelman avulla niin, että samalla poistuu mahdollisuus käyttää komentoriviä muuten kuin Fluxboxin kautta. Lisäksi koneeseen asennettiin PERL-tulkki järjestelmän ohjaamiseen käytetyn komentojonon käyttämiseksi.

Scrapheapin Fluxbox-käyttöliittymää muokattiin niin, ettei käyttäjällä ole mahdollisuuksia käynnistää kuin internet-selain, VNC-sovellus, tiedostoselain, järjestelmän ohjaukseen käytetty komentojonotiedosto sekä komentorivi, näistä vain tiedostoselain, VNC-sovellus ja internet-selain on mahdollista käynnistää ilman ylläpitäjän salasanaa. Internet-selaimen käytön yhteydessä luomat käyttäjätiedostot myös poistetaan jokaisen käynnistyksen yhteydessä ja palautetaan esiasetettuun tilaan asettaen esimerkiksi aloitussivun Convirturen kirjautumissivuksi.

Jokaiseen järjestelmään liitettiin NFS-protokollaa käyttäen kaksi verkkohakemistoa NAS-levypalvelimelta, joihin säilytettiin kaikki virtualisointiin liittyvät työtiedostot kuten kiintolevyt, CD-levykuvat ja konfiguraatitiedostot. Paremman tietoturvan vuoksi palvelinkoneiden SSH-palvelin ei vastaa kuin sisäverkon puolelta, Convirturen vaatimuksiin kuuluneen root-oikeuden vuoksi.

Palvelinkoneiden suunnittelun yhteydessä päätettiin, että muuttuvia tiedostoja sisältävät hakemistot, kuten esimerkiksi /var, sijoitettaisiin tavalliselle kiintolevylle SSD-levyn si-

jaan. Kolmesta käyttöönvaratusta kiintolevystä kaksi kuitenkin rikkoutui kaksi viikkoa laitteiston saapumisen jälkeen, joten ne poistettiin kokonaan käytöstä. On kuitenkin oletettavaa, että tämä ei juurikaan vaikuta SSD-levyjen käyttöikään, lopulliseksi järjestelmän tilantarpeeksi todettiin alle 5 gigatavua joten todennäköisyys saman sektorin käytölle on suhteellisen pieni.

Järjestelmän keveys yhdistettynä suureen laskentatehoon ja SSD-kiintolevyyn tekee perusjärjestelmän käytöstä hyvin nopeaa, uudelleenkäynnistykseen kuluvaksi ajaksi todettiin alle 20 sekuntia.

4.2 Virtualisointiohjelmisto

Virtualisointiin käytetty KVM-Qemu -ohjelmistopaketti asennettiin palvelinkoneisiin Ubuntun Apt-pakettienhallintajärjestelmää käyttäen, jolloin järjestelmään asentui myös kernelin lisämoduulit prosessorin virtualisointitekniikoiden hyödyntämiseksi.

KVM-Qemu käyttää verkkoyhteyden muodostamiseen tap-tunnelointia, jonka luomiseen palvelinkoneilta tuli löytyä silta verkkokortille. Täten palvelinkoneiden eth1-verkkosovitin sillattiin sovittimeksi br0, jonka kautta virtuaalikoneiden verkkoyhteys mahdollistui. (KVM, Faq, 2011)

Virtualisoinnin ohjaamiseen käytetyn ConVirtin palvelinsovellus asennettiin Picard-palvelimelle, virtualisointiagentit asennettiin sekä Picardille, Datalle että Infolle. ConVirtin palvelin- ja agenttisovelluksen asennusohjelmat asensivat omatoimisesti kaikki tarvittavat järjestelmäkomponentit, joihin kuului mm. python-tulkki ja MySQL-palvelin. Palvelinsovelluksen asetustiedostoa muokattiin niin, että kaikki työtiedostot poislukien templatet tallennettiin NAS-levypalvelimelle, vähentäen SSD-levyn kuormitusta.

Virtualisointilaitteita ohjaamaan luotiin PERL-scriptikieltä käyttäen liitteen 1 mukainen sovellus, jonka avulla palvelun laitteiden käynnistämiseen ja sammuttamiseen liittyvät operaatiot ketjutettiin käytön helpottamiseksi. Sovellus käyttää SSH-sovellusta ohjataksaan laitteita ja tämän vuoksi palvelinkoneisiin sekä NAS-levypalvelimeen vaadittiin RSA-avaimet salasanatonta todentamista varten.

4.3 Tallennusratkaisu

Projektia varten toimitettiin Buffalo Technologyn valmistama NAS-levypalvelin, jonka sisältämä 4 teratavua kiintolevytilaa muodostui kahdesta RAID 0 –tilaan asetetusta kiintolevystä. RAID 0 –tilan etu on parantunut suorituskyky yksittäisiin kiintolevyihin verrattuna, sillä levyille kirjoitettu data jakaantuu molemmille levyille. Tilan ongelma on heikentynyt kykyä selvittää levyjen rikkoutumisesta, sillä tilanteessa jossa toinen levyistä hajoaa, ei tietojen pelastamiselle ole enää mitään mahdollisuutta. (Wikimedia Foundation, RAID, 2011)

Levypalvelinta testatessa todettiin ongelmaksi palvelimen tuki pelkästään AFP- ja SMB-protokollille, joiden käyttö osoittautui suorituskyvyltään NFS-protokollaa heikommaksi. Valmistajan dokumentaation perusteella protokollan käyttöönotto ei ollut mahdollista ilman levypalvelimen käyttämän ohjelmiston avaamista ja muokkaamista. Buffalo Technologyn laitteiden muokkaamisohjeisiin erikoistuvan sivuston tietojen ja nmap-sovelluksen käytön perusteella tultiin tulokseen, että kyseessä on Unix-sukuista käyttöjärjestelmää käyttävä laite.

Lisätutkimuksissa todettiin, että levypalvelimen käyttöjärjestelmän muutoksiin tarvittava SSH-palvelin oli valmiiksi asennettuna, mutta käytöstä poistettuna. Vakioasetukset eivät myöskään sallineet käyttäjätunnus-salasana –yhdistelmää käyttävää kirjautumista, vaan vaati RSA-avaimen käyttöä. Levypalvelimen järjestelmässä oli kuitenkin mahdollista suorittaa komentoja `acp-commander` –ohjelmaa käyttäen, jonka avulla laitteelle saatettiin lähettää Buffalon oman protokollan mukaisia käskyjä. `Acp-commander` –ohjelman avulla muokattiin SSH-palvelimen asetuksia niin, että kirjautuminen root-tunnuksella oli mahdollista. Samalla root-tunnukselle luotiin RSA-avain, jonka avulla voitiin kirjautua suoraan levypalvelimen käyttöjärjestelmään. (Nas Central, Buffalo LS-WVL, 2011)

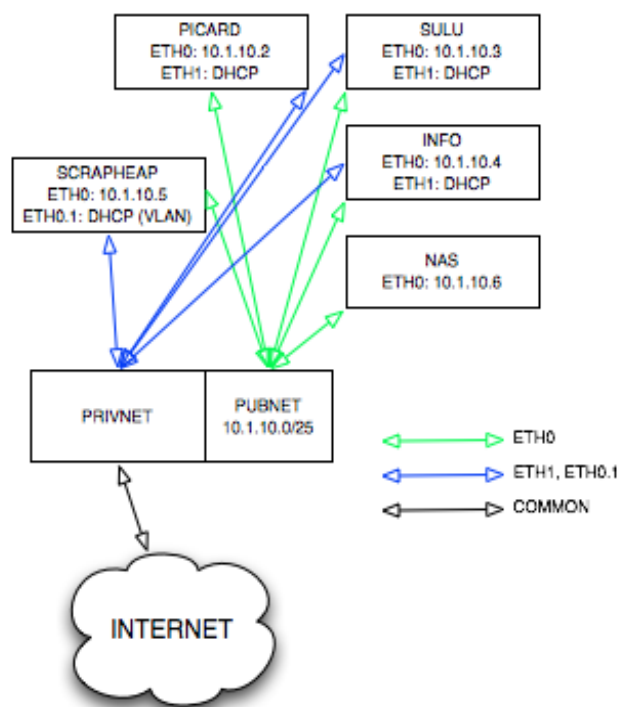
Levypalvelimeen asennettiin avaamisen jälkeen NFS-protokollan vaatimat kernel-moduulit ja sovellukset, jonka lisäksi mahdollistettiin käyttäjätunnuksen ja salasanan käyttö sisäänkirjautumisessa. Järjestelmään lisättiin kaksi NFS-levyjakoa, PrimStore ja Secstore, jotka liitettiin palvelin- ja terminaalikoneisiin sekä lisättiin ConVirtin käyttämään tietokantaan. NFS-levyjaot näkyvät järjestelmässä tavallisina hakemistoina, esimerkiksi SecStore-verkkohakemisto asetettiin hakemistoon `/mnt/secstore` ja ilman lisäselvittämistä, kuten aktiivisten levyliitosten tutkimista, ei hakemistoa tunnista verkkolevyä käyttäväksi.

Palvelua rakennettaessa todettiin että tiedostojen siirtäminen virtuaalikoneilta ja -koneille on tarpeettoman hankalaa, sillä esimerkiksi USB-kiintolevyllä olevat tiedot olisi ensin tallennettava joko Internetiin tai NAS-palvelimelle ja sieltä virtuaalikoneelle. Ratkaisuna ongelmaan terminaalikone asetettiin liittämään kaikki liitetyt massamuistit automaattisesti tiettyyn hakemistoon, joka jaettiin edelleen verkkoon käyttämällä Samba-palvelinta. Täten käyttäjän on mahdollista käyttää virtuaalikoneelta terminaalikoneen verkkojakoja, johon on liitetty omaan alikansioonsa käytettävä USB-massamuisti.

4.4 Verkko

Verkkoa suunniteltaessa todettiin että virtuaalikoneiden käyttämä julkinen verkko ja palvelun muodostavien laitteiden välinen verkko on erotettava toisistaan jotta virtuaalikoneiden toiminta ei vaikuttaisi itse palvelun toimintaan. Verkko noudattaa perinteisen tähtiverkon rakennusmallia, jossa jokainen laite on omalla johdollaan kiinni verkon keskuskytkimessä, joka tarvittaessa liitetään esimerkiksi käyttöpaikan yleiseen lähiverkkoon.

4.4.1 Perusrakenne



Kuvio 4: Verkon rakenne

Verkkojen erottaminen toisistaan toteutettiin käyttämällä keskuskytkimenä toimineen Hewlett Packard Procurve E2510-24G –kytkimen tukemaa VLAN-toimintoa.

Kytkimen porteista 1-17 ja 23-24 muodostettiin PRIVNET-verkko ja porteista 18-23 PUBNET-verkko, joiden VLAN-ID-arvoiksi asetettiin 1 ja 2.

Koska terminaalikoneessa ei ollut mahdollista käyttää kahta verkkosovittinta, asetettiin portti 23 niin, että kytkin ohjaa liikenteen PRIVNET-verkkoon mikäli VLAN-tunnus on lisätty ethernet-kehukseen ja PUBNET-verkkoon mikäli kehys on normaali. Portti 24 on tarkoitettu kytkimen liittämiseksi esimerkiksi organisaation verkkoon tai suoraan internetiin ja yhdistetty PRIVNET-verkkoon. Porttien asetuksiin on mahdollista tehdä muutoksia kytkimen hallintapaneelin kautta ja PUBNET-verkon porttien määrä haluttiin pitää mahdollisimman pienenä liikennehäiriöiden vähentämiseksi. (Hewlett Packard, Datasheet, 2011)

Verkkolaitteiden välillä käytettiin tavallista CAT6-kaapelia RJ-45 –liittimin, joka mahdollisti 1Gbps siirtonopeuden laitteiden välille. Mikäli kytkimen mukana olisi toimitettu valokuitumodulit, liittäminen runkoverkkoon olisi ollut mahdollista käyttäen sekä valokuitua että tavallista ethernet-kaapelia.

4.4.2 PRIVNET-verkko

Palvelinten TP-Link –verkkosovittimet (eth1) sekä terminaalikone kytkettiin PRIVNET-verkkoon mahdollistamaan virtuaalikoneiden verkkoyhteydet. Koska PRIVNET-verkon tarkoituksena tulee olemaan myös Internet-yhteys, jätettiin DHCP-palvelin pois verkon palveluista sillä Internetiin tai runkoverkkoon liitettäessä kyseisessä verkossa oleva DHCP-palvelin tulisi aiheuttamaan ongelmia mikäli kyseisessä verkossa jo tarjottaisiin DHCP-palvelua.

Terminaalikoneen liittämiseksi PRIVNET-verkkoon tuli käyttöjärjestelmään ladata 8021q-moduuli ja luoda virtuaalinen verkkosovitin lähettämään VLAN-tageilla varustettua liikennettä. Luominen tapahtuu Debian-pohjaisissa järjestelmissä lataamalla vaadittu moduuli komennolla ”modprobe 8021q” ja lisäämällä tiedot ”/etc/network/interfaces”-tiedostoon, verkkosovittimen nimeksi asetetaan käytettävän fyysisen sovittimen nimi ja yhdistetään siihen pisteellä haluttu VLAN-tunniste. Terminaalikoneen tapauksessa nimeksi muodostui ”eth0.1”, joka käyttää liikennöintiin eth0-sovittinta VLAN-tunnisteella 1. (Debian Wiki, NetworkConfiguration, 2011)

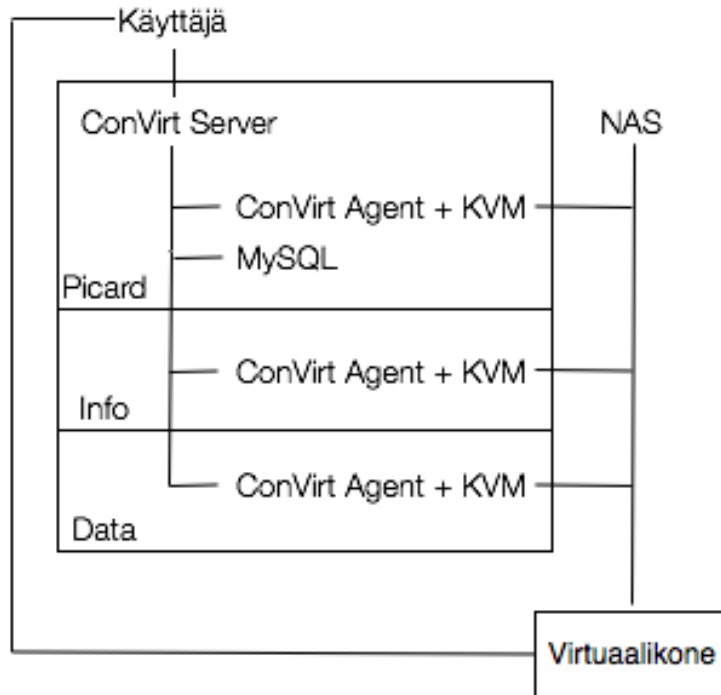
4.4.2 PUBNET-verkko

PUBNET-verkko luotiin toimimaan palvelun osien välisenä eristettynä verkkona, johon kytkettiin palvelinten Intel-verkkosovittimet (eth0), NAS sekä terminaalikone. Verkon osoitteeksi asetettiin 10.1.10.0/25 (desimaalimuodossa 255.255.255.128) joka tarjoaa 126 laiteosoitetta ja käyttää broadcast-osoitetta 10.1.10.127. Terminaalikone kommunikoi verkon kanssa käyttäen tavallisia VLAN-tunnuksettomia ethernet-kehyksiä.

5 KÄYTTÖÖNOTTO JA JATKOKEHITYS

Palvelun valmistuessa lopullinen ohjelmistokokonaisuus muodostui oheisen kuvan mukaisesti.

(Kuvio 5) Kuvan suhteet osoittavat, missä järjestyksessä eri ohjelmistot ja laitteet toimivat sekä yhdessä että käyttäjän kanssa.



Kuvio 5: Palvelimien ja ohjelmistojen yhteydet

5.1 Käyttöönotto

Palvelun käyttöönottoa varten luotiin kirjallinen järjestelmäkuvaus sisältäen tekniset yksityiskohdat ja käyttöohjeet perustoimintojen käyttämisestä varten. Peruskäyttöä varten luotiin myös Ubuntu Linux 11.04 –käyttöjärjestelmään perustuva referenssinä toimiva virtuaalikone, jolla palvelun toimintaa voidaan esitellä uusille käyttäjille. Palvelun tilaajalle luotiin ylläpitäjätasoinen tunnus hallinnoimiseksi ja yksi normaali tunnus normaalin käyttäjän oikeuksien esittelemiseksi.

Palvelun todettiin skaalautuvan melko hyvin, täysin uuden palvelinkoneen lisäämiseksi järjestelmään tarvitsee tietokoneeseen asentaa käyttöjärjestelmän lisäksi agentti- ja virtualisointisovellukset. Uusien palvelimien asennusta voi helpottaa käyttämällä valmiiden palvelimien kiintolevyjä lähteinä, joista kloonataan tyhjälle koneelle kokonainen järjestelmä palveluun liitettäväksi.

5.2 Jatkokehitys

Palvelinkoneiden jatkokehitystä tulenee ohjaamaan uudet LTS-julkaisut Ubuntu Linuxista, seuraavan julkaisun on suunniteltu tapahtuvan huhtikuussa 2012. Tähän version päivittäessä on otettava huomioon mahdolliset yhteensopivuusongelmat ConVirtin kanssa, joista esimerkki saatiin jo Debian 6 ja ConVirt – yhdistelmän kanssa. Yksittäisten komponenttien päivittämisen kanssa ongelmia ei lähitulevaisuudessa ole odotettavissa, sillä käytetty järjestelmä tarjoaa laajan laitteistotuen ja käytetyt komponentit eivät ole erikoiskäyttöön tarkoitettuja.

Verkkolaitteiston jatkokehittämiskohteita ei ole muita kuin kytkin, jonka 24 portin kapasiteetista on rakennushetkellä käytössä 8 porttia. Mikäli järjestelmään lisätään uusia palvelinkoneita tai tallennuslaitteita, enimmäisvaatimus on 2 porttia laitetta kohden, jolloin järjestelmään on mahdollista lisätä vielä enintään 8 palvelinta. Maksimimäärää on mahdollista kasvattaa tiedonsiirtokyvyn kustannuksella terminaalikoneen kohdalla käytettyä VLAN-tekniikkaa käyttäen.

Tallennusratkaisun jatkokehityksessä suurin kohde on tallennusvarmuuden parannus, sillä levypalvelimen nykyinen RAID 0 –levypakka ei ole varma tapa tallentaa tietoa. Mikäli levypalvelinta ei haluta vaihtaa enemmän kiintolevyjä mahdollistavaksi laitteeksi, on mahdollista ottaa käyttöön toinen samankokoinen levypalvelin ja peilata palvelimien verkkohakemistot keskenään, jolloin todennäköisyys kiintolevyrikoista johtuville tiedostojen kaotamisille pienenee.

Ohjelmistojen osalta jatkokehityskohteita tulee olemaan virtualisoinnin ohjaamiseen käytetyt sovellukset. Canonical on ilmoittanut että Ubuntu Serverin cloud-osa tulee siirtymään Eucalyptuksesta Openstackin käyttöön ja onkin tutkittava onko uudesta ohjausjärjestelmästä merkittävää hyötyä. Terminaalikoneen salasanaa vaativat osaset olisi myös mahdollista siirtää käyttämään ConVirtin käyttäjätietokantaa, jolloin järjestelmän käyttöön riittäisi yksi salasana erillisten ConVirt- ja järjestelmäsalausanojen sijaan.

6 POHDINTA

Projektin aloitusvaiheessa työtehtävänä oli tutkia mahdollisuuksia luoda virtualisointijärjestelmä, jonka avulla olisi mahdollista säästää aikaa sekä resursseja uusien ohjelmistoympäristöjen testauksessa. Koko työn perustana käytettiin erään opettajakunnan jäsenen luomaa spesifikaatiota, jota alettiin toteuttaa käytännössä.

Huolimatta projektissa sattuneista takaiskuista kuten laitteistorikoista, voidaan projektin todeta onnistuneen melko hyvin. Täydelliseksi projektin olisi tehnyt laskentaresurssien automaattinen jakaminen manuaalisen sijaan. Toteutuksen aikana kertyi paljon tietoa erilaisista linux-jakeluista ja ohjausskriptaa varten opiskeltua PERL-kieltä on mahdollista hyödyntää vielä paljon suuremmissakin projekteissa.

Nyt luotu järjestelmä on oiva lähtökohta seuraavan version suunnittelulle, sillä sitä voidaan käyttää referenssinä uuden järjestelmän suunnittelussa. On toivottavaa että kohdan 5.2 jatkokehitysideoita arvioidaan seuraavassa suunnitteluvaiheessa ja suunnittelussa käytetään myös apuna käyttäjiltä tulleita palautteita toimivuudesta ja käytettävyydestä. On toivottavaa, että järjestelmää käytetään mahdollisimman laajasti ja pyritään tutkimaan sen mahdollisuudet eri käyttöjärjestelmien ja mahdollisesti jopa prosessoriarkkitehtuurien käyttämiseksi käyttäen samaa henkeä kuin tätä versiota rakennettaessa.

LÄHTEET

Wikimedia Foundation, Virtualization, 2011 [Viitattu lokakuussa 2011]
<http://en.wikipedia.org/wiki/Virtualization>

Wikimedia Foundation, Timeline of virtualization development, 2011 [Viitattu lokakuussa 2011]
http://en.wikipedia.org/wiki/Timeline_of_virtualization_development

Intel Corporation, VT-x, 2011 [Viitattu lokakuussa 2011]
<http://www.intel.com/technology/itj/2006/v10i3/1-hardware/6-vt-x-vt-i-solutions.htm>

Amazon EC2 Faqs [Viitattu lokakuussa 2011]
<http://aws.amazon.com/ec2/faqs/>

IANA, Address space registry, 2011 [Viitattu lokakuussa 2011]
<http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.txt>

Cisco Systems, 802.1Q Frame, kuvio, 2011 [Viitattu lokakuussa 2011]
http://www.cisco.com/en/US/tech/tk389/tk689/technologies_tech_note09186a0080094665.shtml

Cisco Systems, 802.1Q frame, 2011 [Viitattu lokakuussa 2011]
http://www.cisco.com/en/US/tech/tk389/tk689/technologies_tech_note09186a0080094665.shtml

CentOS Project, Purpose, 2011 [Viitattu lokakuussa 2011]
<http://www.centos.org/modules/tinycontent/index.php?id=3>

Debian, FAQ, 2011 [Viitattu lokakuussa 2011]
<http://www.debian.org/doc/FAQ/>

Ubuntu Documentation, Common, 2011 [Viitattu lokakuussa 2011]
<https://help.ubuntu.com/community/CommonQuestions>

Ubuntu, Releases, 2011 [Viitattu lokakuussa 2011]
<https://wiki.ubuntu.com/Releases>

KVM, Faq, 2011 [Viitattu lokakuussa 2011]
<http://www.linux-kvm.org/page/FAQ>

Convirture, Wiki, 2011 [Viitattu lokakuussa 2011]
http://www.convirture.com/wiki/index.php?title=Convirt2_faq

Wikimedia Foundation, RAID, 2011 [Viitattu lokakuussa 2011]
<http://en.wikipedia.org/wiki/RAID>

Nas Central, LS-WVL, 2011 [Viitattu lokakuussa 2011]
<http://buffalo.nas-central.org/wiki/Category:LS-WVL>

Hewlett Packard, E2510-24G Datasheet, 2011 [Viitattu lokakuussa 2011]
<http://h20195.www2.hp.com/v2/GetPDF.aspx/4AA0-6431ENW.pdf>

Debian Wiki, NetworkConfiguration, 2011 [Viitattu lokakuussa 2011]
<http://wiki.debian.org/NetworkConfiguration>

LIITE

Komentojonotiedosto järjestelmän perushallintaan

```
#!/usr/bin/perl
```

```
## Declarations
```

```
use strict;
```

```
## Declarations end, subproto begin
```

```
sub menu;
```

```
sub cloudup;
```

```
sub clouddown;
```

```
sub environment;
```

```
sub servicecheck;
```

```
sub nfsmounter;
```

```
## Subproto end
```

```
## Main script
```

```
menu();
```

```
## Main script ends
```

```
sub menu()
```

```
{
```

```
    system('clear');
```

```
    my ($rows,$cols) = split(/ /,`/bin/stty size`);
```

```
    for(my $index=0; $index < $cols; $index++)
```

```
    {
```

```
        print "*";
```

```
    }
```

```
    print "\n*\n*";
```

```
    print " Choose:\n* 1. Bring cloud up.\n*\n* 2. Cloud reboot\n* 3. Cloud
```

```
down\n*\n*";
```

```
    for(my $index=0; $index < $cols; $index++)
```

```
    {
```

```
        print "*";
```

```
    }
```

```
    print "\n\nChoose: ";
```

```
    chomp(my $input = <STDIN>);
```

```
    if($input == "1")
```

```
    {
```

```

                                cloudup();
                                }
if($input == "3")
    {
                                clouddown();
                                }
if($input eq "2")
    {
                                cloudboot();
                                }
}
sub environment()
{
    print "Checking host 10.1.10.2:";
    my $receivedping1 = `ping 10.1.10.2 -c 2|grep 'received'`;
    chomp($receivedping1);
    my @temp = split(' ', $receivedping1);
    $receivedping1 = $temp[3];
    if($receivedping1 == 2)
    {
        print " ok.\n";
    }
    else
    {
        print "HOST 10.1.10.2 NOT RUNNING, RECHECK!";
        environment();
    }
    print "Checking host 10.1.10.3:";
    my $receivedping2 = `ping 10.1.10.3 -c 2|grep 'received'`;
    chomp($receivedping2);
    @temp = split(' ', $receivedping2);
    $receivedping2 = $temp[3];
    if($receivedping2 == 2)
    {
        print " ok.\n";
    }
}

```

```
}
else
{
    print "HOST 10.1.10.3 NOT RUNNING, RECHECK!\n";
    environment();
}
print "Checking host 10.1.10.4:";
my $receivedping3 = `ping 10.1.10.4 -c 2 |grep "received"`;
chomp($receivedping3);
@temp = split(' ', $receivedping3);
$receivedping3 = $temp[3];
if($receivedping3 == 2)
{
    print " ok.\n";
}
else
{
    print "HOST 10.1.10.4 NOT RUNNING, RECHECK!\n";
    environment();
}
print "Checking nas:";
my $receivedpingnas = `ping 10.1.10.6 -c 2 |grep "received"`;
chomp($receivedpingnas);
@temp = split(' ', $receivedpingnas);
if($temp[3] == 2)
{
    print " ok.\n";
}
else
{
    print "NAS NOT RUNNING, RECHECK!\n";
    environment();
}
print "Everything ok, proceeding.\n";
servicecheck();
```

```

}
sub cloudup()
{
    system('clear');
    print "Please ensure that cloud nodes, NAS-box and network switch are running.";
    print "\n";
    print 'Cloud coming up.';
    print "\n";
    sleep 2;
    environment();
}
sub servicecheck
{
    system('clear');
    my $nasstatus=`ssh 10.1.10.6 -l root -i /root/.ssh/id_rsa
/etc/rc.d/extensions.d/S98_nfs.sh status`;
    my @temp=split('\n',$nasstatus);
    if(($#temp+1) == 6)
    {
        print "NAS services ok, remounting drives.";
        nfsmounter();
        print "...ok!\n";
    }
    else
    {
        print "NAS services down, restarting NFS.\nALL WARNINGS
SHOULD BE IGNORED!\n";
        system("ssh 10.1.10.6 -i /root/.ssh/id_rsa -l root
/etc/rc.d/extensions.d/S98_nfs.sh stop");
        system("ssh 10.1.10.6 -i /root/.ssh/id_rsa -l root
/etc/rc.d/extensions.d/S98_nfs.sh start");
        print 'Will now sleep for 120 seconds.';
        for(my $index=0; $index < 120; $index++)
        {

```

```

        print '!';
        system("sleep 1");
    }
    print "Alles gut, rechecking.";
    servicecheck();
}
my $webservstatus=`wget -qO- http://10.1.10.2:8081`;
if(length($webservstatus) != 0)
{
    system("clear");
    print "Convirt up, cloud is ok.\n\nLogin using
http://10.1.10.2:8081\n";
}
else
{
    system("ssh 10.1.10.2 -i /root/.ssh/id_rsa -l root 'cd /srv/convirt
&& ./convirt-ctl start'");
    system("clear");
    print "Convirt up, cloud is ok.\n\nLogin using
http://10.1.10.2:8081\n";
}
}
sub nfsmounter
{
    print ".";
    system("ssh 10.1.10.2 -i /root/.ssh/id_rsa -l root /usr/bin/nfsmount");
    print ".";
    system("ssh 10.1.10.3 -i /root/.ssh/id_rsa -l root /usr/bin/nfsmount");
    print ".";
    system("ssh 10.1.10.4 -i /root/.ssh/id_rsa -l root /usr/bin/nfsmount");
    print ".";
}
sub clouddown
{
    print "WARNING! WARNING! WARNING!\n\n";
}

```



```

        print "This will send shutdown-commands to cloud nodes, killing every VM
in process.\n";
        print "If you really want to do this, type 'KillMeNow', typing something else
will send you back to menu.\n\n";
        print "Type here: ";
        my $tempvalue=<STDIN>;
        chomp($tempvalue);
        if($tempvalue eq 'KillMeNow')
        {
            print "Shutting down cloud nodes.";
            print ".";
            system("ssh 10.1.10.4 -i /root/.ssh/id_rsa -l root shutdown -h now");
            print ".";
            system("ssh 10.1.10.3 -i /root/.ssh/id_rsa -l root shutdown -h now");
            print ".";
            system("ssh 10.1.10.2 -i /root/.ssh/id_rsa -l root shutdown -h now");
            print ".";

            print "ok!\n";
            print "NAS-box needs to be turned off by hand using the PWR-
switch located in the backplate.\n";
            print "Thank you for using this cloud. Have a nice day!\n";
        }
        else
        {
            menu();
        }
    }
sub cloudboot
{
    print "REBOOT!\n";
    system("ssh 10.1.10.4 -i /root/.ssh/id_rsa -l root reboot");
    system("ssh 10.1.10.3 -i /root/.ssh/id_rsa -l root reboot");
    system("ssh 10.1.10.2 -i /root/.ssh/id_rsa -l root reboot");
    for(my $index=0; $index < 30; $index++)
    {

```

```
        print ".";  
        system("sleep 1");  
    }  
}
```