Valtteri Maijala

Sisäisesti virtualisoidun IaaS-palvelun toteuttaminen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikan koulutusohjelma

Insinöörityö

27.2.2015

Metropolia

Tekijä	Valtteri Maijala
Otsikko	Sisäisesti virtualisoidun IaaS-palvelun toteuttaminen
Sivumäärä	77 sivua
Aika	27.2.2015
Tutkinto	insinööri (AMK)
Koulutusohjelma	tietotekniikka
Suuntautumisvaihtoehto	tietotekniikka
Ohjaaja	lehtori Harri Ahola

Etäresurssipalveluiden käyttö on nykyään jokapäiväistä ja niiden tarjooma voidaan käsittää laitteiston, alustan ja/tai ohjelmiston tasolla. Insinöörityön tavoitteena oli tarkastella etäresurssipalvelumalleja, niiden laskutusrakenteita ja toteuttaa laitteistotason etäresurssipalveluratkaisu virtualisoituna käyttäen sisäisiä virtualialisointikerroksia.

Pilottivaiheen jälkeinen varsinainen toteutus tehtiin yhdellä fyysisellä Dell R710-palvelimella. Virtualisointialustana palvelimessa käytettiin VMware vSphere 5.5 ympäristöä. Työssä luotiin olemassaolevan vSphere-ympäristön sisälle etäresurssipalvelun käyttöön uusi oma sisäkkäinen virtualisoitu vSphere-virtualisointiympäristö.

Tähän ympäristöön tuotiin ja otettiin käyttöön etäresurssipalvelun mahdollistavat virtuaalikoneet. Järjestelmään luotiin kolme erilaista ympäristöä, joissa käytettiin erilaisia verkkokonfiguraatiorakenteita. Ensimmäisessä ympäristössä käytettiin suoraa sillattua yhteyttä ulkoiseen verkkoon, toisessa käytettiin porttikohtaista osoitteen muunnosta ja kolmannessa dynaamista osoitteenmuunnosta. Kaikki kolme ympäristöä todettiin toimiviksi ratkaisuiksi, jotka mahdollistivat halutunlaisen verkkoyhteyden toimivuuden.

Työssä rakennetut sisäkkäiset järjestelmät osoittavat virtualisointiympäristöjen konfiguraatiomahdollisuuksien monipuolisuuden. Työssä tarkasteltiin myös käytön mukaisen, varausperusteisen ja allokaatiopohjaisen laskutusmallin toimintaa. Työn perusteella voidaan nähdä etäresurssipalvelujen olevan varteenotettava vaihtoehto myös suuremmille laitteistohankinnoille ulkoistamisen kautta. Kustannussäästöjä voi syntyä esimerkiksi laitteiston käyttöasteen nousun ja nopeasti muuttiviin kapasiteettatarpeiisin vastattessa.

Avainsanat	vCloud Suite, etäresurssipalvelu, IaaS, poc

Author	Valtteri Maijala
Title	Producing virtualized IaaS-type of cloud service with VMware vCloud Suite environment
Number of Pages	77 pages
Date	27 Feb 2014
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Information Technology
Instructor	Harri Ahola, Senior Lecturer

Cloud services, providing infrastructure, platform and/or software are increasingly used for modern IT solutions. Consequently, the aim of this thesis was to establish and analyze three different cloud service environments built on nested virtual machines.

After a pilot phase, a production system was constructed on the Dell R710 server. Cloud enabling appliances were imported into the nested vSphere environment, created on the bare metal VMware vSphere 5.5.

Three different environments were developed: 1) direct connection to an external network, 2) port based-network address translation between its own and an external network, and 3) dynamic network address translation through the second environment with the access to the external network.

All of the three established environments enabled good operation of the network connection which is fundamentally important for good operation of cloud services. Virtualized cloud infrastructure could flexibly support multiple network configurations throughout the system.

As a result, cloud services proved to be a viable solution to manage the IT load within the outsourcing of hardware and knowhow. Cost savings can be achieved by the increased adaptability for changing the requirements of hardware resources and with fast-paced changing hardware load within the cluster.

Keywords	vCloud Suite, cloud service, IaaS, poc

Sisällys

Lyhenteet

Appliance	Versio virtuaalikoneesta, joka on yleensä osittain valmis implementoitavaksi ympäristöön.
Broadcast	Yleislähetyksessä lähetetään dataa ennalta määräämättömälle joukolle samassa aliverkossa.
Distributed Resource Scheduler (DRS)	Ominaisuus, joka mahdollistaa kuorman jakamisen ja uudelleensijoittelun automatisoidusti.
dvSwitch	Hajautettu virtuaalinen kytkin.
Dynamic Host Configuration Protocol (DHCP)	Teknologia verkkolaitteiden automaattisen osoittuksen aikaansaamiseksi.
Dynamic Power Management (DPM)	Tehokkaan virransäästön ja sitä kautta kustannussäästöt mahdollistava teknologia.
ESXi	VMwaren hypervisor. Katso hypervisor..
Host Power Management (HPM)	Virtualisointipalvelimen alasajon ja nostamisen ohjaaminen kuorman mukaan mahdollistava teknologia.
Hypervisor	Ohjelmisto, jonka päällä virtuaalikoneprosessit ajetaan.
Infrastructure-as-a-Service (IaaS)	Etäresurssipalvelutarjontakokonaisuus, jossa tarjotaan suoraan infrastruktuuria.
Intel VT-d ja VT-x	Intelin virtualisointia tukeva teknologia.
Isolation address	Eriytysosoitteen, yleensä yhdyskäytävän, avulla toissijainen palvelin määrittelee vikaantuneen tilansa.
iSCSI	Internet Small Computer System Interface eli SCSI-käskyjä verkon yli kuljettava teknologia.
Kernel based Virtual Machine (KVM)	Linuxin ytimen virtualisointimoduuli.

MMU	Memory Management Unit eli prosessorin muistinhallintayksikkö.
Multicast	Ryhmälähetyksessä lähetetään dataa yhdestä monelle tietyn ryhmän sisällä.
Nested virtualization	Sisäkkäisien virtualisointikerrosten mahdollistava tekniikka.
Network partition	Virtualisointiapalvelimen tila, jossa se saa yhteyden tietosäilöön, muttei omaan eriytysosoitteeseensa.
Network isolation	Virtualisointipalvelimen tila, jossa palvelin ei saa yhteyttä tietosäilöön eikä eriytysosoitteeseensa.
Openfiler	Insinöörityössä käytetty BSD-jakelu jaetun iSCSI-levykapasiteetin muodostamiseksi.
On / off premise	Ohjelmiston käyttöpaikan suhde suorituspaikkaan sen ollessa joko paikallinen tai etäresurssisuoritus.
Pay-as-you-go -malli	Käytön mukaisen laskutuksen malli, joka perustuu tosialliseen kuormitukseen eri resurssien suhteen.
Platform as a Service (PaaS)	Etäresurssipalvelutarjoomataso, jossa tarjotaan ohjelmistorajapintaa.
Promiscious mode	Lupa verkkokortille vastaanottaa ja välittää muutakin kuin sille kuuluvaa liikennettä. Käytettävä yhdessä nested-virtualisoinnin kanssa.
Power Usage Efficiency (PUE)	Indeksi konesalin virrankulutuksen jakautumissuhteesta.
Reservation model	Etäresurssipalveluiden laskutusmalli, joka perustuu asiakkaan tekemiin varauksiin resursseista.
Resource pool	Resurssivaranto eli kokonaisuus, jolle varataan käyttöön haluttu kapasiteetti laitteistoresursseja.
Software as a Service (SaaS)	Etäresurssipalvelumalli, joka käsittää yksinkertaisimman ja valmiimman tason palveluita verkosta asiakkaan saataville, esimerkiksi sähköpostin.
Split brain	Tilanne, jossa ryppäässä suoritetaan kahta instanssia virtuaalikoneesta yhden sijaan ympäristön vikaantumisen vuoksi.
Storage Area Network (SAN)	Yleensä räkkiin asennettu korkean saavutettavuuden vikasietoinen levypalvelinjärjestelmä.
Storage DRS	Teknologia, joka mahdollistaa virtuaalikuorman siirron levyn kuormittavuuden suhteen.

Storage vMotion	vMotion-tyyppinen siirto virtuaalikoneen kiintolevyille.
Template	Virtuaalikoneesta koostettu helposti monistettava kuva, johon voidaan liittää tuomisvaiheessa erilaisia konfigurointiskriptejä.
Thermal Design Power (TDP)	Tehoarvo, jolla prosessori on speksattu maksimissaan toimivan.
Thick Lazy Eager Zeroed	Tapa provisioida kiintolevykapasiteettia kirjoittamalla luotu kiintolevykapasiteetti levylle.
Thick Provisioned	Tapa provisioida kiintolevykapasiteettia varaamalla luotu kiintolevykapasiteetti levyltä kirjoittamatta sitä.
Thin Provisioning	Tapa provisioida kiintolevykapasiteettia yliprovisioimalla sitä.
Trunk-portti	VLAN-verkkojen portti, josta liikenne kulkee 802.1q-paketoituna.
vApp	Kokoelma virtuaalikoneita ja niiden verkkorakenteita.
vCloud Director	vCloud Suiten etäresurssipalvelukäyttöliittymä ja yhteennitova komponentti.
vCloud Suite	VMwaren etäresurssipalvelusovellusalustakokonaisuus. Sisältää vSphere-ympäristön.
vhv.allow	Asetus konfiguraatiotiedostossa, jolla sallitaan sisäiset virtualisointikerrokset hypervisorin sisällä.
Virtuaalikone	Prosessi, jota ajetaan hypervisorin päällä.
Virtualisointi	Tapa esittää fyysisiä laitteistoresursseja hypervisorin sisäisille prosesseille siten, että rajatusti ne käyttävät niitä kuin omiaan.
Virtuaalilähiverkot (VLAN)	Virtuaalilähiverkko-tekniikalla voidaan provisioida verkon kokonaiskuvaa ja jakaa se loogisesti pieniin lohkoihin useassa eri sijainnissa.
VMDK	Virtuaalisen kiintolevyn VMwaren tiedostoformaatti.
VMFS5	Tiedostojärjestelmä VMDK-tyyppisille kiintolevyille.
vMotion	Teknologia virtuaalikoneen siirtämiselle fyysiseltä palvelimelta toiselle lennossa.
VMware Player	VMwaren työpöytävirtualisointisovellus.
VMware vCenter	VMwaren vSphere-ympäristön hallintapalvelin.

VMware Chargeback	VMwaren virtuaalikoneiden kustannuksia analysoiva ja laskutusrakenteiden käyttämisen mahdollistava sovellus.
VMware vShield Edge	VMwaren vShield-tuotteen palomuuri-, reititin- ja kytkinsovellus.
VMware Workstation	VMwaren työpöytävirtualisointisovellus.
vmx	virtuaalikoneen konfigurointitiedosto.
VNI	VXLAN Network Identifier on verkon yksilöivä tunniste.
vSphere	VMwaren sovelluskokonaisuus, joka sisältää vCenterin ja ESXi-hypervisorin.
vSwitch	virtuaalinen kytkin vSphere-ympäristössä.
VTEP	VXLAN Tunnel End Point on VXLAN-verkkojen ulostulopiste tätä käyttämättömään.
Virtual Extensible LAN (VXLAN)	Etäresurssipalvelukäyttöön luotu virtuaalilähiverkkojen kaltainen verkkosegmenttien laajennusteknologia.

1 Johdanto

Etäresurssipalveluiden käyttö on nykyään jokapäiväistä ja palveluna ne voidaan tarjota toteuttaa laitteiston, alustan ja/tai ohjelmiston tasolla. Tosiasiassa etäresurssipalvelut ovat hyvä keino tuoda laitteistotasosta lähtien etäresurssipalveluita virtualisoinnin kautta. Riskinä ovat kuitenkin virtualisoinnin mukana tuovat haavoittuvuudet ylimääräisen sovelluskerroksen myötä sekä mahdolliset inhimilliset virheet palvelun tarjoamisessa. Siten on erityisen tärkeää, että etäresurssipalvelujen hankittaessa ja tarjottaessa on ymmärrettävä niihin sisältyvät riskit ja haasteet sekä mahdolliset kustannusvaikutukset esimerkiksi lisensointirakenteissa.

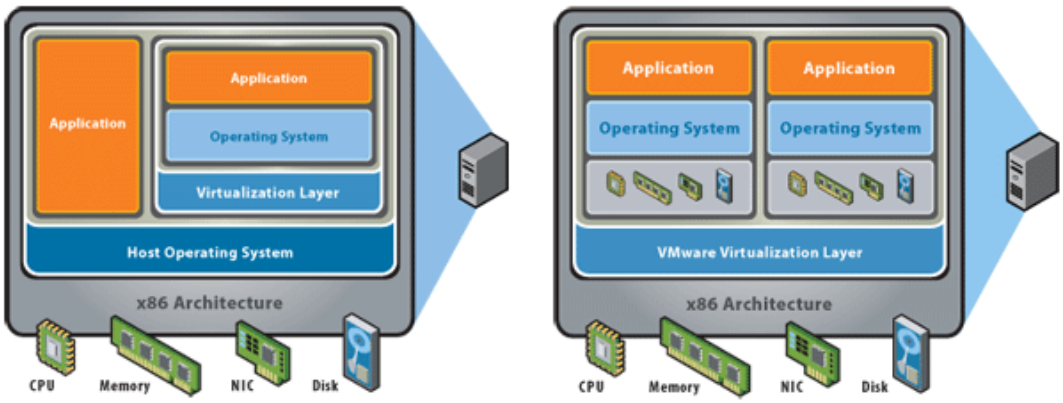
Tämän insinöörityön tavoitteena on tarkastella etäresurssipalvelumalleja, niiden laskutusrakenteita ja toteuttaa laitteistotason etäresurssipalveluratkaisu virtualisoituna käyttäen sisäisiä virtualialisointikerroksia. Työ toteutetaan VMware vCloud Suite -tuotteella. Tällä tuotteella on tarjota laitteistotason etäresurssipalveluita organisaatioille, jotka voivat edelleen tarjota palveluitaan alemman tason palveluina, esimerkiksi sovellustasolla.

Investointipäätösten edessä on otettava huomioon etäresurssipalvelurakenteiden moninaiset kustannusrakenteet. Lisäksi on huomioitava edellytykset etäresurssipalveluiden tehokkaasta käytöstä eritoten tietotaidon suhteen. Etäresurssipalveluita tarjottaessa on tarjasteltava asiakkaan tarpeiden lisäksi myös omien investointien kustannusvaikutukset kokonaistaloudellisesti.

2 Virtualisointi

Virtualisoinnin tavoitteena on nostaa mahdollisimman suuri osa palveluista, laitteistoriippumattomaksi ja näin mahdollistaa helppo muokattavuus, testaus, eriytys ja siirreltävyys. Virtualisoitavan käyttöjärjestelmän alla pyörivää ohjelmistoa, jonka tavoitteena on tarjo-

ta virtualisoitavalle prosessille laitteiston niin levyn, muistin, väylien kuin prosessorin resursseja, kutsutaan hypervisoriksi. Nyrkkisääntönä virtualisointi ja laitteiston resurssien jakaminen suoritetaan joko ohjelmistopohjaisesti ytimen tai laitteistolla hypervisorin avulla, jotka on erotettu kuvassa 1. Kuvassa vasemmalla on tyypin 2 hypervisor eli käyttöjärjestelmän sisällä suoritettava prossessi. Oikealla kuvassa on tyypin 1 hypervisor, joka suoritetaan ilman alemman käyttöjärjestelmän tukea.



Kuva 1. Ydinpohjainen virtualisointi ja laitteistovirtualisointi eroavat virtualisointikerroksen sijainnin suhteen toisistaan kuvan mukaisesti [1].

Ohjelmistopohjainen virtualisointi

Ohjelmistopohjaisessa virtualisoinnissa virtualisointia suorittava käyttöjärjestelmä näkee suoraan laitteistoresurssit, mutta niiden käyttö virtuaalikoneelle on rajoittuneempaa. Tämä etäisempi taso laitteistoresursseista lähes rinnastaa ohjelmistopohjaisesti virtualisoidun järjestelmän perinteisiin käyttöjärjestelmissä ajettaviin ohjelmistoihin. Ohjelmistopohjaisessa virtualisoinnissa hypervisor on käyttöjärjestelmän sisäisesti suoritettava prosessi siinä missä muutkin prosessit eikä sillä ole tyypillistä erityisasemaa järjestelmässä.

Laitteistopohjainen virtualisointi

Hypervisorin täysimittainen toiminta edellyttää laitteistolta asianomaista tukea. x86-ympäristössä 64-bittisiä virtuaalikoneita suoritettaessa edellytetään laitteistolta VT-x -tyypin tukea.

Usean sisäkkäisen virtualisointikerroksen luominen (nested virtualization) edellyttää [2] aiemman tuen lisäksi EPT-tyyppistä tukea. Sisäkkäisessä virtualisoinnissa mahdollisestetaan virtuaalikoneiden ajaminen virtuaalikoneiden sisällä. Tämä on erityisen hyödyllistä testattaessa erilaisia konfiguraatioita esimerkiksi verkkoteknisistä tai ohjelmistoteknistä syistä. Koko insinöörityö on toteutettu useampaa sisäkkäistä virtualisointikerrosta käyttäen. Perinteisten virtualisointiratkaisujen osalta virtualisoinnin ei katsota edellyttävän mitään erityisiä verkkolaitteiden tai levyjärjestelmän osalta. Laitteiston tukiessa voidaan virtuaalikoneelle osoittaa omia osoitettuja laitteita. Laitteiden osoitus yksittäisen virtuaalikoneen käyttöön on ominaista grafiikkaan, levyjärjestelmiin ja verkkoyhteyksiin keskittyneillä lisälaitteilla. Virtualisoinnin näkökulmasta prosessorin virtualisoinnissa on kaksi erityistä ominaisuutta, jotka vaikuttavat sen hyötysuhteeseen ja sisäiseen monimutkaisuuteen: sisäiset käskyt (Virtual Instruction Set) ja muistinhallintayksikkö (MMU, Memory Management Unit).

Prosessorin virtualisointi

Laitteistovirtualisointia prosessorin kannalta voidaan toteuttaa [3] täysin ohjelmistopohjaisesti emulaationa, jolloin jokainen operaatio, jonka virtuaalikone suorittaa, koodataan hypervisorin tasolla uudelleen. Tällä tavoin saavutetaan yhteensopivuus virtualisoitavaa käyttöjärjestelmää muokkaamatta ja eristetään virtuaalikoneet toisistaan. Tämä aiheuttama ylimääräinen kuormitus verrattuna natiiviin ei-virtualisoituun ympäristöön verrattuna on huomattava. Haluttaessa hyötysuhteeltaan parempaa virtualisointia, voidaan virtualisoinnissa käyttää laitteiston mukana tuomia käskykantapohjaisia laajennoksia. Tällaisia ovat esimerkiksi Inteliltä VT-x ja VT-d. Nämä mahdollistavat hypervisorin tehokkaamman toiminnan sallimalla hypervisorin päästää ei-erityinen (User mode) osa koodista suoraan laitteistolle suoritettavaksi ilman binäärikoodausta. Osa suoritettavasta koodista, joka edellyttää kohotettua toiminnallisuutta (Priviledged mode) edelleenkin suoritetaan hypervisorin kautta.

Tietokoneen sisällä toimivat ohjelmat ja prosessit ovat erotettu toisistaan eriasteisiin turvallisuustasoihin. Prosessin (1) vaatiessa kohotettua toiminnallisuutta korkeamman toi-

minnallisuuden vaativa systeemikutsu kaapataan (1) käyttöjärjestelmän toimesta. Seuraavaksi vaihdetaan (2) prosessori ydinmoodiin ja siirrytään suorittamaan kaapattua toimintoa. Tämä toiminto suoritetaan (3) käyttöjärjestelmän ytimen korkeammalla tasolla (4) ja lopulta palautetaan alkiuperäiselle tasolle prosessorin moodin vaihtuessa alemman tason toiminnallisuuteen (5). Lopuksi prosessi jatkaa toimintaansa omalla tasollaan ilman prosessorin tilan vaihtumista kunnes tähän taas on tarve. Tätä on hahmotettu kuvassa 2.

Process	Hardware	Operating System
1. Execute instructions (add, load, etc.)		
2. System call: Trap to OS		
	3. Switch to kernel mode; Jump to trap handler	
		4. In kernel mode; Handle system call; Return from trap
	5. Switch to user mode; Return to user code	
6. Resume execution (@PC after trap)		

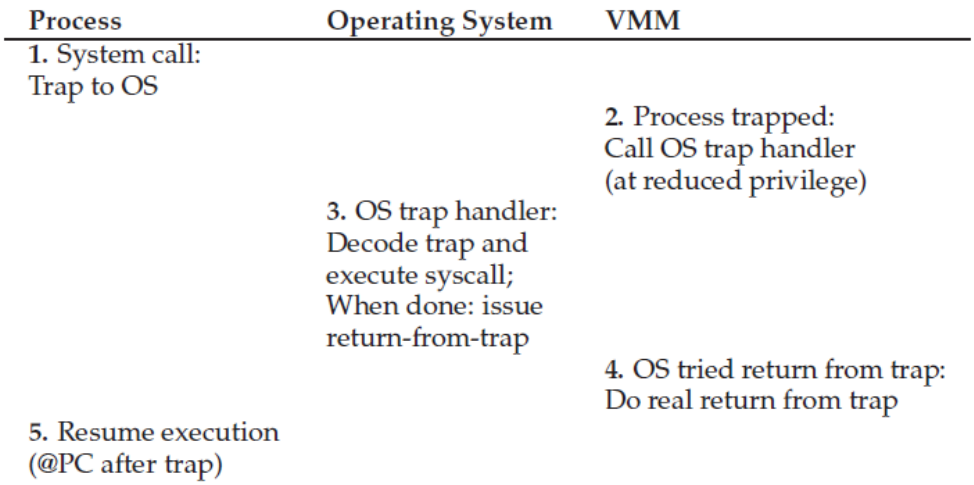
Kuva 2. Systeemikutsun suoritus [4].

Systeemikutsun kulku on virtualisoimattomassa ympäristössä kolmiportainen kuvan 3 tapaan. Systeemikutsun kaappauksen (1) jälkeen käyttöjärjestelmä purkaa ja suorittaa komennon korkeammalla tasolla (2), jonka jälkeen prosessi palaa alkutilaansa (3).

Process	Operating System
1. System call: Trap to OS	
	2. OS trap handler: Decode trap and execute appropriate syscall routine; When done: return from trap
3. Resume execution (@PC after trap)	

Kuva 3. Systeemikutsun suoritus virtualisoimattomana [4].

Virtualisoidussa ympäristössä, kuvassa 3, systeemikutsu kaapataan täysin samanlaisesti kuin virtualisoimattomassa vaihtoehdossa, mutta kaappaaminen tapahtuukin hypervisorin (2) puolesta. Tämä siirtää kaapatun komennon virtualisoidulle käyttöjärjestelmälle (3), missä komento puretaan ja suoritetaan. Tämän jälkeen prosessi pyrkii palautumaan normaaliin tilaan, joka kuitenkin edelyttää hypervisorin toimimista (4) välissä, minkä jälkeen prosessi on toimittanut korkeamman tason toimintansa (5).



Kuva 4. Systeemikutsun suoritus virtualisoituna [4].

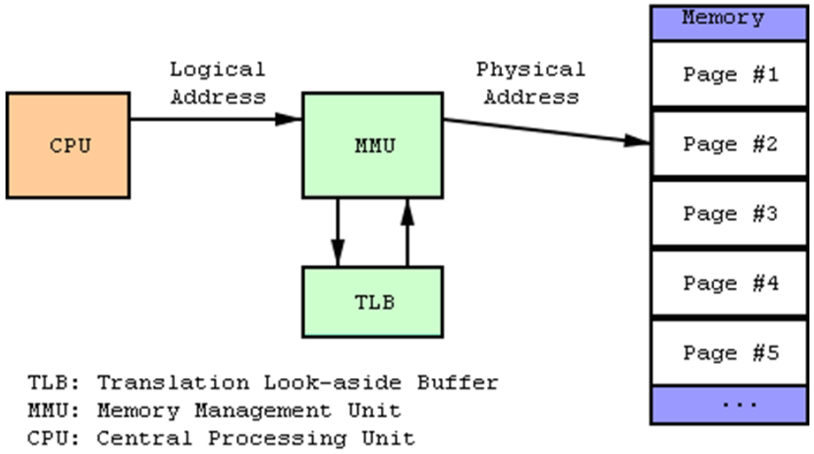
Edellä todettiin virtualisoinnin monimutkaistavan käyttöjärjestelmän toimintaa laitteiston kannalta huomattavasti ja kolmivaiheisesta toiminnosta saadaankin virtualisoituna viisivaiheinen. Vastaavasti muistin virtualisoinnin kanssa, mitä käsitellään hieman edempänä, kuormituksen muutos on kolmesta vaiheesta peräti seitsemään.

Virtuaalimuisti

Haltsonen & Rautasen Tietokonetekniikan mukaan [5] tietokoneessa toimivilla prosesseilla on niiden näkökulmasta yhtenäinen jatkuva muistiavaruus. Tosiasiasassa prosessien näkemä ja käyttämä muisti on virtuaalimuistia. Virtuaalimuisti mahdollistaa prosesseille jatkuvan osoiteavaruuden käytön ja ulkoistaa erityisemmän muistinhallinnan ulos prosessilta käyttöjärjestelmälle ja muistinhallintayksikölle. Käytöllä on myös tietoturvan tasoa nostava vaikutus. Lisäksi sivutuksella, mitä sivutaan myöhemmin, saatetaan käyttää tosiasiallista muistikapasiteettia suurempia muistikapasiteetteja. Virtuaalimuisti on jo-

ko tietokoneen keskusmuistissa tai heittovaihtomuistissa. Muistinhallintayksikkö (Memory Management Unit) on suunniteltu hallitsemaan osoitteenkuvausta ja muistinsuojausta. Näitä on suunniteltu sekä sivutetuille että segmentoiduille virtuaalimuisteille. Prosessorin muistinhallintayksikkö koostuu [3] kahdesta pääosiosta: välimuistina toimivasta 5 TLB-välimuistista (Translation Lookaside Buffer) ja Page Table Walkerista (PTW). Laitteistossa toimivassa tietokoneessa on kaksi muistia: näennäismuisti (Virtual Address Space, VA) ja fyysinen muisti (Physical Address Space, PA). Näennäismuisti koostuu päämuistista eli tässä kontekstissa keskusmuistista (RAM) ja heittovaihtomuistista eli tukimuistista (swap). Eritasoisten muistien käyttämiseksi on oltava osoitteenkuvausalgoritmi ja korvausalgoritmi. Osoitteenkuvausalgoritmi tarjoaa keinon kuvata, missä haettu tieto on. Korvausalgoritmin tehtävänä on minimoida tarpeettomat tiedonsiirrot.

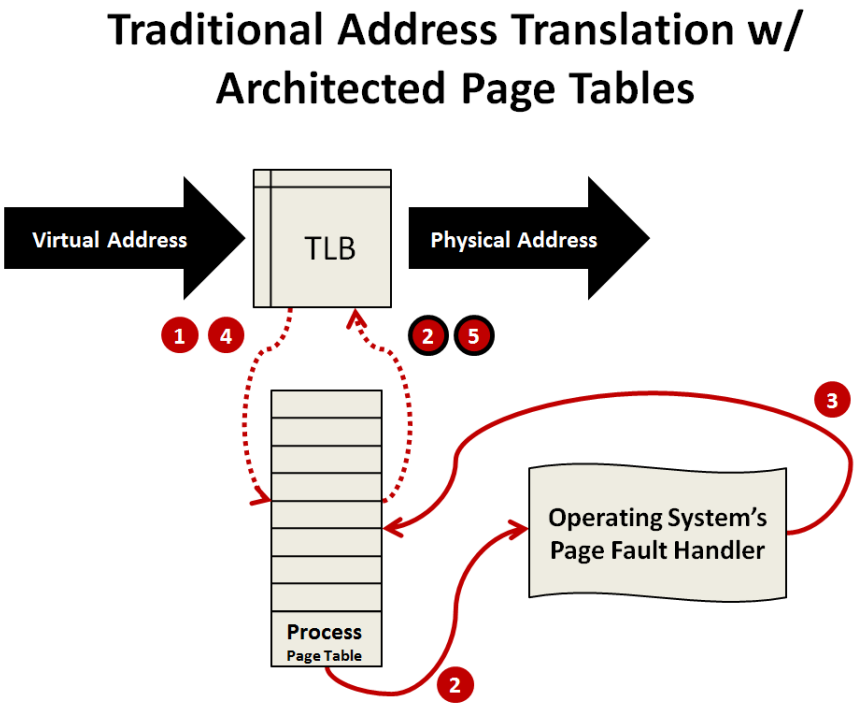
Virtuaalimuistia voidaan joko sivuttaa tai segmentoida sen käyttämiseksi. Sivutetussa virtuaalimuistiratkaisussa virtuaalinen osoiteavaruus jaetaan riveihin ja samanmittaisiin sivuihin eivätkä kaikki sivut välttämättä mahdu keskusmuistiin. Mikäli ne eivät mahdu keskusmuistiin, kirjoitetaan ne heittovaihtomuistiin, joka sijaitsee kiintolevyllä. Siitä sijaitseeko virtuaalimuistin sivun sisältämä sana heittovaihtomuistissa vai keskusmuistissa, vastaa TLB-puskuri (Translation-Lookaside Buffer) ja sivutaulu. TLB-puskurissa on tiedot ainoastaan osaa päämuistia koskevista sivuista ja sivutaulussa on tieto koko heittovaihtomuistin ja keskusmuistin sisällöstä. Virtuaalimuistin sivun sisältö keskusmuistista tai heittovaihtomuistista saadaa käyttämällä sivua hakuavaimena sivutauluun ja TLB-puskuriin. Näissä on virtuaalisivuosoitteita kohden sana. Mikäli sanan sisältö on keskusmuistissa, sanan sisältö on keskusmuistin sivuosoite. Mikäli taas ei, sanan sisältö on heittovälimuistin osoite sanan sisällölle. Sivuvirhe havaitaan, kun virtuaalisivun sanan sisältö päätyy ainoastaan heittovaihtomuistiin eikä sitä ole keskusmuistissa. Tällöin sisältö siirretään keskusmuistiin, tarvittaessa korvaten jo siellä oleva sivu, ja sivu on käytettävissä. Tästä korvaamisesta vastaa korvausalgoritmi. Korvausalgoritmin mukaan sivu, johon on tehty muutoksia sen ollessa keskusmuistissa, täytyy se kirjoittaa uudelleen heittovälimuistiin. Tämä tieto löytyy TLB-muistista. Sivuttamisen sijaan virtuaalimuistia voidaan segmentoida. Tällöin virtuaalimuistiin luodaan erimittaisia segmenttejä ohjelmistojen toiveiden mukaisesti. Segmentoidun virtuaalimuistin sisäinen hierarkia muodostuu segmenttinumerosta ja siirroksesta segmentin sisällä. Segmentoidun muistin kanssa toimitaan samoin kuin sivutetun muistin: mikäli segmentin sisältö ei ole keskusmuistissa, se siirretään sinne heittovälimuistista.



Kuva 5. TLB-välimuisti suhteessa keskusmuistiin [6].

Segmentoinnin ongelmana kuitenkin on segmenttien ei-vakiintunut pituus, jolloin ei voida olla aivan varmoja, mahtuuko segmentti sille toivottuun sijaintiin vai ei. Lisäksi segmenttien väliin jää heikolla käytöllä olevia tyhjiä muistisegmenttejä, joiden käyttäminen onkin huomattavan haastavaa.

Kun prosessi haluaa käsitellä virtuaalimuistissa sijaitsevaa dataa, MMU (Memory Management Unit) tarkistaa viimeisimpien muistioperaatioiden TLB-välimuistista onko fyysisen muistin osoite saatavissa. Mikäli näin on, muistin käyttön edellytykset täyttyvät osoitteistuksen suhteen ja prosessin tarvitsema tieto haetaan fyysisen osoitteen perusteella. Mikäli kuitenkaan TLB-välimuistissa ei kyseistä VA-PA -paria löydy (1), käynnistyy MMU:lla PTW-toiminto, joka käy sivutustiedoston ja PA-muistin lohkot yksitellen läpi ja lopulta yhdistää (ylempi 2) virtuaalimuistin sisällön fyysiseen muistiin tai antaa virheen (alempi 2). Tätä on hamoteltu kuvassa 6. Muistinhakuprosessi uudelleen käynnistettäessä virtuaalisen ja fyysisen muistin sisällöt ovat yhdistetyt (3). Mikäli kuitenkaan fyysisestä muistista ei löydy haettavaa sisältöä, MMU suorittaa keskeytyksen (4) (interrupt) ja käyttöjärjestelmä hoitaa [7] muistin hallinnan eteenpäin tavallaan sivutustiedoston kautta täydentämällä keskusmuistin sisällön ja TLB-muistin sisältö on eheä (5).



Kuva 6. TLB-välimuistin toiminta [8].

2.1 Virtualisoinnin keinot ja tavoitteet

Virtualisointipäätösten tullessa eteen yrityksellä on jo todennäköisesti olemassa olevaa tietoverkkoinfrastruktuuria, palvelimia sekä prosesseja, joita se pyrkii suorittamaan. Näiden prosessien suorituskyvyn nostaminen tai palveluasteen kasvattaminen, kuten investointien yleensäkin, ovat pääjohdonnaiset syyt virtualisointiratkaisuiden tarkasteluun. Kokonaisvaltaiseen virtualisointiratkaisuun siirryttäessä kuitenkin on otettava huomioon virtualisoitaviksi suunniteltujen prosessien erityispiirteet, kuten esimerkiksi jo valmiiksi olevat ominaisuudet, jotka virtualisoinnilla saavutettaisiin, yhdistettynä suuriin kuormituspiikkeihin joko keskusmuistin tai suoritustehon kannalta. Sikäli kun virtualisointi-investoinnin reaalinen tuottoaste on alle yhden, investointia ei pitäisi tehdä. Investoinnin tuottavuuteen alentavasti vaikuttaa esimerkiksi muuttuva lisensointirakenne sen perustuessa fyysisen palvelimen ominaisuuksiin virtuaalisen palvelimen sijaan siinä tapauksessa, että vikasietoisessa ryppäässä ajettaessa lisensointimallin mukaisesti kulut lasketaan kokonaisryppään prosessorikantamäärästä eikä virtuaalikoneen ominaisuuksista, kuten esimerkiksi joissakin Oraclen tuotteissa on tapana ollut. Kaavan 1 investoinnin tuottokertoimen, ja sitä kauttasijoitetun pääoman tuoton (Return on Investment, ROI), on oltava yli yhden, jotta

investointi on tuottava eli kokonaistaloudellisesti positiivinen.

$$\frac{\text{Investointikustannus}}{\text{Laskennallinen tuotto investoinnista}} = \text{Investoinnin tuottokerroin} \quad (1)$$

2.1.1 Vikasietoisuustoimet, palvelun jatkuvuus ja konesalin tehostustoimet

Korkea saatavuus ja vikasietoisuus

Korkea saatavuus toimii [9] virtualisointipalvelimien välisen ensisijaisen ja toissijaisen virtualisointipalvelimen välisesti. Ensisijaisessa roolissa oleva palvelin tulkitsee ympäristön tilan ja välittää sen vCenterille. Ensisijainen palvelin määritetään ryppään luontivaiheessa automaattisesti tyypillisesti tietosäilöjen lukumäärän perusteella. Tyypillisesti ensisijaisia palvelimia on yleensä yksi kappale. Se määrittelee saatavuuden erityisen sykkeen (Heartbeat) perusteella palvelun saatavuutta niin verkon kuin tiedon taltionnin kannalta. Sykettä tarkastellaan oletusarvoisesti yhden tai kahden tietosäilön tason sykkeen ja yhden verkkosykkeen kautta. vSphere sallii useamman tietosäilön ja eriytysosoitteen (isolation address) asettamista käsin ryppään asetuksissa. Eriytysosoite on oletusarvoisesti palvelimen yhdyskäytävä.

Ensisijaisella palvelimella on neljä päävastuuta ryppäässä. Ensimmäisenä se tarkkailee virtualisointipalvelimien tilaa ja tarpeen tullen uudelleen käynnistää kadotetun palvelimen ajetut virtuaalikoneet. Toisena tarkkaillaan vikasietoistettujen virtuaalikoneiden päälläoloa ja kolmantena ylläpitää listaa ryppään palvelimista ja vikasietoistetuista virtuaalikoneista. Viimeiseksi ensisijainen palvelin antaa vCenterille keskitetyn rajapinnan ryppään hallitsemiseen sekä raportoitiin ryppään tilan osalta. Toissijainen palvelin puolestaan vastaa paikallisten virtuaalikoneiden tilan valvomisesta sekä sen päivittämisestä ensisijaiselle palvelimelle. Ryppään vikasietoisuusominaisuudet aktivoituvat, jos jokin seuraavista virhetiloista toteutuu: palvelin sammuu, palvelin joutuu verkon suhteen eristyksiin (network isolation) tai menettää yhteyden ensisijaistason palvelimeen (network partitioned). Virhetiloja

tarkastellaan sykkeen perusteella kerran sekunnissa. Sykkeen menetettyä ensisijainen palvelin tekee tietosäilön ja hallintaverkon kautta tehtävän testin menetettyä palvelinta kohden. Vastaavasti toissijaistason palvelin pyrkii tarkastamaan, onko se eristynyt ensisijaisesta palvelimesta ainoastaan verkon vai myös tietosäilön tasolla.

Tietosäilötason testissä tarkastetaan onko tietosäilössä erityistä syke-tiedostoa ja koska se on päivitetty. Hallintaverkon testissä ensisijainen palvelin tavoittelee yhteyttä ping-komennolla toissijaisen palvelimen hallintaverkon verkko-osoitteeseen. Toissijainen palvelin kohdistaa ensisijaisesta palvelimesta poiketen ping-komentonsa eriytysosoitettaan kohti. Mikäli palvelin havaitsee itse olevansa sekä tietosäilön että verkon kautta saavuttamattomissa, oletusarvoisesti palvelin ja virtuaalikoneet jätetään käyntiin. Uudelleen käynnistetäessä saavuttamattomat virtuaalikoneet uudella palvelimella samasta virtuaalikoneesta on tosiasiallisesti käynnissä kaksi instanssia. Tämä niin sanottu Split Brain -tilanne ratkeaa VMFS-tiedostojärjestelmän lukitustiedoston avulla. Uuden palvelimen ottaessa haltuunsa uudelleenkäynnistettävät virtuaalikoneet se myös ottaa haltuun virtuaalikoneiden tietosäilöt. Tästä syystä ryppääseen palaava vanha palvelin ei kykene niitä siten käyttämään. Tämä estää tiedon korruptoitumista sallimalla vain uusimman palvelimen käyttää tietosäilöä. Ryppääseen palaamisen jälkeen palanneen palvelimen virtuaalikone sammutetaan.

Jatkuva saatavuus tukee myös ohjelmiston ja sovelluksen tasolla toimivaa vikasietoisuutta. Käyttöjärjestelmän tasolla VMware Tools -työkalut tarkkailevat virtuaalikoneen ja ESXi-palvelimen välisiä sykkeitä sekä I/O-laitteiden aktiivisuutta. Mikäli sykettä ei vastaanoteta, oletusarvoisesti kahden minuutin, aikana, virtuaalikone käynnistetään uudelleen. Ohjelmistotason vikaisietoisuuden tarkastelu edellyttää yhteensopivan ohjelmiston käyttämistä. Kuten virtuaalikoneen tasolla, mikäli vikasietoistetun ohjelmiston sykettä ei havaita, uudelleen käynnistetään virtuaalikone. Oletusarvoisesti virtuaalikone käynnistetään uudelleen kolme kertaa aikayksikössä riippuen virtuaalikoneen uudelleenkäynnistysparametrin tasosta. Palvelimen tai palvelun, mikäli vikasietoistus on virtuaalikoneen tai sovelluksen tasolla, keskeytyessä uudelleenkäynnistetään virtuaalikone tarvittaessa. Virtuaalikoneiden käynnistämisen priorisointi kategorisoidaan nousevasti poistetun (disabled) tasolta matalan (low) ja keskitason (medium) kautta korkeaan (high). Tämä mahdollistaa uudelleenkäynnistyksen tilanteissa kriittisten virtuaalikoneiden priorisoinnin vähemmän kriittis-

ten edelle mahdollistaen halutunlaisen palautumisen ongelmatilanteissa. Ääritapauksissa, kuten resurssien käydessä liian niukiksi, alemman prioriteettitason virtuaalikoneita ei esimerkiksi saateta käynnistää lainkaan.

vSphere-ympäristössä voidaan myös ajaa kahta identtistä virtuaali-instanssia siten, että ne toimivat toisista erillään, mutta toimittavat samaa funktiota. Tämän ominaisuuden nimi on Fault Tolerance (FT). FT:ssä vikasietoistettavasta virtuaalikoneesta luodaan identtinen kopio, molemmille syötetään tismalleen samat käskyt, komennot ja toiminnot sekä siten myös toiminnot. Toimittaessa FT-ominaisuuden kanssa ryppäässä on oltava vähintään kaksi mahdollisimman samanlaista palvelinta muun muassa ESXi-version, laitteiston ominaisuuksien - mukaan lukien prosessoriperhe ja kellotaajuudet - ja vikasietoistettavan virtuaalikoneen tarvitsemien resurssien kannalta. Muodostettaessa FT-tyyppisesti vikasietoistettuja virtuaalikoneita on tarkastettava ryhmäyttämissäännöstöjen vaikutukset molempiin virtuaalikoneisiin nähden. Pahimmassa tapauksessa väärin asetetut säännöstöt saattavat estää sen toiminnan, jos molemmat virtuaalikoneet eivät voi sijaita ryppäässä säännöstön mukaisesti.

Jatkuvuus

Virtualisointialustana vSpheren ydinominaisuutena on vMotion-teknologia, joka mahdollistaa haluttaessa virtuaalikoneen siirtymisen yhdeltä palvelimelta toiselle keskeyttämättä virtuaalikoneella toimivia prosesseja ja aiheuttamatta palvelun olennaista hidastumista. Virtuaalikone voidaan siirtää palvelimelta toiselle myös sammutettuna. vSpheren versiosta riippuen siirtoja voi yhtäaikaisesti olla kahdesta kahdeksaan rinnakkain. vMotion edellyttää virtuaalikoneen sijaitsevan jaetun levykapasiteetin päässä molempien siirtoon osallistuvien palvelimien osalta sekä osoitetun vMotion-tason siirtoverkon toimivaa konfiguraatiota. Kriittisten kirjoitus- ja lukuoperaatioiden kanssa toimittaessa voidaan myös optimoida virtuaalikoneen suorituskykyä siirtämällä se sille sopivaan levyvarantoon Storage vMotion -tyyppisellä siirrolla. vSphere 5.1 esitteli mukanaan myös mahdollisuuden toteuttaa sekä virtuaalikoneen että sen kiintolevykapasiteetin siirtämisen ilman käytössä olevaa niin sanottua jaettua yhteistä levykapasiteettia. Näiden käsin hallinnallisten toi-

mien lisäksi virtualisoitu ympäristö mahdollistaa myös joukon automatisoituja toimia, kuten muun muassa virtuaalikoneesta kahden identtisen virtuaali-instanssin ajon rinnakkain kahdessa fyysisessä eri palvelimessa sekä palvelunkeskeytymisdiagnostiikan siten, että virtuaali-instanssi uudelleenkäynnistetään, mikäli sitä suorittava palvelin, virtualikone tai siinä suoritettava ohjelmisto on saavuttamattomissa asetetun ajan. Kaikki nämä ominaisuudet ovat omiaan nostamaan muiden yrityksen osa-alueiden tuottavuutta. Kuten kuvasta 7 nähdään, työntekijöiden jokaisesta työtä tuottamattomasta tunnista on huomattava, mikä nopeasti kattaa virtualisointiratkaisujen aiheuttamat kulut vain tätä aspektia ajatellen.

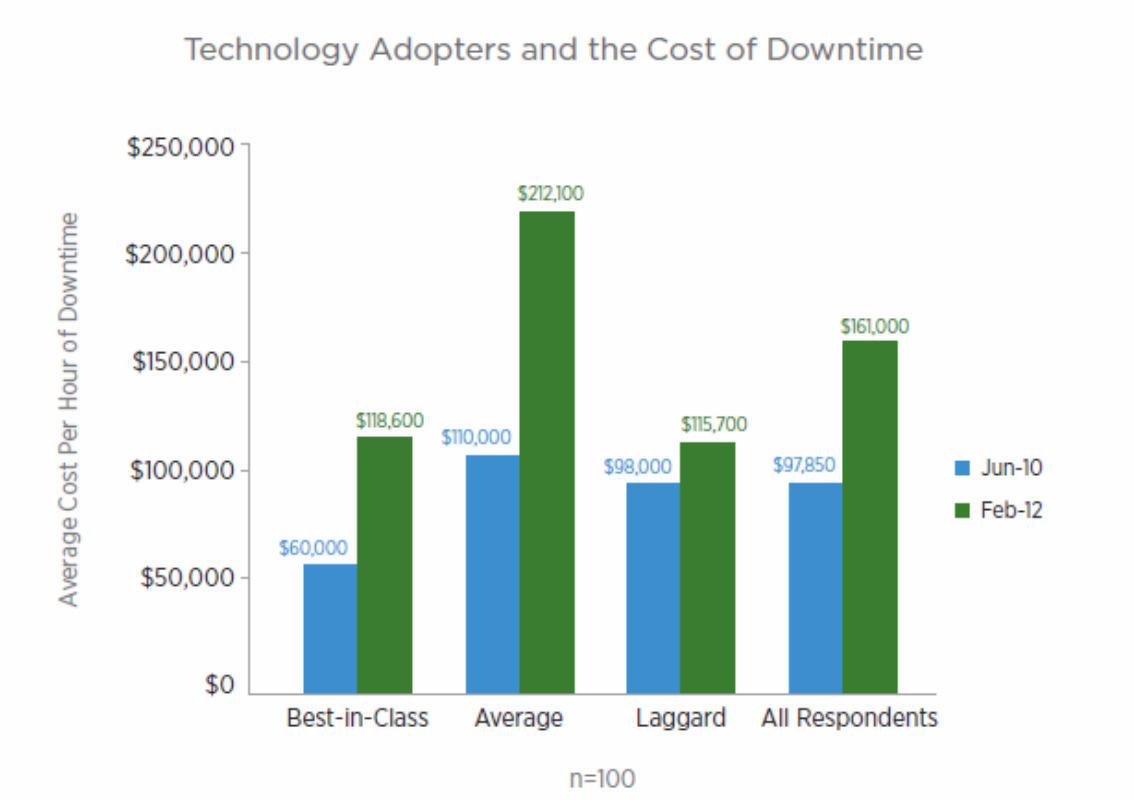
Virtualisoituun infrastruktuuriin siirryttäessä on erityisen tärkeää huomioida fyysisten laitteistoresurssien muodostamat rajoitukset. Siirrettäessä fyysisien palvelimien kuormaa virtualisoiduksi, voidaan teoriassa yhteen palvelimeen laittaa jopa viidenkymmenen pienen palvelimen kuorma. Ongelmia tässä kuitenkin muodostunee virtuaalikoneiden samanaikaisten kuormitusten kumuloivalla vaikutuksella esimerkiksi levyintensiivisissä sovelluksissa. Asiallinen virtualisointisuhde fyysisistä palvelimista virtuaalisiin siirryttäessä voisi olla esimerkiksi vuoden 2010 suosituksen [10] mukaan 16:1 tai jopa 6:1 mikäli virtualisoitava kuorma on poikkeuksellisen vaikeasti ennakoitavaa. Esimerkiksi suhteella 8:1 virtualisoitu SAP-palvelinryppään (kahdenkymmenen palvelimen rypäs pääosaisena käyttöryppäänä sekä kolmen palvelimen ryväs varalla) tehokkuus parani [10] taulukon 1 mukaisesti.

Taulukko 1. SAP-järjestelmän virtualisoinnin avainlukuja [10].

	Lähtötilanne	Virtualisoinnin jälkeen
Keskimääräinen käyttöaste	15 %	70 %
Sähkö- ja jäähdytyskulut	1,0	alle 0,7
Palvelimen toimitusviive	4 - 6 viikkoa	1 tunti

Konesalin ympäristökuorma

Konesaleissa toimivien palvelimien kokonaisenergiatehokkuutta voidaan hahmottaa erityisellä GreenGridin määrittelemällä [12] PUE-indeksiarvolla (Power Usage Efficiency).



Kuva 7. Teknologian omaksujien keskimääräiset saavuttamattomuuden hinnat [11].

Indeksi koostuu palvelinkeskeisten- ja verkkolaitteiden virrankulutuksen suhteesta konesalin kokonaiskulutuksesta ja on siten huomioonottava niin omavaraisen energiantuotannon kuin suoraan verkosta syötettävän virran. Indeksistä siis nähdään, montako energiayksikköä yhtä palvelimen tuottamaa yksikköä kohden tarvitaan sen jäähdyttämiseksi. Esimerkiksi mikäli yhden kilowatin tehoisen palvelimen asianomainen muu lämpökuorma olisi kaksi kilowattia, olisi kaavan 2 indeksin arvo kolme. Esimerkiksi Googlen [12] konesalien indeksi juoksevana kahdentoista kuukauden keskiarvona saavutti vuonna 2013 arvon 1,12. Toisin sanoen yhden kilowatin palvelimen kuorman tuomat lisäkulut ovat olleet ainoastaan 0,12 kilowattia eli kokonaishyötysuhteeltaan palvelinsali toimii 89 %:n hyötysuhteella. Hyötysuhde saadaan ottamalla PUE-arvosta käänteisluku kaavan 3 mukaisesti. Energiantuottomuodoista puhuttaessa voidaan mitata ja verrata myös esimerkiksi veden kulutusta tai uusiutuvien energianlähteiden osuutta kokonaistuotannossa. Näillä luonnollisesti mitataan eri asiaa, mutta ne ovat myös hyvin yksinkertaistettavissa olevia mittareita, joilla voidaan hahmottaa tuotannon osa-alueita. Esimerkiksi hiilikuormitusta ajatellen korkeampi PUE-indeksi voisi olla oikeutettu, mikäli se on tuotettu uusiutuvista energialähteistä, joiden hiilijalanjälki per energiayksikkö on matalampi kuin muiden vertailukohteiden.

$$\text{PUE-indeksi} = \frac{\text{Salin kokonaisteho}}{\text{Verkkolaitteiden ja palvelinten teho}} = \frac{3\ \text{kW}}{1\ \text{kW}} = 3 \quad (2)$$

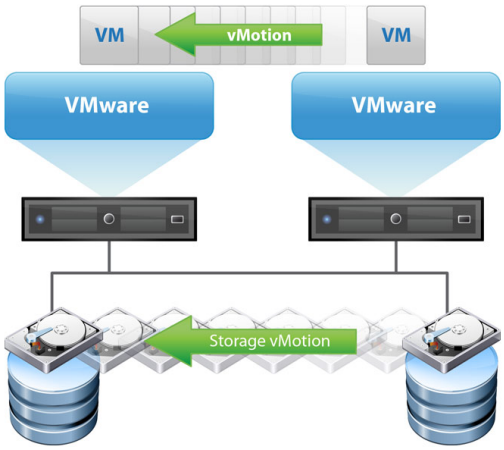
$$\text{Hyötysuhde} = \text{PUE-indeksi}^{-1} = \frac{1\ \text{kW}}{3\ \text{kW}} = 33\% \quad (3)$$

2.1.2 Kuorman siirto ja tasapainotus sekä virransäästöominaisuudet

Virtualisoinnin yhteydessä on luonnollista sijoittaa virtuaalikoneita ja suorituskykyä vaativia sovelluksia tarkoituksenmukaisesti koko ryppään laajuudelle eri palvelinten välillä. Esimerkiksi erityisen matalaa latenssia vaativat sovellukset olisi aiheellista sijoittaa mahdollisimman optimaaliseen asetelmaan suhteessa niiden käyttämiä tietokantapalveluita ja muita sen käyttämiä resursseja. Tähän vastaa VMware vSpheren kuorman tasaukseen tarkoitettu DRS lisäoptioinaan Distributed Power Management (DPM) ja Host Power Management (HPM) kanssa. Nykyaikaisessa modernissa konesalissa palvelinten synnyttämä kokonaiskustannus energiakuluissa on jopa 58 %, kuten eBayn vuonna 2004 rakentamassa [13] Phoenix 1 -salissa Pohjois-Amerikassa.

Distributed Resource Scheduler

Distributed Resource Scheduler (DRS) mahdollistaa kuorman tasaamisen ryppään sisällä siten, että jokainen palvelin olisi asianomaisen kuormituksensa suhteen mahdollisimman optimaalisessa tilassa. Resursseja voidaan rajata DRS-ryppään sisällä erityisellä DRS-allokaatiolla, jossa virtuaalikoneryppäälle ennakolta määritetään kuinka paljon se voi varata ryppään resursseja. DRS:llä on myös mahdollista hallita virtuaalikoneita siten, että virtuaalikoneiden kilpaillessa resursseista ne ohjautuvat toivotunlaisesti oikealle taholle. Riittävän pitkään kilpailun jatkuessa voidaan DRS:n asetuksista riippuen automaatiotoiminnolla siirtää kuormaa toiselle palvelimelle vMotion-teknologialla kuvan 8 mukaisesti.



Kuva 8. vMotion ja Storage vMotionin eroavat toisistaan siirretyn toiminnallisen komponentin osalta [14].

Näin virtuaalikoneiden saatavilla olevat resurssit lisääntyvät ja parantavat niiden suorituskykyä. Kriittisten sovellusten osalta on huomattava mahdollisuus asettaa erityisiä sääntöjä sen mukaan, miten halutaan virtuaalikoneiden ryhmäytyminen ryppäässä hallita: esimerkiksi kahden nimipalvelimen sijainti fyysisesti samalla palvelimella, vaikkakin virtualisoituna, tulisi ehdottomasti välttää. Pahimmillaan fyysisen palvelimen toiminnan riittävästi häiriintyessä myös nimipalvelimen toiminta häiriintyy mahdollisesti aikaansaaden laajempia vaikutuksia verkossa.

Storage DRS ja tietosäilöprofiilit

vSphere 5.0:n lanseerauksen yhteydessä VMware laajensi perinteisen DRS:n toimintamuotoa myös tallennuskapasiteetin puolelle. Storage DRS:n avulla on mahdollista erilaisten tallennusprofiilien (Storage Profile) kautta osoittaa ennalta tietylle käyttäjäjoukolle tietyn tyyppistä levytilaa esimerkiksi varmistusta tai suorituskykyä silmälläpitäen. Storage DRS:n avulla on myös mahdollista tasapainottaa kuormitusta ryppään sisällä mikäli yhden tai useamman tietosäilön kuormitusaste taikka vapaan kapasiteetin ylittää asetetun rajaarvon. Ryhmäkohtainen tietosäilöprofiilihallinta mahdollistaa ryhmille ainoastaan ryhmille osoitettujen levyjärjestelmien käytön sallimisen helposti ja yksinkertaisesti. Lisäksi tietosäilöprofiilit mahdollistavat joustavan ja läpinäkyvän tavan tuoda hinnoittelurakenne myös tiedon tallennuskapasiteetin puolelle keskusmuistin alueelta. Tietosäilöprofiilit ovat olen-

nainen osa myös vCloud Suitea ja osa etäresurssipalveluja ja niiden hinnoittelua. Storage DRS toimii aivan kuten perinteinen DRS keskittyen tietovarantojen kuormitusasteeseen sekä kuormituksen että vapaan kapasiteetin suhteen. Etäresurssipalveluita tuotettaessa on tarpeen antaa asiakkaalle mahdollisuus priorisoida tietosäilökapasiteetteja keskenään. Tämä mahdollistaa asiakkaalle valinnanvapauden siitä, mikä tieto on arvokasta nopeuden suhteen esimerkiksi.

Ryhmäyttämissäännöstöt

Kuormaa tasatessa ja kriittisiä palveluita virtualisoitaessa on erityisen tärkeää erotella virtuaalikoneet siten, etteivät ne muodosta palvelussa heikointa lenkkiä palvelujen tarjoamisen osalta. Näitä kutsutaan ryhmäytymissäännöiksi (affinity rules) Esimerkiksi julkisten nimipalvelimien kohdalle, joita täytyy olla kaksin kappalein oman julkisen osoiteavaruuden kanssa, on luonteva luoda ryhmäsäännöstö, jonka mukaan nimipalvelinten on sijaittava erillisillä fyysisillä virtualisointipalvelimilla. Toisaalta voidaan myös erityisen matalaa viivettä tai suurta kuormitusta, esimerkiksi verkkoliikenteenosalta, edellyttää ryhmäyttämistä samalle palvelimelle minimoidakseen haitan muulle ympäristölle.

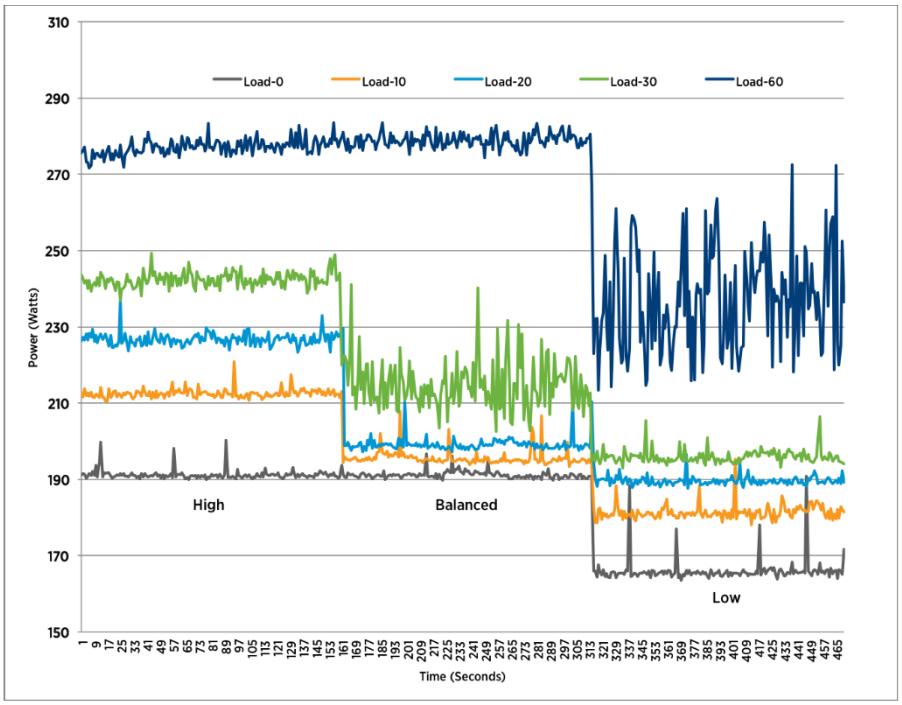
Virransäästöominaisuudet

VMware vSpheressä ympäristössä on mahdollista konfiguroida Distributed Power Management (DPM) -virransäästöominaisuus, jonka tavoitteena on DRS-ryppäässä kontrolloida fyysisten palvelinten virtojen hallintaa. DPM toimii normaalin DRS:n kuormantasauksen rinnalla joko sammuttaen tarpeettomia DRS-ryppään reservikynnyksen ylittäviä palvelimia taikka nostaen uusia palvelimia ryppään käytettäviksi resursseiksi. Kustannussäästöt DPM:n käytöstä syntyvät palvelinten vuosittaisen virrankulutuksen vähenemisestä, PUE-arvon sitä ilmentäessä sekä laitteiston yleisen rasitustason laskemisesta ja sitä kautta käyttöiän noususta. DPM:n luonnollisena heikkoutena on muutamien minuuttien reagointiaika.

Haluttaessa DPM:n hyödyt, muttei alasajon aiheuttaman vasteajan haittoja vastaavissa määrin, on mahdollista käyttää VMware vSphere 5.0- ja uudemmissa toimivaa Host Power Management (HPM)-ominaisuutta. Host Power Management jättää fyysisen palvelimen päälle ja pyrkii laskemaan kustannuksia laskemalla prosessorin kellotaajuutta. Prosessorin kellotaajuuden lasku on vain yksi toimenpide, jolla kyetään laskemaan kokonaislämmöntuottoa ja virrankulutusta prosessorilta.

vSpheren tukeman ja käyttämän [15] ACPI-standardin (Advanced Configuration and Power Interface) mukaisesti [16] yhteensopivalla käyttöjärjestelmällä on mahdollisuus asettaa laitteisto esimerkiksi erilaisiin virransäästötiloihin laitteiden tai prosessorin osalta. Laitteistotasolla tunnettu tila on esimerkiksi S4-hibernaatiotila, jossa keskusmuistin sisältö kirjotetaan levylle ja laitteisto sammutetaan. C-tyypin tilat ovat prosessoriin ydinkohtaisesti kohdistuvia, joista esimerkiksi voidaan mainita C0, jossa prosessori on käynnissä normaalisti ja C1-tila, jossa prosessori ei kysesellä hetkellä suorita käskyjä, mutta on palautettavissa välittömästi käyttökuntoiseksi. Virtualisoinnissa voidaan käyttää vielä tätä C1-tilaa tehostetumpaa C1(E)-tilaa. Runsaasti yksisäikeisiä suoritteita tehtäessä sekä muutamia harvoja poikkeuksia lukuunottamatta alemman tason C1(E)-virransäästötila on kokonaistaloudellisesti edukkaampi kuin koko prosessorin pitäminen kaikkine lohkoine alemmalla kellotaajuudella. Prosessorin tukiessa erityistä Turbo boost -teknologiaa, jossa prosessorin kellotaajuutta nostetaan ohjekellotaajuuden ylitse lämmöntuoton puitteissa C1(E)-tilasta on merkittävää hyötyä, sillä näin prosessorissa ovat päällä ainoastaan aktiiviset ja kuormaa ajavat osiot. Näin saavutetaan korkeampia kellotaajuuksia pienemmällä lämpökuormalla.

Mikäli virransäästöominaisuudet ovat laitteiston tasolta ohjattuja, kyetään yksittäisiä prosessoriytimiä sammuttamaan. vSpheren taholta toimiva virransäästöominaisuus ei siihen kykene. VMwaren tekemän vertailun mukaan [17] vSphere 5.1 tai 5.5 virransäästöominaisuudet ovat jopa 18 % parempia kuin laitteiston omat virransäästöominaisuudet. Toisen testin kuva 9 kokoaa esimerkinomaisesti Dell PowerEdge R710 -palvelimen virransäästön aiheuttamat muutokset sähkön kulutuksessa, jotka vaihtelevat lepotason 13 % säästöstä (Load-0) parhaimmillaan 23 %:n säästöön 30 %:n kuormituksella (Load-30) [15].



Kuva 9. Dell PowerEdge R710 -palvelimen virrankulutusgraafi eri kuormitusasteilla HPM-virransäästömoodissa ja ilman erityisempiä virransäästösäätöjä [15].

Levitys

Käyttäjän ja ohjelmiston virheiden varalle virtualisointiympäristössä on mahdollista tehdä sekä erinäisiä suojaus että monistustoimenpiteitä infrastruktuuria ajatellen. Perinteisessä fyysisessä ympäristössä käyttöjärjestelmät levitetään työasemiin esimerkiksi verkosta levykuva levittämällä erityisen levitysympäristön avulla tai asennusmedialta asentamalla suoraan. Virtualisointiympäristössä virtuaalikoneita voidaan kloonata virtuaalikoneen tilakuvasta, joka on muodostettu kyseisen virtuaalikoneen hetkellisestä olotilasta. Vaikkakin tilakuva säilyttääkin virtuaalikoneen tiedot kyseisellä hetkellä oikein, ei sen käyttö ole perusteltua backup-toimissa. Tilakuvat sopivat hyvin nopeatempoiseen testaukseen ja kehitykseen, jossa nopea palautuminen aiempaan tilaan on erityisen tärkeää. Toinen vaihtoehto on luoda jo olemassaolevasta virtuaalikoneesta virtuaaliaihio (template), joka mahdollistaa virtuaalikoneen luomisvaiheessa aihion muokkaamisen erilaisten yksilöintisovellusten, kuten esimerkiksi Microsoftin esivaiheen kustomointiin tarkoitetun Sysprep:n avulla.

Virtuaalikoneiden koostuvat useista tiedostoista. Täten virtuaalikoneiden tuominen toteutetaan perinteisesti paketoinnin avulla. Työssä on käytetty avointa virtualisointialustariippumatonta OVF-paketointiformaattia (Open Virtualization Format) virtuaalikoneiden tuomiseen ympäristöön. Standardi [18] on muun muassa XenSourcen, Microsoftin ja VMwaren vuonna 2007 esittämä [19], jonka ensimmäinen versio hyväksyttiin syyskuussa 2008. OVF-paketointi sisältää XML-kielisen OVF-kuvaustiedoston, mahdollisesti yhden tai useamman virtuaalikiintolevyn ja muita virtuaalikoneen ulkopuolisia komponentteja, kuten ISO-levykuvia. Paketointiin on mahdollista [19] lisätä todennus- ja integraatiotarkoituksessa .cert-varmennetiedosto sekä .mf-manifestitiedosto esimerkiksi lisenssiehtojen ja muun metadatan lisäksi. Siirreltäessä OVF-muodossa tuotavia virtuaalikoneita, ne on yleensä tapana pakata tar-paketoinnilla OVA-tyyppiseksi (Open Virtualization Appaliance) tiedostoksi. Virtualisointia hyväksikäyttävillä valmistajilla on tapana tuoda suoraan virtualisointialustalla käytettävät tuotteet OVA-paketoituina applianceina eli valmiiksi koottuina virtuaalikonekokonaisuuksina, jotka voidaan pääsääntöisesti suoraan tuoda ja konfiguroida ympäristöön sen kummemmitta asennusrutiineitta. Insinöörityön ohjelmistoista niin vCenter, vCloud Director kuin vShield Manager on kukin tuotu ympäristöön appliance-tyyppisessä muodossa.

Testaus

Virtualisoitu ympäristö mahdollistaa laajan valikoiman erilaisten laitteistokokoonpanojen ja käyttöjärjestelmien rinnakkaisen testaamisen. Nopea palautuminen muutoksista tilakuvien avulla, uusien levykuvien tuominen ja muokkaaminen automatisoidusti ovat vain osa tätä virtualisoinnin mukana tuomaa joustavuutta testaustarkoituksissa. vSphere-ympäristö tukee myös lennosta lisättäviä ja poistettavia lisälaitteita, kuten prosessoreita, verkkokortteja tai keskusmuistia.

2.2 Virtuaalikoneen komponentit

Kylmä käynnistämätön virtuaalikone koostuu tiedostoista. Virtuaalikoneen toiminnallisuuden ytimenä toimivat virtuaalikoneen konfigurointitiedosto (.vmx) sekä virtuaalinen kiintolevy (.vmdk). Virtuaalikoneen konfiguraatiotiedostossa kuvaillaan virtuaalikoneen ominaisuuksia laitteistotasolla; esimerkiksi verkkokortit ja niiden MAC-osoitteet, levyohjaimet ja usb-ohjaimet. Virtuaalikiintolevy sen sijaan sisältää tiedostojärjestelmän ja virtuaalikoneen suoraan omassa käytössä olevat tiedot.

Konfiguraatiotiedosto

Virtuaalikoneen konfiguraatiotiedostossa (.vmx) kuvataan muun muassa prosessorien määrä, verkkokortin ominaisuudet, kiintolevyn väyläominaisuudet sekä virtuaalikoneen virtuaalikonfiguraatioversio, joka on erittäin olennaisessa osassa sisäkkäisiä virtualisointikerroksia käytettäessä. Konfiguraatiotiedostossa myös määritellään sovellettava rakenne, jolle suunnitellussa ympäristössä virtuaalikonetta ajetaan.

Virtuaalikiintolevy

Virtuaalikoneen virtuaalikiintolevystä (.vmdk) puhuttaessa täytyy erottaa virtuaalikoneen näkemä kiintolevykapasiteetti faktisesti käytössä olevasta tai käytetystä kapasiteetista levyjärjestelmässä. Virtuaalikiintolevytiedosto voidaan alustaa kolmella eri tavalla. Ensimmäinen tapa on niin sanottu Thin Provisioined -malli, toinen ja kolmas tapa perustuvat Thick Provisioined -malleihin. Etukäteen levytilaa allokoimattomasta eli sitä varaamattomasta Thin Provisioned -mallilla toimista levystä allokoidaan alun pitäen vain virtuaalikiintolevytiedostojen perustamistiedot fyysiselle levylle. Virtuaalikoneen levyä käyttäessä virtuaalikiintolevytiedostot kasvavat samassa suhteessa pitäen sisällään ainoastaan sinne kirjoitetun datan. Edellä mainitut kaksi mallia molemmat varaavat levyltä koko sille virtuaalilevyn luonnissa allokoidun tilan fyysiseltä levyltä ainoastaan eroten toisistaan kehitty-

neempien levyjärjestelmien osalta. Thick Provisioned Lazy Zeroed -tyyppinen virtuaalilevy varaa koko kiintolevykapasiteetin ja normaalissa ei-kehittyneessä ympäristössä varautuu myös koko kapasiteetti fyysiseltä levyltä. Kuitenkin kehittyneempien levyjärjestelmien on mahdollista olla varaamatta edellä mainittujen tyyppisten virtuaalilevyjen viemää tilaa fyysiseltä levyltä. Thick Provisioned Lazy Eager Zeroed -mallilla alustettu virtuaalikiintolevy varaa virtuaalikiintolevyn luomisen yhteydessä ja kirjoittaa sen lähtökohtaisesti täyteen nollabittejä, jolloin edes kehittyneempien levyjärjestelmien ominaisuudet eivät kyseisen tyyppisen levyn varausta kykene poistamaan. Virtuaalikiintolevyjen virtuaalikoneille näkyvät koot sekä niiden tosiasiallisesti kapasiteettia vievä sisältö tulevat olennaisiksi suurien virtuaalikoneiden tai suuren määrän virtuaalikoneita kanssa.

Käynnissä oleva virtuaalikone

Käynnistettäessä virtuaalikone muodostetaan virtuaalikoneen keskusmuistin kokoinen (.vmem) tiedosto sekä virtuaalikoneen hakemistoon että Thin Provision -tyyppisesti palvelimen keskusmuistiin, ellei kyseiselle virtuaalikoneelle ole erityisiä varausehtoja asetettu resurssien varaamisen osalta. Lisäksi virtuaalikoneen hakemistoon muodostetaan erityinen lukitustiedosto (.lck), jolla estetään muiden palvelimien hallinnollisesti tahaton haltuunotto niin virtuaalikoneen kiintolevyn, konfiguraatiotiedoston kuin muidenkin relevanttien komponenttien osalta. Otettaessa tilakuva (snapshot) virtuaalikoneesta sijoitetaan se virtuaalikoneen kanssa samaan kansioon. Tilakuva voidaan ottaa joko sammutetusta tai käynnissä olevasta virtuaalikoneesta. Mikäli se otetaan käynnissä olevasta virtuaalikoneesta, muodostetaan myös virtuaalikoneen keskusmuistista oma tiedosto virtuaalikoneenhakemistoon.

Käynnistysvaiheessa keskusmuistia varataan fyysiseltä palvelimelta ainoastaan virtuaalikoneen tosiasiallisesti käyttämä kapasiteetti. Tilakuva otettaessa virtuaalikoneen kiintolevyn yhteyteen luodaan erityinen muutostiedosto, johon kirjoitetaan tilakuvan jälkeiset tapahtumat. Tämä aiheuttaa tiedon sirpaloitumista yhden yksinkertaisen kiintolevyn muututtua kahdeksi toisistaan riippuvaiseksi. Pitkällä aikavälillä tilakuvilla on siten virtuaalikoneen suorituskykyä laskevia vaikutuksia. Mikäli kahdella toisiaan muistuttavalla virtu-

aalikoneella on identtistä tai ei-aktiivista dataa omissa keskusmuisteissaan ja fyysisen palvelimen keskusmuistin varausaste on korkea, voidaan erityisellä kuristusmenetelmällä (ballooning) pienentää virtuaalikoneen keskusmuistin todellista kuormitusta. Tällöin nämä kaksi virtuaalikonetta jakavat yhteisen muistiosion fyysisen palvelimen keskusmuistissa. Tämä mahdollistaa suuremman suhteen virtuaalikoneita suhteessa niiden viemiin laitteistoresursseihin kuin virtualisoimattomilla palvelimilla.

Sisäkkäiset virtualisointikerrokset

Virtualisoinnissa voidaan myös virtualisoida virtualisointia. On siis mahdollista ajaa fyysisessä palvelimessa esimerkiksi ESXi-hypervisoria, jossa olevassa virtuaalikoneessa on toinen ESXi, joka sisältää kolmannen ESXi:n, joka sisältää neljännen ESXi:n. Tämä sisäkkäinen nested-tyyppinen virtualisointi on omiaan auttamaan erilaisia testausympäristöjä ja -olosuhteita muodostumaan ja mahdollistamaan niiden toteutusta. Sisäkkäisenvirtualisoinnin ansiosta voi kokonaista palvelinsalia virtualisoida täysin toiminnassa ja kaikki ominaisuudet toiminnassa vikasietoisuudesta kuormantasaukseen. Virtualisointiohjelmistojen suhteen VMware Workstation tai ESXi eivät vaadi sen kummempaa modifikaatiota virtuaalikoneelle. Virtuaalikoneen .vmx-tiedostoon on kuitenkin lisättävä erityinen vhv.allow -tagi, jolla sallitaan sisäkkäistyyppinen virtualisointi. Lisäksi alemman tason virtuaalikoneen verkkokortti on asetettava myös muita sallivampaan Promiscuous-tilaan, jolloin kyseinen virtuaalikone näkee ja tarpeen tullen näkee ja käsittelee kaiken liikenteen oman liikenteensä ulkopuolelta.

2.3 Virtualisoinnin haasteet ja riskit

Toteutettaessa monimutkaisia virtualisointiympäristöjä on erityisen olennaista havaita komponenttien keskinäiset vaikuttavuudet. Esimerkiksi mikäli virtuaalikoneet sijaitsevat toisella virtuaalikoneella ajettavassa levyjärjestelmässä, jälkimmäisen kaatuessa ensimmäinenkin virtuaalikone putoaa pelistä pois. Toisaalta virtuaaliympäristössä voidaan siirtää virustorjunnan kuormaa pois virtuaalikoneilta itseltään hypervisorin tasolla toimivalle so-

vellukselle ja tällä tavoin vähentää virustorjunnan rasitusta järjestelmälle. Keskitetyissä ratkaisuissa haasteeksi muodostuu juurikin niiden keskitettävyys: yksi vaikuttaa moneen ja vaikutukset ovat siten laajoja.

Virtualisoidun ympäristön heikkouksiin liittyvät seikat

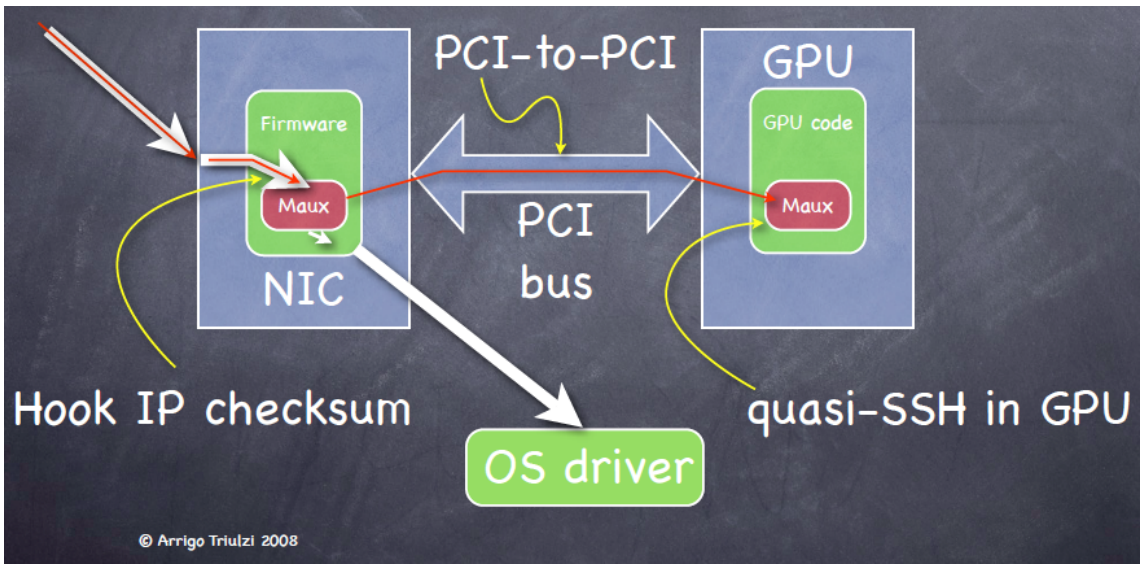
Toteutettaessa monimutkaisia virtualisointiympäristöjä on erityisen olennaista havaita komponenttien keskinäiset vaikuttavuudet. vCloud ympäristössä palomuurin virkaa toimittaa ainoastaan korkean saatavuuden tuella tukevoitettu vShield Edge, jonka toiminnan vaikeuttaminen vaikuttaa kaikkeen, jota kyseisen virtuaalikoneen takana pyörii. Oikeaoppisesti palomuuri- ja reititinratkaisut olisivat kahdennettu ja vikasietoistettu, mutta tähän vCloud Suite 5.1 ei vielä ainakaan tarjoa ratkaisua. Tällöin tosiasiallisesti vShield Edge muodostaa ympäristön Akilleen kantapään ollen yksittäinen komponentti, jonka vaurioituminen lamauttaa ympäristön hetkellisesti. Kuitenkin vShield Edge on mahdollista asettaa vikasietoisesti ainoastaan korkean saatavuuden tilaan, silti ollen yhä selkeä heikkous järjestelmässä verrattuna esimerkiksi selvästi vikasietoistettuihin ratkaisuihin. Huonosti testatut päivitykset ovat ongelma etenkin, jos dominoefektimäisesti vaikutukset ulottuvat huomattavan kauas. Esimerkiksi Microsoftin Hyper-V-ympäristön eräät päivitykset ovat rikkoneet ajoittain [20] virtualisointiympäristön pahasti.

Tietoturva ja vastuukysymykset

Erilaisten virtualisointiratkaisujen kanssa tietoturvasta on huomiotava sekä perinteiset että virtualisoinnin mukana tuomat seikat erityisesti ympäristöä suunniteltaessa. Korkeamman toimivuuden virtualisointiratkaisujen kanssa käytettävistä toiminnoista esimerkiksi korkean saatavuuden (High Availibility) ja kuorman tasauksen (DRS) mukana tuomat ongelmat ovat uusia. Myöskin on huomioitava Guest-to-Host -tyyppiset virtuaalikoneesta hypervisorin resursseihin kohdistuvat hyökkäykset. Näillä tavoin on mahdollista toteuttaa [21] niin sanottu palvelunestohyökkäys hypervisor-tasolla. Tällöin resurssit, joita kyseinen yksikkö on suorittamassa, reagoivat ja aiheuttavat pahimmassa tapauksessa suu-

ria kuormituspiikkejä hypervisorilla ajettavien virtuaalikoneiden käynnistyessä uudelleen sekä tiedon eheytymättömyyden kustannuksia. Optimaalisessa tilanteessa palvelinryppään olisi mahdollista kuitenkin torjua tämä kuormantaustekniikoin taikka korkean käytettävyyden tekniikalla. Mikäli tällaisella palvelunestohyökkäyksellä hypervisorin tasolle olisi mahdollista uudelleenkäynnistää virtuaalikoneita, olisi myös mahdollista mountata ja siten myös kirjoittaa ja lukea kyseisiltä levyiltä mitä tahansa informaatiota aivan kuten normaalia paikallista levyä. Windows-käyttöjärjestelmässä ajettaville VMware Workstation- ja Player-alustoille on kehitetty virus [22], joka pyrkii tartuttamaan virtuaalikoneita perustuen Windowsin haavoittuvuuksiin.

Suoranaisia viruksia ei ESXi taikka muille VMwaren virtualisointituotteille ole raportoitu, mutta muun muassa VMwaren ajureiden heikkouksiin perustuen on mahdollista [23] saada virtuaalikoneesta yhteys sellaiseen esimerkiksi verkkoon, johon ei ollut ylläpitäjän tarkoitus käyttäjää päästää. Osasyynä tähän on ajureiden huono laatu sekä VMwaren tarpeettoman hiljainen suhtautuminen alustansa haavoittuvuuksien paikkaamiseen. Vastaavia haavoittuvuuksia on olemassa myös muille hypervisoreille, kuten esimerkiksi KVM:lle [24]. Toisaalta mikäli pahantahtoinen riittävän kyvykäs taho pääsee laitteistoon fyysisesti käsiksi, eivät erilaiset virtualisointiratkaisutkaan edes auta, jos ongelma on laitteistotasolla, kuten Arrigo Triulzin Project Maux Mk. II -niminen toteutus [25] kertoo esimerkinomaisesti kuvassa 10, jossa kuvataan haavoittuvan verkkokortin firmwareohjelmiston kautta luotua kaapattua ympäristöä. Verkkokortin oman firmware-muistin kapasiteetin vähyyden vuoksi pääasiallisena tiedon tallennuskapasiteettina projektissa käytettiin näytönohjaimen muistia. Maux II toimii tarkistamalla jokaisen verkkokortin läpi kulkevan paketin, ja mikäli ohi menevä paketti täyttää asetetut ehdot, sen sisältämä sisältä suoritetaan. Kyseisenlaisen aktiviteetin torjumiselle ainoa vaihtoehto olisi vaihtaa koko verkkokortti, mikä on perin haastavaa etenkin emolevyyn integroitujen verkkokorttien osalta. Tällainen haavoittuvuus voi olla omiaan erityisesti etätyöpöytätyöskentelyssä, jos kyetään tarkkailemaan mahdollisen etäyhteyden ulkopuolella tapahtuvaa liikennettä. Lisäksi mikäli ympäristössä on toteutettu kahden virtuaalikoneen Fault Tolerance -tyyppinen vikasietoinen järjestely, on huomattava, ettei liikenne virtuaalikoneiden välillä ole millään lailla salattua [26]. Virtuaalikoneiden välinen liikenne kuitenkin sisältää kaiken liikenteen sekä paikallisten laitteiden että verkkoliikenteen kannalta vikasietoisuuden toteutustavan vuoksi.

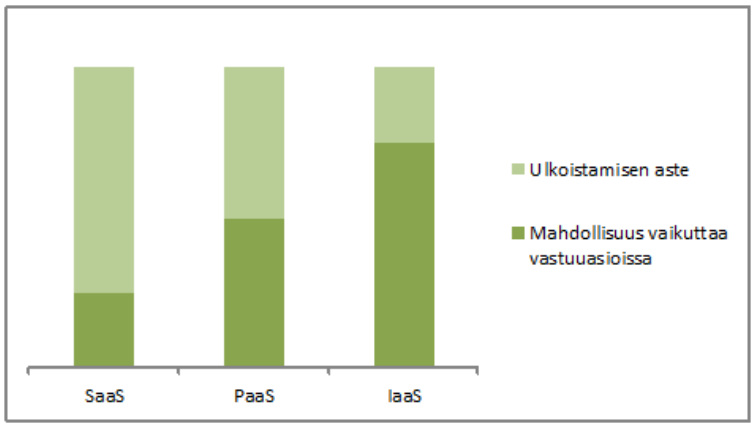


Kuva 10. Haavoittuneen verkkokortin kautta päästään käsiksi näytönohjaimessa CUDA-rajapinnalla ajettavaan SSH-palvelimeen [25].

Ostettaessa tai myydessä etäresurssipalveluja ovat asteesta riippumattomat etäresurssipalvelun vastuukysymykset ja riittävän tietotaidon määrän määrittely hankalia. Voidaan esittää esimerkiksi kyseenalaiseksi etäresurssipalveluntarjoajan vastuu vihamielisen toiminnan tapahduttua palvelinsalissa tai samassa palvelimessa toimiessa, jossa on ostajan ulkopuolisen tahon toimintaa. Esimerkiksi etäresurssipalveluntarjoajan Maux II -tyyppisen käsittelyn kohteeksi joutunut palvelimen verkkokortti tai onnistunut Guest-to-Host -tyyppinen hyökkäys on omiaan vaarantamaan ostajan tietoturvan ja yksityisyyden asteen. Myös voidaan asettaa kyseenalaiseksi laajempia virtualisointitarjoomia ostettaessa, mikä on ostavan organisaation tietotaidon oltava, jotta kuitenkin kyetään käyttämään ostettua etäresurssikapasiteettia rittävissä määrin. Mielestäni tätä on pyritty havainnollistamaan kuvassa 11, josta nähdään, että ulkoistuksen asteen suhde on jotakuinkin kääntäen verrannollinen mahdollisuuksiin vaikuttaa vastuukysymyksissä, sillä tällöin asiakkaan hankkima palvelutaso on lähempänä laitteistoa ja asiakas itse hoitaa osansta toimivuudesta.

Käytössä olevien resurssien yliprovisiointi

Aivan kuin ei-virtualisoiduissa ympäristöissä, voi myös virtualisoidussa ympäristössä yliprovisioida resursseja. Esimerkiksi tallennuskapasiteetin määärällisen ja suorituskyvyl-



Kuva 11. Ulkoistamisen asteen kasvaessa ostavan organisaation mahdollisuudet vaikuttaa suoraan toimintaansa kasvavat.

lisen tai yliprovisioimalla keskusmuistia saattaa olla kauaskantoisia vaikutuksia. Niiden vääränlaisella jakamisella saatetaan tukkia kaikki kyseistä resurssia aktiivisesti käyttävät virtuaalikoneet. Yliprovisioitaessa levytilaa levyjärjestelmäprofilointi yhdistettynä Storage DRS -ominaisuuteen auttavat automatisoimaan asennussijaintien kuormituksen tasaamista sekä mahdollistavat käyttäjille annettavaksi oikeanlaista kapasitettia kyseiseen tarkoitukseen nähden. Resurssipoolien avulla voidaan hallita virtuaalikoneiden keskinäistä prosessorin käyttöastetta ja tosiasiallista keskusmuistikapasiteetin saavutettavuutta niiden jäädessä niukaksi. Levyjärjestelmän tallennusprofiilin avulla on mahdollista rajoittaa tietyn tyyppisten virtuaalikoneiden ja tiettyjen käyttäjien levytilankäyttöä esimerkiksi suorituskykyä ja sitä kautta kustannustasoa silmällä pitäen.

Palveluiden saatavuutta maksimoivat ominaisuudet

Korkean saatavuuden osalta riskit syntyvät siinä, jos virtuaalikoneiden suorittamat palvelut joutuvat erilleen toisistaan vielä käynnissä ollessaan. Näin edes osittain toimiessaan ja parametroidusti vCenter tulkitsee virtuaalikoneen joko sammuneeksi, keskeytyneeksi tai erityneeksi ja käynnistää hetken ajan päästä uuden virtuaalikoneen. Tästä aiheutuu palvelun keskeytyminen helposti minuuteiksi. Lisäksi mikäli vanhaa virtuaalikonetta ei saada ajettua alas, uuden virtuaalikoneen käynnistäminen vielä vanhan ollessa päällä saattaa

ryppään toimimaan kaksijakoisesti, jos lukitustiedostojen kanssa on ongelmia.

3 Etäresurssi

Etäresurssi koostuu laitteistoresursseista sekä niitä hyödyntävästä ohjelmistokerroksesta. Täten sen voidaan nähdä sisältävän sekä fyysisiä että abstrakteja ominaisuuksia. Fyysiset ominaisuudet sisältävät muun muassa laitteiston, verkon ja tallennuskapasiteetin hallinnan. Abstrakti taso taas sisältää ohjelmiston käsitteen, mikä mahdollistaa etäresurssin toimimisen laitteistossa. Etäresurssipalvelu on määritelty NIST:n (National Institute of Standards and Technology) mukaan palveluksi, joka mahdollistaa kaikkialla läsnäolevan, mukavan, tarvittaessa verkkoyhteyden mahdollistavan tavan hallita resursseja, joita voidaan varata ja vapauttaa joustavasti mahdollisimman vähäisin ylläpitotoimin. Tämä etäresurssipalvelu koostuu viidestä perusominaisuudesta, kolmesta palvelumallista sekä neljästä erilaisesta käyttöönottolaajuudesta.

3.1 Etäresurssin viisi ydinominaisuutta

NIST:n standardi [27] määrittelee etäresurssin sisältävän ydinominaisuuksinaan kertaitsepalvelun, laaja-alaisen verkkoyhteyden, resurssien yhteenliittämisen, nopean elastisuuden ja palvelun mitattavuuden.

Kertaitsepalvelun (On-demand self-service) avulla asiakkaan on mahdollistettava yksipuolisesti säädellä etäresurssin eri resurssien määrää ja laatua. Esimerkiksi palvelimen kelloaikaa ja tietoverkon kautta käytettävää tallennuskapasiteetin muuttamista on oltava mahdollista muokata ilman ihmiskontaktia.

Laaja-alaisen verkkoyhteyden (Broad network access) kautta on kyettävä perinteisin menetelmin verkkoyhteyden avulla hallitsemaan etäresurssin ominaisuuksia heterogeenisen ympäristön avulla. Tähän ryhmään voidaan nähdä kuuluvan esimerkiksi mobiililaitteet, kannettavat tietokoneet ja työasemat ja tabletit.

Resurssien yhteenliittäminen (Resource pooling) sisältää palveluntarjoajan mahdollisuuden palvella samoin resurssein useita asiakkaita palvelurakenteella, mikä mahdollistaa samanaikaisen virtuaalisten ja fyysisten resurssien asiakaskohtaisen osoitustilan muutoksen tarpeen ilmetessä. Järjestelmä on sijaintiriippumaton toiminnaltaan siten, ettei asiakkaalla yleisesti ole sen erityisempää kontrollia sijainnistaan kuin maan tai palvelinkeskuksen tarkkuudella. Resursseilla tarkoitetaan esimerkiksi tallennuskapasiteettia, prosessorikapasiteettia, keskusmuistia tai verkkoyhteyttä.

Nopean elastisuuden (Rapid elasticy) sisällyttää etäresurssille edellytyksen käsitellä etäresurssin eri resursseja siten, että niitä voidaan osoittaa, osittaa ja vapauttaa joustavasti, ja joskus jopa automaattisesti. Tämä mahdollistaa palvelun säätämisen kysynnän mukaisesti joko sisään tai ulospäin yhteismitallisesti. Kuluttajalle palvelun säätömahdollisuudet näyttäytyvät rajoittamattomina ja määrittelemättömissä määrin kaikkina hetkinä ilman aikasidonnaisuutta.

Palvelun mitattavuus (Measured service) edellyttää etäresurssipalvelun mittaavan, ohjaavan ja siten optimoivan toimintaansa sopivan tason mittareilla. Resurssien käytettävyyttä voidaan tarkkailla, ohjata ja raportoida mahdollistaen läpinäkyvyyden sekä asiakkaalle että palveluntarjoajalle kyseisen rakenteen osalta.

3.2 Etäresurssin palvelulaajuudet

Etäresurssin palvelut voidaan jakaa periaatteessa kolmeen kategoriaan: ohjelmisto palveluna, sovellusalusta palveluna sekä infrastruktuuri palveluna. Nimiensä mukaisesti palvelukokonaisuudet tarjoavat ohjelmistoa, sovellusalustaa ja infrastruktuuria.

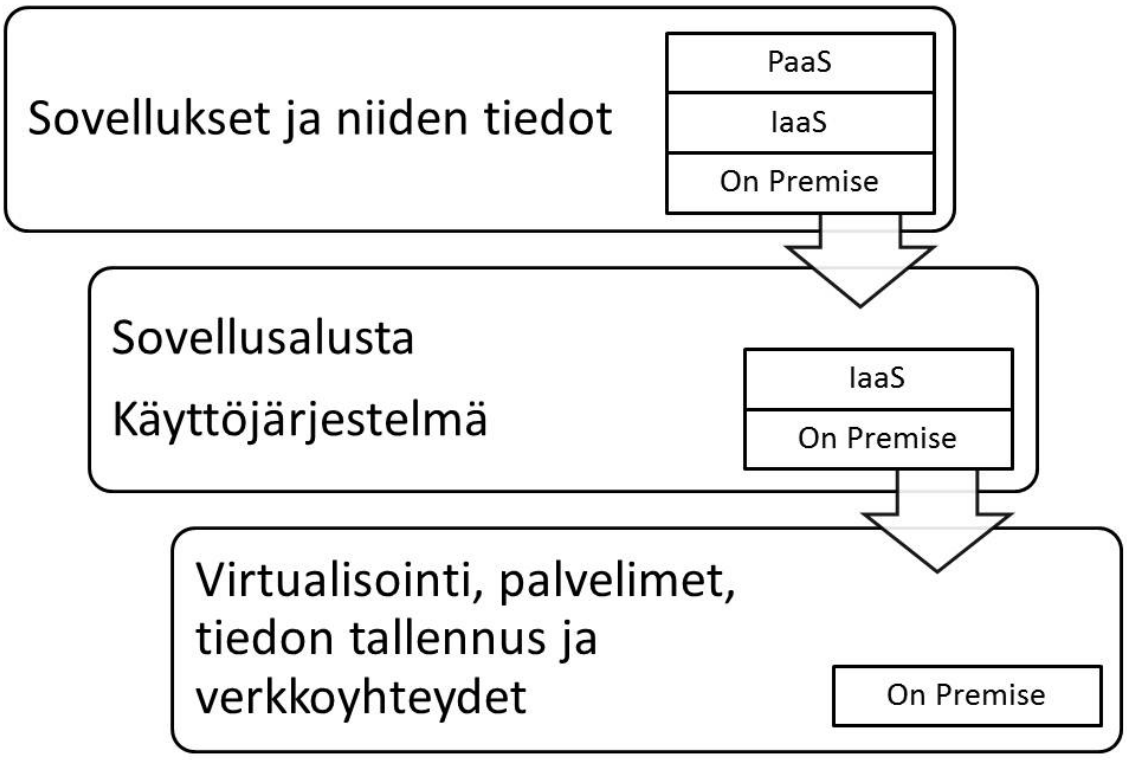
Ohjelmisto palveluna (Software as a Service, SaaS) sallii asiakkaan käyttää palveluntarjoajan laitteistolla etäresurssissa ajettavaa ohjelmistoa verkkoyhteyden yli. Ohjelmistojen on oltava käytettävissä edellä mainittujen puitteiden mukaisesti joko kevyen selaimella käytettävän käyttöliittymän kautta (esimerkiksi sähköpostipalvelut) tai ohjelmiston avulla. Asiakas ei hallitse tässä nimenomaisessa mallissa allaolevaa etäresurssiarkkitehtuuria

sisältäen verkkoinfrastruktuurin, palvelimet, käyttöjärjestelmät, tiedon tallennuksen ja yksittäisten ohjelmistojen ominaisuudet poislukien kuitenkin asiakaskohtaiset kustomoinnin mahdollisuudet yksittäisten ohjelmistojen kohdalla. SaaS-tyyppisiä palveluita ovat esimerkiksi erilaiset sähköpostipalvelut, Cisco WeBex ja Office 365.

Sovellusalusta palveluna (Platform as a Service, PaaS) mahdollistaa asiakkaalle oman asiakaskeskeisen ohjelmiston tuomisen etäresurssi-infrastruktuuriin tai mahdollistaa sellaisen luominen palveluntarjoajan tarjoamien ohjelmointikielten, kirjastojen, palvelujen ja työkalujen avulla. Asiakas ei hallitse tai kontrolloi allaolevaa etäresurssiratkaisua mukaanlukien verkkoyhteydet, palvelimet, käyttöjärjestelmät ja tallennusratkaisut kuitenkin omaten mahdolllisuuden hallita tuomiaan tai luomiaan ohjelmistoratkaisuja ohjelmistoratkaisun kontekstissa. Esimerkillisiä PaaS-tyyppisiä palveluita ovat muunmuassa Azure, Elastic Beanstalk ja Google App Engine.

Infrastruktuuri palveluna (Infrastructure as a Service, IaaS) luo asiakkaalle laajat puitteet toimia etäresurssiympäristössä. Prosessointikapasiteetin, tiedon tallennuksen osiointi sekä yksityiskohtaiset konfigurointimahdollisuudet muunmuassa verkkoyhteyksien ja muiden ydinosa-alueiden kohdalla on mahdollista. Asiakas kuitenkaan ei hallitse tai ohjaa itse etäresurssikäytössä olevaa laitteistoa, mutta omaa valtuudet kontrolloida käyttöjärjestelmiä, tallennusratkaisuja ja käyttöönotettuja ohjelmistoja sekä mahdollisesti kykenee hallitsemaan rajoitetusti verkkolaitteiston komponentteja --- esimerkiksi palomuuria. Esimerkkejä IaaS-tyyppisestä etäresurssipalvelusta ovat Amazon EC2, UpCloud ja Rackspace.

Eräs aspekti, jota voidaan palvelulaajuuksien suhteen arvioida, on On Premise (paikallinen palvelu) eli koko infrastruktuurin hoitaminen omana tuotantona. Tämä varsinaisesti ei ole etäresurssirakenteen kannalta olennainen, mutta se auttaa hahmottamaan kuvan 12 tapaan sen laajuutta suhteessa aiempiin mainittuihin laajuudellaan.



Kuva 12. Kuvassa etäresurssipalvelulaajuudet PaaS ja IaaS suhteutettuna paikalliseen sovellukseen tosiasiallisen hallinnan ja vastuiden suhteen. Mukaillen [28].

3.3 Käyttöönottolaajuudet

Etäresurssisuoritteisia palveluita on mahdollista järjestää NIST:n standardin mukaisesti neljässä mallissa. Suurin eroavaisuus malleilla on niiden omistajuudessa ja suhteellisessa sijainnissa käyttäjiin nähden. Etäresurssipalvelut tuotetaan joko paikallisesti (On Premise) tai ulkoa (Off Premise).

Yksityisessä mallissa (Private Cloud) etäresurssiarkkitehtuuri on ainoastaan yhden organisaation käyttöön käsittäen näiden asiakkaat, kuten esimerkiksi organisaation sisäiset yksiköt. Yksityinen pilvi ei ota kantaa infrastruktuurin ja ohjelmistotason omistajuussuhteisiin, ja siten ne voivat olla joko itse organisaation tai kolmannen osapuolen hallitsemia, omistamia sekä operoitavissa. Palvelut voidaan tuottaa joko talon sisältä tai ulkoa.

Yhteisöllisessä mallissa (Community Cloud) etäresurssiarkkitehtuuri on osioitu yksilölliselle yhteisölle tietyn organisaation asiakkaille, joilla on yhteisiä intressejä etäresurssin käytölle. Näitä voivat olla esimerkiksi yhteisön tehtävä, tietoturvasyyt, tietoturvapolitiikka. Nimenomainen omistajuus, hallinnallisuus tai hallittavuus ei nimenomaisesti ole yksilöitynyt yhteen yhteisöön kuuluvaan tahoon vaan voi olla jakautunut useammalle taholle. Malli ei ota kantaa palvelujen tuotannon suhteelliseen sijaintiin käyttäjään nähden.

Julkinen malli (Public Cloud) on nimensä mukaisesti osoitettu julkiseen käyttöön eikä sen voida katsoa olla osoitettu yksittäiselle tietylle organisaatiolle. Se voi olla omistettu, hallittu ja operoitu joko yksityisen yrityksen, akateemisen tahon, valtiollisen organisaation tai näiden yhdistelmän toimesta. Suhteellinen sijainti käyttäjään nähden voi olla mikä tahansa eikä malli siihen ota kantaa.

Risteymämalli (Hybrid Cloud) on edellämainittujen mallien yhdistelmä. Siinä on vähintään kaksi aiemmin mainittua mallia yhdistetty teknisin keinoin siten, että ovat yksilöllisiä palvelurakenteita sallien tiedon ja sovellusten siirreltävyyden. Tämä on erinomaisen tärkeää etäresurssipalveluiden yhteydessä ilmenevien äkillisten kuormituspiikkien aikana. . Risteymämalli voisi esimerkiksi olla kahden etäresurssin palveluntarjoajan yhteenliittymä, jossa ne pyrkivät yhteisellä etäresurssikapasiteetilla toimimaan sekä yksityisessä että yhteisessä etäresurssipalvelukentässä.

3.4 Kustannusten läpinäkyvyys

Yleisesti IT-palveluiden kanssa voidaan nähdä haasteellisena palveluiden tuottamiseen kuluneiden kustannusten maksajan tunnistaminen helposti ja riittävällä tarkkuudella. vCloud-ympäristöön on mahdollista konfiguroida erillinen Chargeback-sovellus, joka pitää kirjaa kulutetusta laskutehosta, keskusmuisti- ja kiintolevykapasiteetista ja yhdistää ne muodostetun hinnoittelutaulukon kanssa luoden selkeämmän tilaston kuormituksesta. Laskutuksen selkeyttä voidaan parantaa koostamalla kustannnukset kiinteistä hinnoista, kuten räkkikaapin yksikkökustannuksista, sähkö- ja tietoliikennekaapelointien muodostamisesta sekä allokaatiopohjaisesti etäresurssina varattujen virtuaalikapasiteetin hinnasta ja

kuormituksen perusteella tapahtuneen kapasiteetin varauksen ja käytön perusteella. Tämän avulla on mahdollista saada reaalinen ja vanha tietoa täydentävä näkökulma IT-palveluiden toteuttamisen ja ylläpidon kokonaiskustannuksia ajatellen.

3.5 Suorituskyvyn allokointi ja resurssointi

Etäresurssiarkkitehtuurin sisäiset laskutusrakenteet samanaikaisesti sekä auttavat asiakasta hahmottamaan ympäristön suosituksesta syntyvien kustannusten rakennetta että mahdollistavat palvelutasosopimuksien kanssa resurssien tehokkaamman provisioinnin ja käytön. Palvelutasosopimuksilla on mahdollista kategorisoida allokaatiomallien pohjalta ryhmiin jaettujen resurssien keskinäinen suorituskykytasoero ja palvelutaso sovitun mittariston mukaisesti. Mittareina voidaan esimerkiksi pitää vastetta kasvaneeseen suorituskykytarpeeseen, uusien virtuaali-instanssien käynnistysnopeutta tai palvelun yleistä palveluntarjoajasta riippuvaa saavutettavuutta. Erilaisilla rakenteilla ja hinnoittelurakenteilla fyysisen palveluntarjoajan on mahdollista ylläpitää korkeaa käyttöastetta haluamissaan laitteistoissa sekä ohjata kysyntää vähemmän käytetyille vapaammille resursseille tarpeen tullen.

Chargeback on vCenter

vCenter Chargeback on etäresurssipalvelumalleille luotu ohjelmisto erilaisten laskutusmallien luomiseksi ja laskutukseen tarvittavan datan keräämisen automatisoimiseksi. Ohjelmisto mahdollistaa helpon tavan tarjota palveluita laskutusjärjestelmän sitä tukien. Kun Chargeback otetaan käyttöön, luodaan sille paikallinen oma tietokanta, johon se koostaa vCenter-palvelimelta käyttö- ja kuormitusdatan. Tämän jälkeen kerätystä datasta koostetaan aiemmin määriteltyjen säännöksien mukaisesti asiakaskohtaisia tietueita. Tietueiden perusteella asiakaskohtainen laskutus mahdollisestaan tarkasti ja helposti. Chargebackin käyttämät säännökset luodaan joko vCenteristä tai vCloud Directorilta.

Käyttöperusteinen malli

Juoksevassa laskutuksessa etäresurssin tasolle ei luoda omaa erityistä resurssiallokaatiorakennetta vaan etäresurssissa ajettavat virtuaalikoneet toimivat virtuaalipalveluntarjoajan resurssien asetusten mukaisesti. Mikäli esimerkiksi palveluntarjoajalla on resurssien takausasteena 50 %, on se sama myös tämän palveluluokan virtuaalikoneilla oletusarvoisesti. Tässä mallissa on myös mahdollista asettaa taattu taso, mihin asti resurssit varmasti ovat käytössä ja mahdolliset resurssirajoitukset, mikäli niitä halutaan toteuttaa, toteutuvat virtuaalikonetasolla. Ominaista tälle rakenteelle [29] on kustannusten syntyminen vasta instanssien ollessa toiminnassa. Taulukosta 2 voidaan nähdä kustomoinnin tapahtuvan virtuaalikoneen tasolla.

Varausperusteinen malli

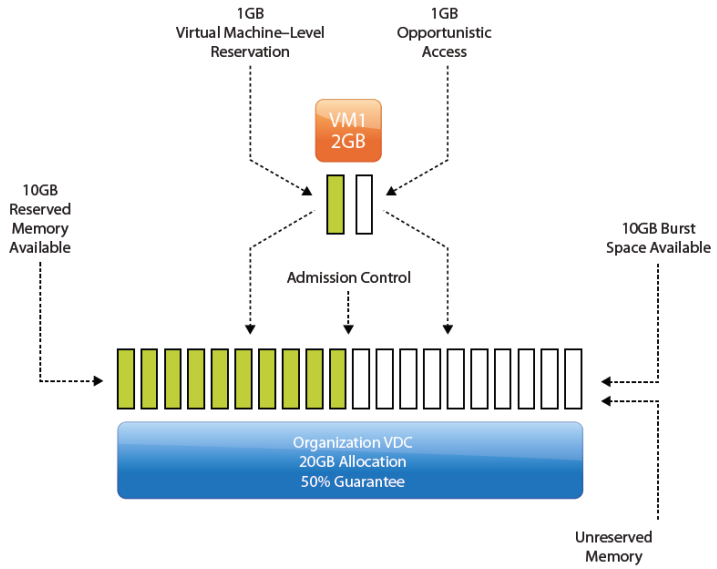
Siirryttäessä pois virtuaali-instanssipohjaisista varausrakenteista resurssien laajempaan varauskäsittelyyn päädytään varausallokaatioperusteiseen rakenteeseen. Funktionaalisesti käytön perusteinen malli vastaa täysin resurssien takaustasolla varausperusteista mallia. Nimensä mukaisesti tässä mallissa [29] luodaan resurssivaranto ja siinä suoritettavat instanssit ovat ilman yksilöityjä rajoituksia. Periaatteessa käyttöperusteisen mallin virtuaalikonekohtaiset rajoitukset siis yhdistetään ja siirretään resurssivarantotasolle. Kuten taulukosta 2 nähdään, rajoitukset ovat resurssivarannon tasolla.

Allokaatioperusteinen malli

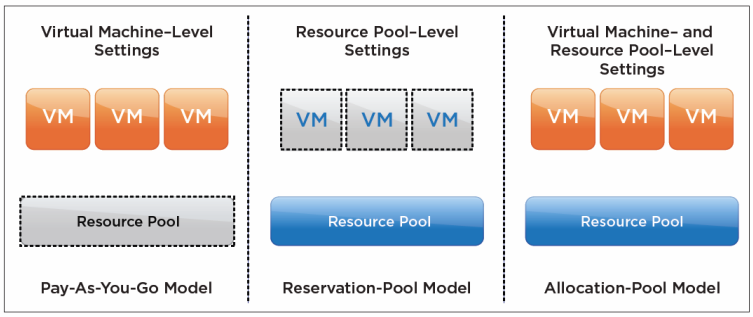
Yhdistettäessä varausperusteisuus ja käyttöperusteisuus päästään allokaatiopohjaiseen ratkaisuun, jossa instanssille varataan palveluntarjoajalta ensin takausrajan verran resursseja sen omaan resussivarantoon, minkä jälkeen virtuaalikonekohtaiset rajoitukset voidaan ottaa käytöön. Oletusarvoisesti [29] virtuaalikonekohtainen takausraja prosessoriteholle ja muistikapasiteetille ovat asianomaisissa maksimiarvoissaan.

Taulukko 2. Resurssien allokaation sijoittuminen virtuaalikone- ja resurssivarantotasolle.

	Käyttöperusteinen	Varausperusteinen	Allokaatioperusteinen
Virtuaalikonetaso	X	-	X
Resurssivarantotaso	-	X	X



Kuva 13. Allokaatiomallissa luvattu osa resursseista allokoidaan virtuaaliympäristön käyttöön suhteutettuna luvattuun tasoon [29].



Kuva 14. Kolmihaarainen resurssien allokointipohja etäresurssina [29].

4 Käytettyjen ohjelmistojen taustaa ja teoriaa

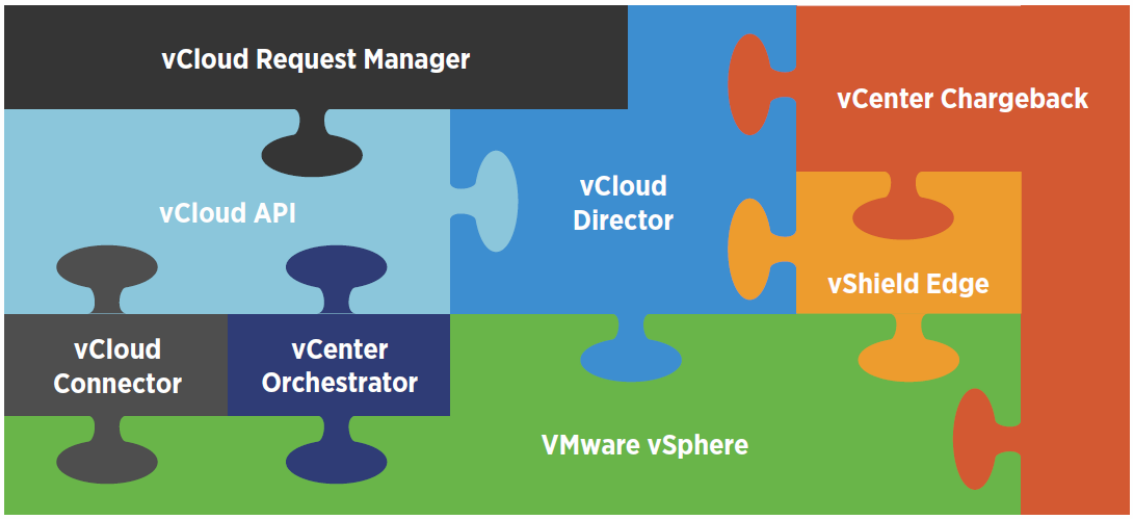
Insinöörityön perusinfrastruktuuri perustuu VMware vSphere 5.1 -ympäristön tuomaan hajautettuun kytkimeen, vShield Managerin ja Edgen tuomiin verkko-ominaisuuksiin sekä vCloud Directorin nämä yhteen nivomaan teknologiaan. Yhdessä nämä voidaan käsittää sisältyvän vCloud Suite -tuotepaketin ydinkokonaisuudeksi. Lisäksi työ perustuu juuriltaan myös verkkoteknologisesti iSCSI:n toimintaan. Seuraavassa esitellään eräitä työssä käytettyjen tuotteiden ja teknologioiden ominaisuuksia sekä niiden tuomia mahdollisuuksia.

4.1 vSphere

VMware vSphere 5.1 koostuu ESXi-hypervisorista sekä vCenter (vCSA, vCenter Server Appliance) -hallintaohjelmistosta sekä siihen liitettävistä muista toimintaa tukevista palveluista. Insinöörityössä vSphere-ympäristö on rakennettu sisäkkäisesti virtualisoidusti alla olevan vSphere-ympäristön sisälle.

4.2 Virtuaalikytkimet

vSpheren ilmaisversioon kuuluva vSwitch (vNetwork Standard Switch, vSS) on VMwaren perustavanlaatuisin virtuaalinen kytkin, joka toimii paikallisesti hypervisorin tasolla. Sitä on luonnollista käyttää esimerkiksi Virtual LAN -käytön yhteydessä sen tukiessa 802.1q-tyyppistä VLAN-erottelua ja yksinkertaisia tietoturvaominaisuuksia kuten MAC-osoitteen suodatusta sekä MAC-osoitteeseen perustuvaa lähteen ja kohteen suodatusta. vSwitch tukee myös liikenteen kaistarajoitusta, kuorman tasausta usean verkkokortin välillä verkkokortteja ryhmittelemällä sekä vikasietoisuutta asettamalla verkkokortteja joko rinnakkain tai varalla käytettäviksi.



Kuva 15. Kuvakooste vCloud Suiten sovelluskokonaisuudesta [30].

Hajautettu virtuaalinen kytkin eli dvSwitch (vDS, vNetwork Distributed Switch) on käytännössä usean eri palvelimen välille muodostettava virtuaalinen kytkin, joka mahdollistaa notkeamman ja tehokkaamman infrastruktuurisuunnittelun. Hajautettu kytkin tukee esimerkiksi alkeellisia kuormantasaussääntöjä, liikenteen priorisointia sekä virtuaaliverkkojen osoitusta valituille porteille. dvSwitchin käyttö edellyttää vCenterin asentamista ja sen Premium-tason lisensointia. Nexus 1000v on VMwaren ja Ciscon valmistama ohjelmistopohjainen hajautettu kytkin, jolla kyetään hajautetun dvSwitch-kytkimen suhteen monimutkaisempiin konfiguraatioihin.

4.3 vCloud Suite

vCloud Suite koostuu laaja-alaisemmasta mahdollisuudesta käyttää erinäisiä sovelluksia lokitietojen keräämiseen, niiden analysointiin sekä toimien automatisointiin. vCloud Suitella on myös mahdollista yhdistää yksityinen ja julkinen etäresurssilaitteisto yhtenäiseksi saumattomaksi laitteistokannaksi ja mahdollistaa työkulkuperusteisen hajautuksen etäresurssikuormalle ja sen muokkaamiselle. Insinöörityössä keskityttiin vCloud Suiten ytimen toimintaan, joka käsittää vSphere-ympäristön, vCloud Directorin sekä vShield Managerin.

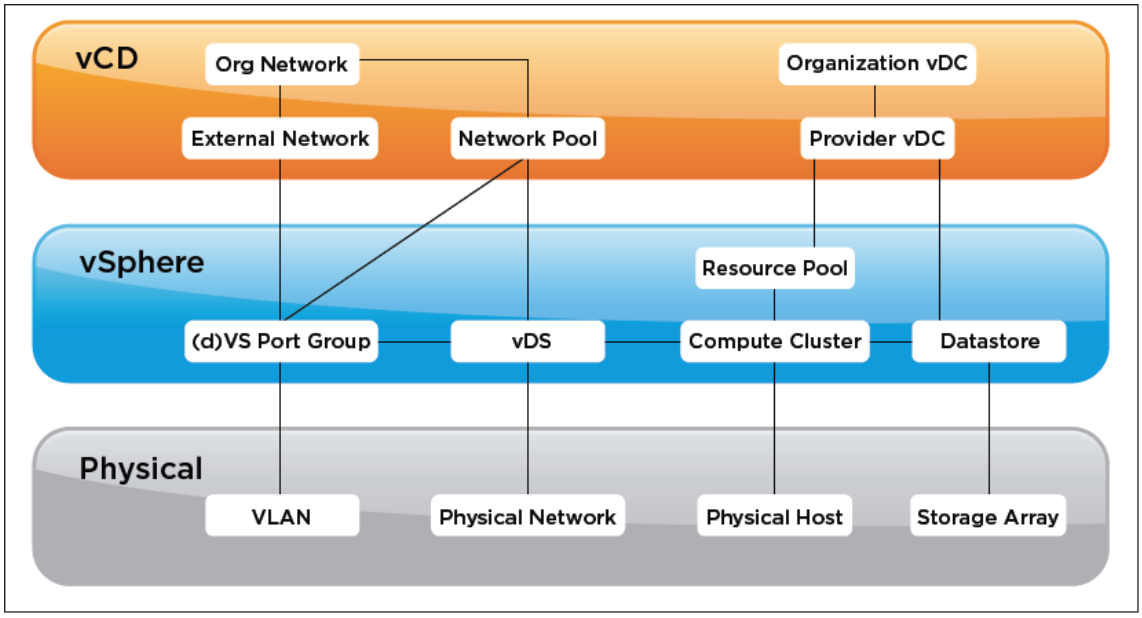
4.3.1 vCloud Director

vCloud Director (vCD) nivoo yhteen vSpheren hypervisorin sekä vCenterin vShield Managerin kanssa. vCloud Directorin yhteydessä otetaan käyttöön jokin aiemmin sivulla 31 käsitellyistä malleista, jonka pohjalle luodaan virtuaaliorganisaatio. Resurssien allokoinnin lisäksi sovelluksella hallitaan etäresurssin käyttäjien ulottuvilla olevia valmiita virtuaalikonekokonaisuuksia sekä verkkorakenteita. vCloud Director on etäresurssin organisaattori samassa suhteessa kuin vCenter on virtuaalikoneiden palvelimille datacenterin organisaattori. Muokattaessa etäresurssin kokoonpanoa vCloud Director ohjaa komennot vShield Edgelle ja vCenterille.

Virtuaaliorganisaatiolle, jonka käyttöön ylipäätänsä etäresurssiresursseja allokoidaan, luodaan myös erityinen virtuaalinen katalogi. Katalogin avulla asianomainen järjestelmänvalvoja kykenee tuomaan haluamansa laisia virtuaalikone- ja verkkokonfiguraatioilla varustettuja kokonaisuuksia valmiina käyttöön organisaatiossaan. Tällöin esimerkiksi valmiin testiympäristön palvelimet voidaan esikonfiguroida siten, että testiympäristön käyttäjä tilaa kokonaisuuden katalogista ja vCloud Director hoitaa asianomaiset konfiguroinnit omalta osaltaan siten, että ympäristö on käytössä huomattavan nopeasti ilman ylimääräisiä välikäsiä. Virtualisointiympäristö mahdollistaa myös virtuaalikoneiden luomisen joko esimerkiksi aihion tai käyttäjäoikeuksien kautta tiettyyn rajoitettuun palvelutasoon esimerkiksi tietosäilöprofiilien avulla.

4.3.2 vShield Manager

VMware vCloud Suite sisältää vCloud Network Security -moduulin (vShield Manager, vShield Edge, vCNS) etäresurssista, jolla on tarkoitus järjestää etäresurssinssa ajettavien virtuaalikoneiden verkkoyhteydet. Verkkoyhteyksiä voidaan muodostaa paikallisesta solusta internetiin sekä muihin soluihin ja soluihin, joilla suoraa yhteyttä ulkoiseen verkkoon ei varsinaisesti välttämättä ole. vCloud Suite 5.1 -versiossa on mahdollista yhdellä Edge-instanssilla olla jopa kymmenen verkkokorttia sivun 40 kuvan 17 tapaan asettaen tietynlaisen käytännöllisen rajoituksen verkkojen monimutkaisuudelle ja ristiinkytkemisel-



Kuva 16. vCloud etäresurssiympäristön kerrosdiagrammi [31].

le. vShield Edge on myös vastuussa infrastruktuurin niin nimi- ja IP-osoitteiden DHCP-jakelupalvelimena (Dynamic Host Configuration Protocol) toimimisesta sekä palomuuritoiminnoista kuin NAT-osoitteenmuunnoksista sekä porttikohtaisella (PAT) että osoitekohtaisella (dynaaminen ja staattinen NAT) tasolla. Porttikohtaisessa osoitteenmuunnoksessa ohjataan ulkoisesta verkosta tavoiteltu portti yhteen tiettyyn sisäverkon osoitteeseen. Dynaamisessa ja staattisessa osoitteenmuunnoksessa osoitetaan jokaiselle sisäverkon IP-osoitteelle ulkoinen IP-osoite joko ennalta määritellyssä järjestyksessä taikka eritellysti määritellen osoitekohtaisesti.

VMware vShield Endpoint on saatavissa hypervisorin tasolla toimivaksi virustentorjuntaohjelmistoksi. Hypervisorin tasolla toimivan valmistajan toimesta esikustomoidun virtuaalikoneen avulla suoritettava virustorjuntaohjelmiston ajaminen mahdollistaa joustavamman infrastruktuurisuunnittelun ja vähentää satunnaisia kuormituspiikkejä. Yksittäisille virtuaalikoneille asennetuissa virustorjuntaohjelmistoissa ongelmana on paikallisen virtusohjelmiston aiheuttama poikkeuksellinen ja hetkellinen kuormitus, mikä heikentää ryppään tasapainoasetelmaa. Tämä yhdistettynä esimerkiksi DRS-ominaisuuden kanssa mahdollisesti aiheutaa tarpeettomia virtuaalikoneen siirtoja palvelimelta toiselle ja yleistä suorituskyvyn laskua.

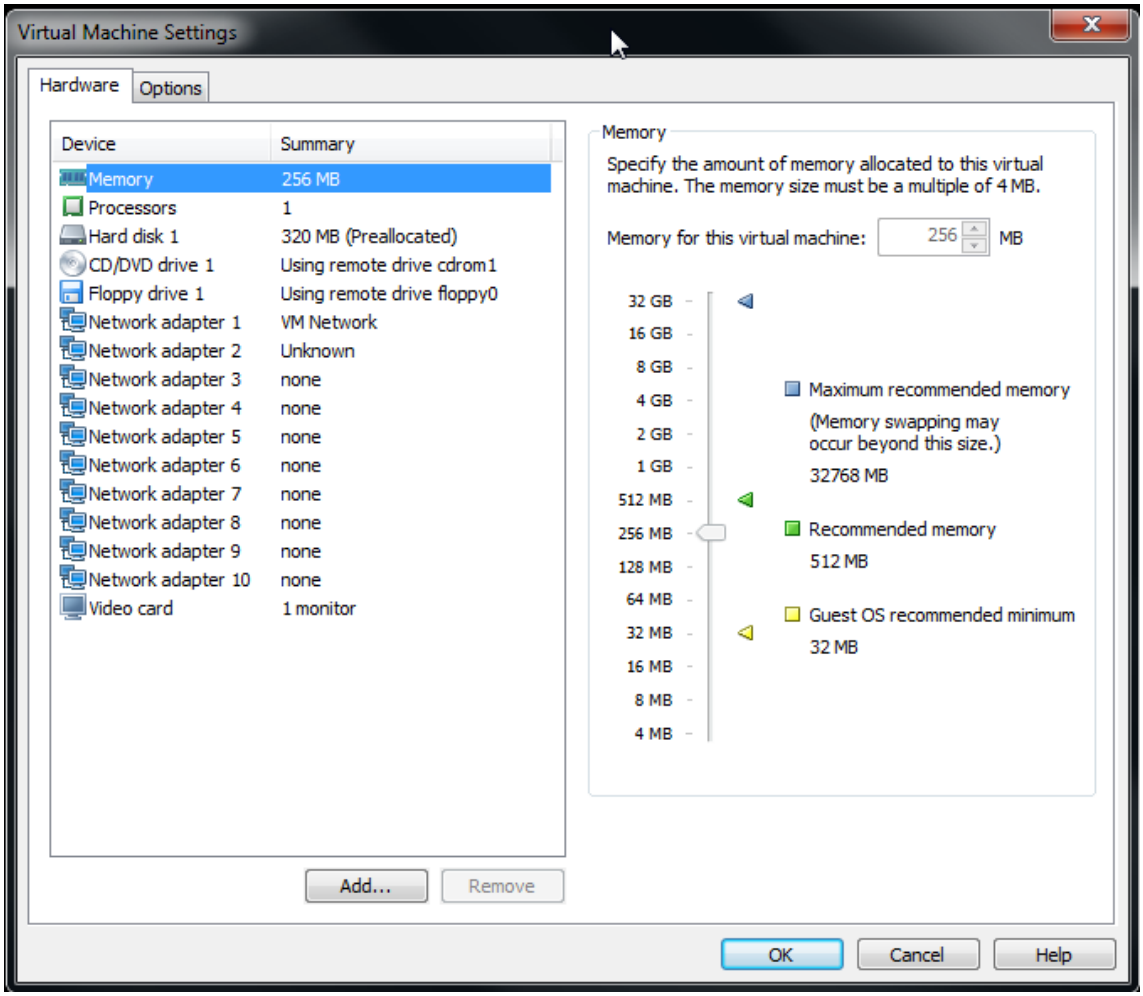
4.3.3 vShield Edge

vShield Managerin kautta asetettava vShield Edge (Edge) kykenee pohjimmiltaan kolmen eri tyypin verkkokonfiguraatioihin, joita ovat suora sillattu yhteys, reititetty yhteys sekä eristetty paikallinen verkko. Monimutkaisemmat verkkorakenteet pohjimmiltaan perustuvat kuitenkin juuri näihin perusrakenteisiin. Tarvittaessa virtuaalikokoelma (vApp) voidaan haluttaessa anonymisoida (Fence) mahdollistaen samojen IP-osoitteistusten jakamisen rinnakkaisille instansseille organisaatiossa. Ulkoisesti arvioiden verkkokonfiguraatiot ovat joko ulkoisen verkon, organisaation tai virtuaalikokoelman tasolla. Osoitteita ja osoiteavaruuksia, joita ei ole määritelty Edgelle vCloud Directorin kautta joko muodostamalla DHCP-palvelin tai staattinen osoiteavaruus, ei lähtökohtaisesti voi etäresurssiympäristön sisäisessä verkossa käyttää etäresurssiympäristön rakenteesta johtuen poislukien suoran siltauksen mahdollistama yhteys. Toiminnalliseti Edge yhdistyy porttiryhmään, joka luodaan asianomaista vApp-kokonaisuutta varten ja ne näkyvät kuvassa 17 ensimmäisen verkkokortin alla niiden ollessa kytkettyjä.

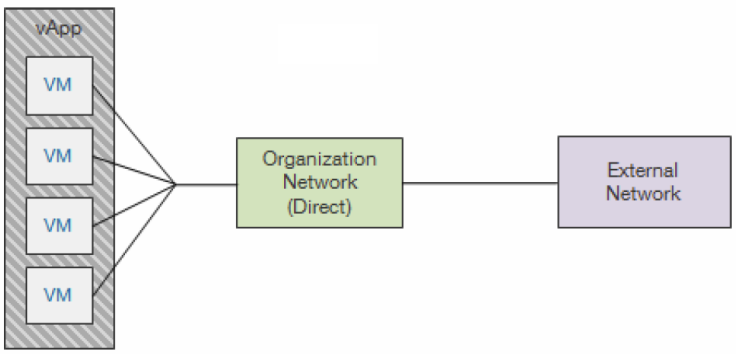
Seuraavissa kuvissa sivulta 40 alkaen esitetään kolme perustavanlaatuista verkkorakennetta, joita IaaS-tason etäresurssipalveluissa voidaan toteuttaa ja joita voidaan kohdata. Nämä ovat pohjana vCloud-toteutusympäristön kanssa toimittaessa:

- suora sillattu yhteys kuvan 18 mukaisesti
- Edgen kautta reititetty yhteys kuvan 19 mukaan
- eristetty rakenne kuvassa 20 Edgen toimiessa paikallisena kytkimenä.

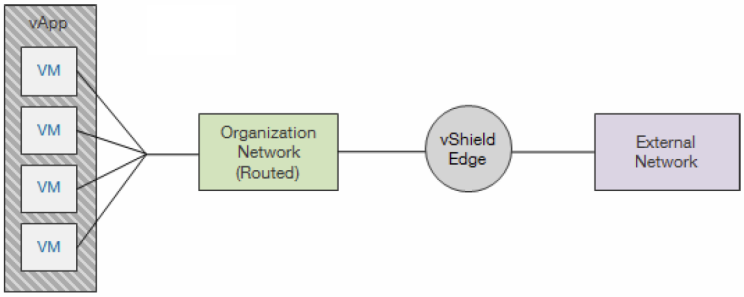
Yksinkertaisin perusrakenne kuvan 18 kaltaisesti ei sisällä lainkaan vShield Managerin hallinnoimaa Edgeä. Siten se myöskään ei voi sisältää aktiivisesti toimivaa DHCP-palvelinta taikka palomuuria. Virtuaalikoneille kyetään asettamaan staattiset kiinteät IP-osoitteet ja verkkoasetukset tai ne voidaan hakea ulkoverkosta perinteisin keinoin suoraan virtuaalikoneilla. Organisaation verkko siis käyttää ulkoisen verkon osoiteavaruutta.



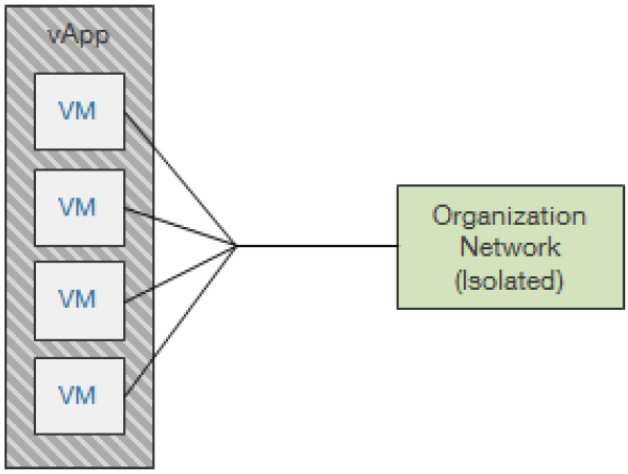
Kuva 17. vShield Edgen oletusarvoinen verkkokonfiguraatio ilman vApp-liitännäisyyksiä.



Kuva 18. Edge ei osallistu verkkoliikenteen reititykseen tai suodattamiseen [32].



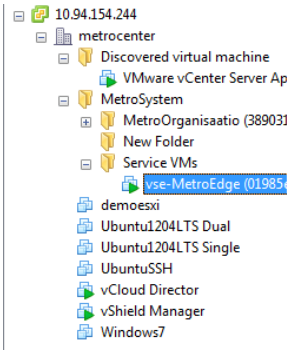
Kuva 19. Liikenne ulos kulkee Edgen kautta [32].



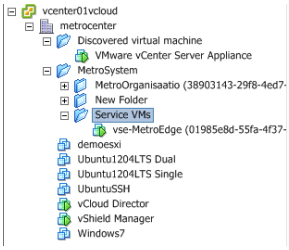
Kuva 20. Eristetystä verkosta ei ole reittiä ulkomaailmaan [32].

Reititetyssä mallissa kuvan 19 tapaisesti ulkoisen verkon ja organisaation verkon väliin muodostuu virtuaalinen Edge-instanssi. Edge mahdollistaa palomuurilla ja osoitteenmuunnoksella liikenteen suodatuksen ja ohjauksen sekä edellyttää vCloud Directorilta lähtöisin olevaa osoitteen osoittamista joko sen omalla DHCP:llä tai staattisesti. Lisäksi Edge myös haluttaessa toimii välittävänä nimipalvelimena ulkoisen ja sisäisen verkon välillä.

Eristetyssä verkossa kuvan 20 mukaisesti Edge toimii reitittimenä ilman pääsyä paikallisen verkon ulkopuolelle. Edge toimittaa mahdollisen DHCP-palvelimen roolia, mikäli virtuaalikoneille ei ole asetettu staattisia osoitteistuksia.



(a) Workstationin näkymä.



(b) vSphere Clientin näkymä.

Kuva 21. vSphere Clientin ja Workstationin näymät vastaavat toisiaan.

4.4 VMware Workstation

VMware Workstation on työasemassa ajettava virtualisointisovellus, joka täyttää helposti perustarpeet virtualisointiin tutustuttaessa. Sovelluksella on mahdollista luoda niin perinteisiä virtuaalikoneita kuin erikoisempia useamman kerroksen sisältäviä sisäkkäistä virtualisointia hyväksikäyttäviä virtualisointiympäristöjä. Varsinaista virtuaalikokoelmaa VMware Workstation ei tue, mutta kansiorakenteella lähes vastaava ympäristö on osittain toteutettavissa. Workstation tukee kahtakymmentä eri verkkoa, joista yhdessä kerrallaan voi olla Workstationin oma osoitteenmuunnospalvelu käytössä. Loput verkot ovat joko suoraan sillattuja edellisen sivun kuvan 19 tyyppisesti suoraan työaseman verkkokortin ympäristöön tai paikallisia kuvan 20 kaltaisia verkkoja. Yhteen työaseman fyysiseen verkkoporttiin voi olla sillattu yksi paikallinen virtuaalinen Workstationin verkko, mikä on selkeä puute hiemankaan haastavammissa tarpeissa. Workstation mahdollistaa uusimmissa versioissaan myös suoran yhdistämisen vCenteriin, ESXi-palvelimeen tai toiseen työasemaan, jossa ajetaan Workstationia ja näin kyetään siirtämään virtuaalikoneita ilman ovf-paketointia alustalta toiselle ja hallitsemaan toisessa päässä ajettavia virtuaalikoneita lähes kuin ne olisivat paikallisia. Tämä tuo myös mahdollisuuden muuttaa ja päivittää virtuaaikoneen laitteistoversiota korkeammaksi kuin mitä VMware Infrastructure Client sallii oletusarvoisesti. Näkymät 21a ja 21b ovat näkymät vCenterin virtuaalikonevalikoimaan Workstationin ja vSphere Clientin osalta.

4.5 Openfiler

Insinöörityössä käytettiin ESXi-palvelinten saavutettavissa olevien levykapasiteettien muodostamisessa Openfiler-jakelua. Openfiler on suunnattu erilaisten verkon kautta saavutettavien tallennuskapasiteettipalvelujen luomiseen. Se mahdollistaa esimerkiksi usean ESXi-palvelimen välisen iSCSI-kohteen muodostamisen. Tämä on olennainen komponentti vSpheren vMotion-teknologian toimimisessa. Haluttaessa voidaan Openfilerillä myös muodostaa toisenlaisia jaettuja verkkorakenteita, kuten esimerkiksi jaettuja kansiota. Nämä olisivat omiaan esimerkiksi käyttöjärjestelmien asennuslevykuvien jakamiseen virtuaalikoneiden välillä, jos iSCSI-yhteyden takana olevaa kapasiteettia siihen ei haluta käyttää. Openfiler rakentuu virtuaaliympäristöissä vähintään kahdelle virtuaalikiintolevylle. Ensimmäisessä on jakelun asennus- ja konfiguraatiorakenteet. Toinen virtuaalikiintolevy osoitetaan täysin verkon kautta saavutettaviin käyttötarkoituksiin. Useamman kiintolevyn malli on erityisen kätevä, jos halutaan sijoittaa verkon kautta saavutettava kapasiteetti eri medialle kuin käyttöjärjestelmä itsessään sijaitsee.

iSCSI-toiminnallisuutta luotaessa Openfilerin tallennusmedialle luodaan osio (partition). Osio liitetään taltioryhmään (volume group), mikä liitetään iSCSI-yhteyden kohteen (iSCSI target) kanssa. Yhdellä Openfiler-palvelimella voi olla useita kohteita yhtaikaisesti näkyville verkkoon. Oletusarvoisesti kohteiden näkyminen verkkoon on suljettu ja täten Openfilerille täytyy luoda erikseen listat verkko-osoitteista tai -avaruuksista, joille se haluttua kohdetta tarjoaa käytettäväksi. Osoitetason lisäksi on mahdollista konfiguroida käyttäjäkohtainen käyttäjätunnus-salasana-tunnistepari CHAP-tunnistautumiseen. Tämä osio jaetaan suoraan raakana levypintana blokkitasolla. Openfilerin tapauksessa kuitenkin osio on yhdistettävä ensin osioryhmään (volume group), mikä myöhemmin liitetään iSCSI-kohteen (iSCSI target) kanssa muodostaen toimivan kokonaisuuden.

4.6 Verkkoteknologiat

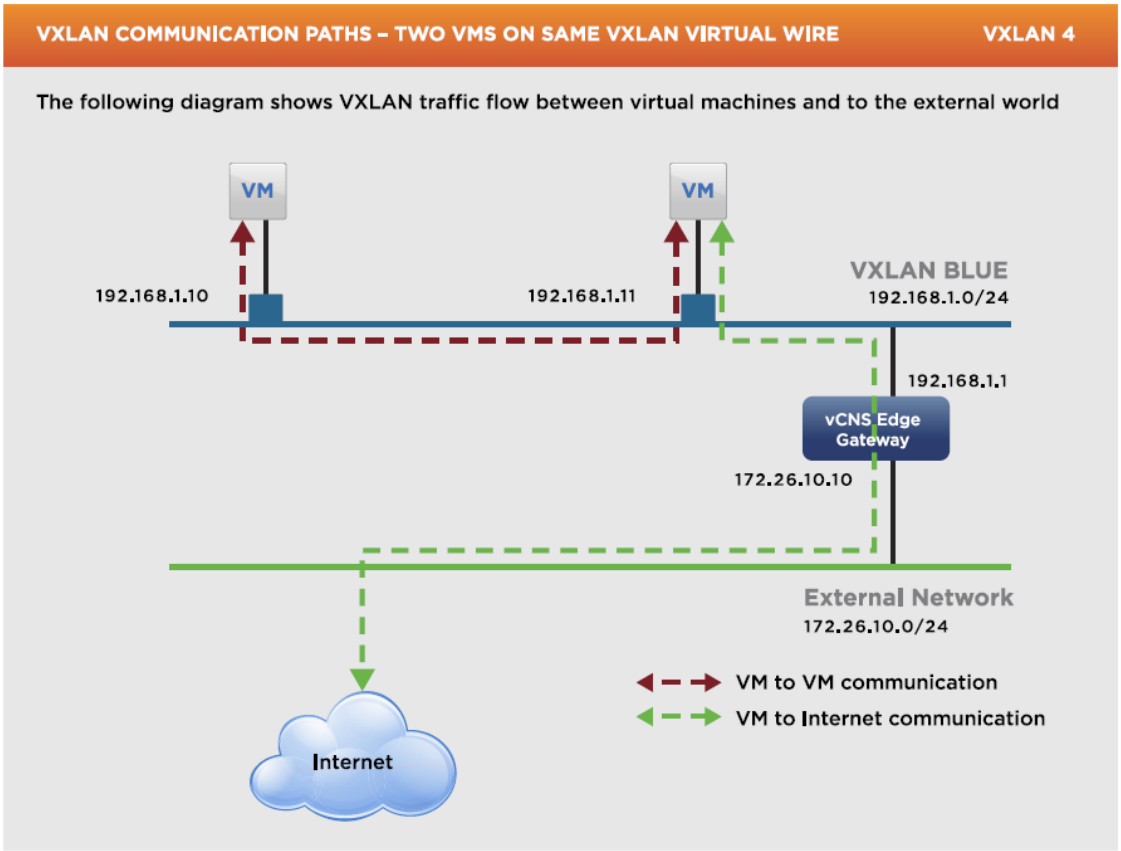
Laitteistovaatimuksien lisäksi virtualisointiympäristöt nojaavat myös verkkotekonologisten ratkaisuiden toimivuuteen ja ominaisuuksiin. Virtuaaliset lähiverkot (VLAN, Virtual Local Area Network) ovat tätä päivää olleet huomattavia aikoja ja uudempina teknologioina VXLAN (Virtual Extensible Local Area Network) ja iSCSI (Internet Small Computer System Interface) ovat työntymässä varteenotettaviksi vaihtoehdoiksi järjestää toimintoja. Tässä työssä verkkoteknologinen käyttö painottui pääosin kahteen viimeisimpään mainituista.

4.6.1 Virtuaalilähiverkot ja VXLAN

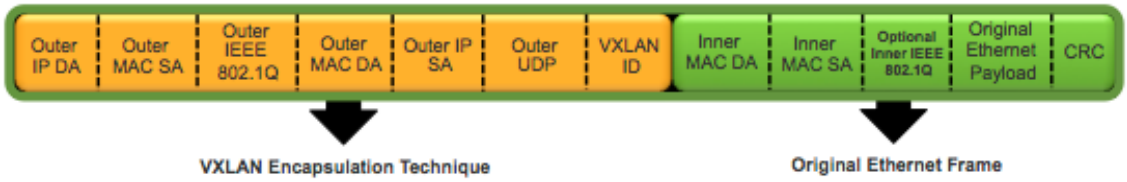
Virtuaalilähiverkkoja (VLAN, Virtual LAN) käytetään, kun halutaan eriyttää samassa fyysisessä verkossa eri verkkosegmenttejä toisistaan helposti ja loogisesti. Oletusarvoisesti ilman virtuaalilähiverkkoja kytkimen kaikki portit kuuluvat samaan verkkosegmenttiin. Lisäksi vituaalilähiverkkojen yhteydessä voidaan yhdistää useampi fyysinen portti jakamaan kuormitusta ja näin nostaa kokonaistiedonsiirtokapasiteettia yli yksittäisen porttikohtaisen maksimikapasiteetin. Virtuaalilähiverkkojen portit voivat olla kahdenlaisia: natiivia virtuaalilähiverkkoaan tunnistavia access-tyyppisiä tai trunk-tyyppisiä. Trunk-tyyppinen portti välittää liikennettä paketoinnin saattelemana ja access-tyyppinen portti paketoinnin purkaen sen vastaanottajalle päin. Virtuaalilähiverkot erotetaan toisistaan 12-bittisen tunnisteen kautta, mikä rajoittaa niiden määrän 4094 kappaleeseen. Insinöörityön luonteesta johtuen virtuaalilähiverkkoja ei enempää käsitellä.

Etäresurssiympäristössä toisistaan erotettavia verkkosegmenttejä tarvitaan helposti huomattavasti useampia. Tähän ratkaisuna VMware ja Cisco tarjoavat VXLAN-teknologian [33]. Virtuaalilähiverkkojen 12-bittisen tunnisteen sijaan VXLAN-teknologia käyttää 24-bittistä VNI-tunnistetta (VXLAN Network Identifier), mikä mahdollistaa noin kuudentoista miljoonan verkkosegmentin luomisen samaan infrastruktuuriin. VNI-tunniste yksilöi virtuaalikoneen MAC-osoitetiedon perusteella liikenteen lähtösijainnin ja luo sille yksilöllisen tunnisteen. Tällöin kahden saman MAC-osoitteen omaavan virtuaalikoneen liikenne pysyy erillään toisistaan, koska liikenne kulkee VNI-tunnisteen alla. Liikenne VXLAN-tunnisteella

varustettuna kulkee erityisten VTEP-tunneleiden (VXLAN Tunnel End Point) välillä. Tällainen tunneli sijaitsee joko hypervisorilla tai verkkolaitteessa --- joko ohjelmisto- tai laitteistopohjaisena.



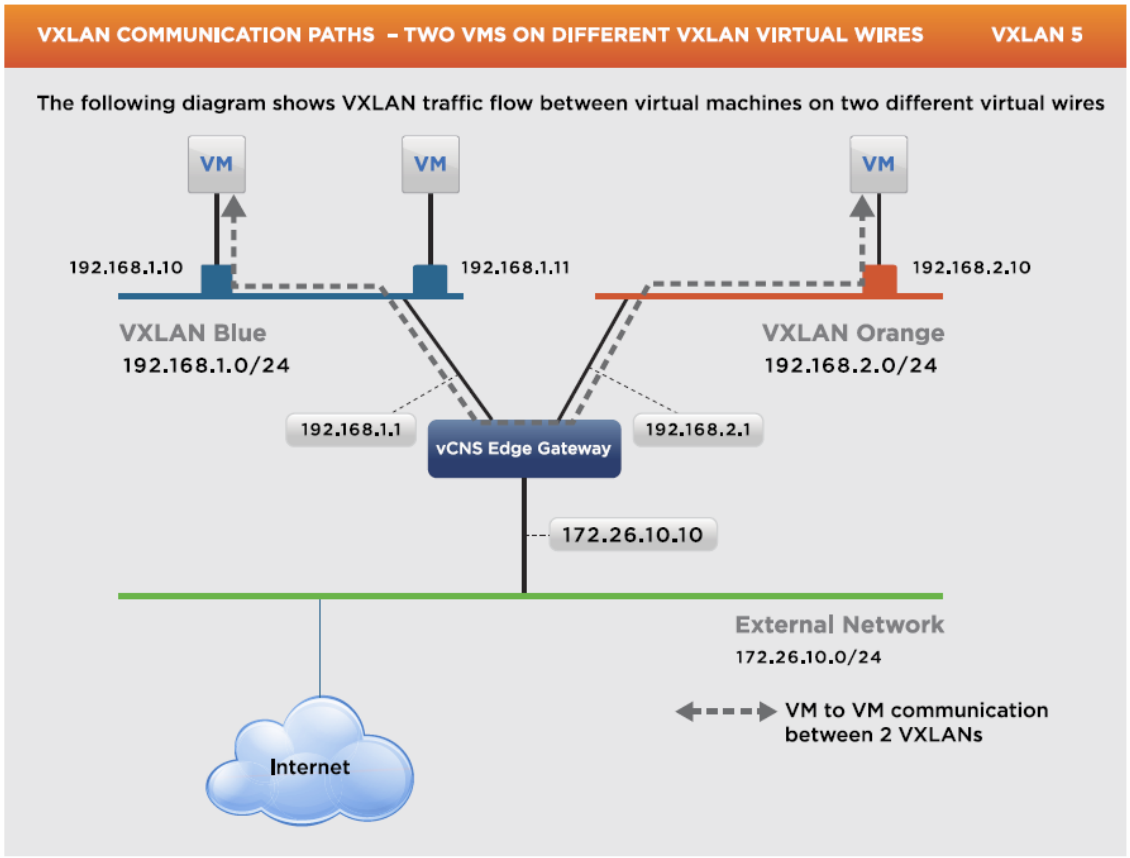
Kuva 22. Violetti liikenne tapahtuu saman VNI:n sisällä ja vihreä on VNI-alueesta ulospäin pyrkivää liikennettä. [34]



Kuva 23. VXLAN framen rakenne eroaa suuresti perinteisestä Ethernet framesta sisältäen olennaisesti enemmän informaatiota [35].

Saman VNI:n sisällä kommunikoidessa virtuaalikoneen ei oleteta tietävän mitään VXLAN-konfiguraatiosta kuvan 22 ympäristössä. Virtuaalikone lähettää MAC-kyselyn VTEP:lle, joka tarkastaa, mihin VNI-segmenttiin virtuaalikone kuuluu. Tämän jälkeen VTEP pyrkii selvittämään, onko tavoiteltava MAC-osoite myös samassa segmentissä. Mikäli näin on, kapseloidaan paketti ulkoisen MAC-osoitteen, VNI:n ja IP-otsakkeen avulla kuvan 23 mukaisesti. Toisessa päässä vastaanottajaa lähinnä oleva VTEP purkaa kapseloinnin vas-

taanottavan virtuaalikoneen MAC-osoitteen ja VNI:n tarkistuksen jälkeen ja paketti päätyy toivotulle vastaanottajalle sille ymmärrettävässä muodossa. Samassa yhteydessä purkava VTEP tallentaa lähettävän virtuaalikoneen MAC-osoitteen taulukkoonsa, jolloin vastatessa lähettäjälle ei ole tarvetta erikseen selvittää sitä. Kuvassa 24 hahmotellaan, miten kahden rinnakkaisen VXLAN-verkon liikenne toimisi.



Kuva 24. VXLAN-verkon liikenne kahden eri verkon välillä [34].

Vastaanottavan tahon MAC-osoitteen selvitys on periaatteessa VXLAN-verkoissa toteutettu virtuaalikoneen näkökulmasta aivan kuin ei-VXLAN -verkoissa. Tosiasiassaselvityksessä käytetään yleislähetysteknologiaa (Broadcast), minkä sisään ryhmälähetyksen (Multicast) paketointi on kapseloitu. Yleislähetystä käytetään lähetettäessä tietoa ennalta määräämättömälle vastaanottajajoukolle. Tällöin saadaan tietoa esimerkiksi verkon päätelaitteista. Tyypillinen yleislähetys on ARP-kysely (Address Resolution Protocol), jolla haetaan tietyn IP-osoitteen omaavan päätelaitteen MAC-osoitetta. Ryhmälähetyksessä lähetetään määrätylle joukolle tietokokonaisuus. Jotta paketteja kyetään vastaanottamaan ryhmälähetyksellä, on ne erikseen tilattava. Ryhmälähetystä käytetään esimerkiksi videoneuvotteluissa ja IPTV-palveluissa liikennöintikaistan säästämiseksi. Täsmälähetyksessä

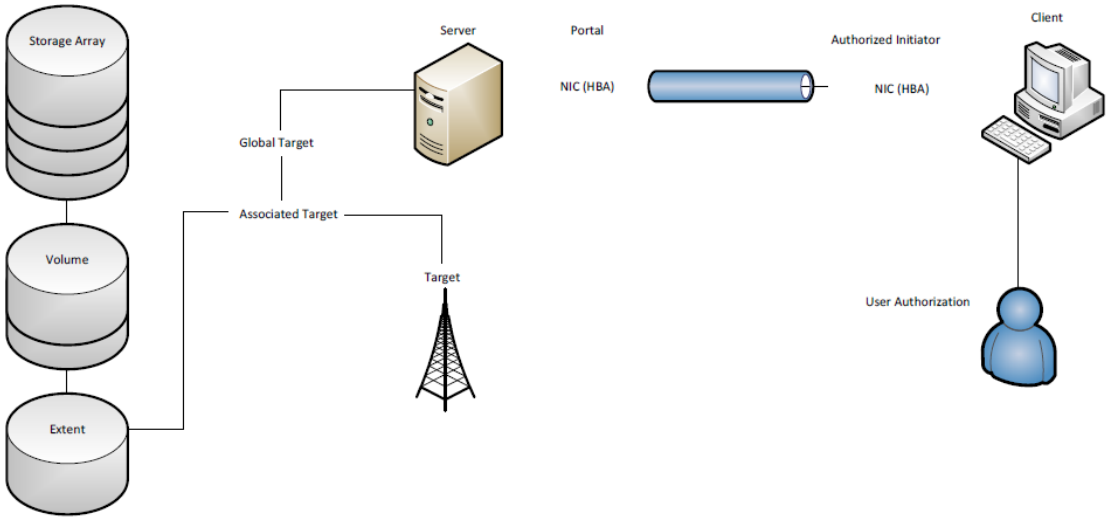
(singlecast, unicast) on yksi lähettäjä ja yksi vastaanottaja, joista molemmat yksilöity yksilöllisellä tunnisteella.

Perinteisen VLAN-verkon piirissä toimittaessa virtuaalikoneen halutessa tavoitella toista virtuaalikonetta, lähettää tämä ARP-pyynnön yleislähetyksenä ja saa vastineeksivastaanottajan MAC-osoitteen mikäli ne ovat samassa verkkosegmentissä. VXLAN-ympäristössä kuitenkin tämä on hivenen hankalampaa, ja ARP-kyselynotsakkeeseen lisätään VNI-tunniste IP- ja UDP-otsakkeiden rinnalle kuvan 23 tapaan. Poikkeuksellisesti kuitenkin tämä yleislähetys toteutetaan ryhmälähetyksenä sen ryhmän sisällä, johon lähettävän virtuaalikoneen VTEP kuuluu.

Jotta tämä onnistuisi, VNI-tunnisteen ja ryhmälähetystunnisteiden tiedot on oltava selvillä VTEP:llä. Tämä liittäminen tapahtuu esimerkiksi protokollariippumattoman ryhmälähetyksen avulla (Protocol Independent Multicast - Sparse Mode, PIM-SM) tai kaksisuuntaisen protokollariippumattoman yleislähetyksen (bidirectional PIM, BIDRI-PIM) avulla. Näitä ei työn luoteen johdosta käsitellä tarkemmin. Lähettävä virtuaalikone selvittää kohdeosoitteen täsmälähetyksen avulla. Tämän paketoinnin VTEP kapseloi ja lähettää sen ryhmälähetyksellä. Vastaanottavan virtuaalikoneen VXLAN-verkkoon yhdistävä VTEP purkaa sen ja virtuaalikone vastaa ARP-kyselyyn. VXLAN-verkkojen toiminnan edellytyksenä on pakettien katkeamattomuus VTEP-pisteiden välillä. Tämä täyttyy konfiguroimalla suuremmat kehykset käyttöön. Perinteinen Ethernet-kehys kykenee kuljettamaan hyötykuormaa 1500 bittiä, kun suurkehys kykenee kuljettamaan jopa 9000 bittiä.

4.6.2 iSCSI-teknologia

iSCSI-teknologian toimivuus pohjimmiltaan nojaa kuvan 25 rakenteeseen eli levypalvelimen yksittäisiin kiintolevyihin ja niiden päälle luotaviin lisäpalveluihin. Kiintolevyjen päälle luodaan osiorakenteita ja siitä on allokoitavissa iSCSI-käyttöön soveltuvasti kapasiteettia. Kapasiteetin käyttämiseksi palvelimelle luodaan asianomaisen verkkoympäristön, joko ethernetin tai kuituverkon, tasoinen tunniste. VMware vSphere-ympäristössä käytettävät levykapasiteetit voidaan jakaa laitteen sisäisen protokollatason tunnisteen (LUN, Logical



Kuva 25. iSCSI:n funktionaalinen rakenne [36].

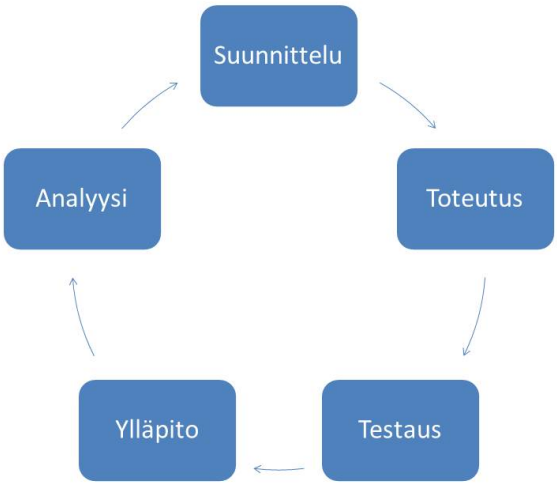
Unit Number) tasoisesti kohteisiin ja jatkeisiin (extent). Tunnistetut kohteet (Associated Target) ovat ensisijaisesti levykapasiteetin ja mahdollisen tunnistautumisen kautta esitettävien kohteiden yhteenliittymiä. Yhdistettäessä tunnistettuja kohteita kutsutaan alkuperäiseen kohteeseen liitettäviä varantoja levylaajenteiksi.

VMware vSphere-ympäristö nojautuu lohkopohjaisissa järjestelmissä VMwaren omaan VMFS-tiedostojärjestelmään (Virtual Machine File System). Uusin versio, VMFS5, tukee jopa 64 TB:n osiokokoja ja 62 TB:n yksittäisen tiedostokokoa. Aiempi VMFS3-versio tuki ainoastaan 2 TB kokoisia levyjä (storage device) ja osiota sekä levylaajenteita (extent). Lisäksi VMFS3 käytti useampaa eri blokkikokoa (block size), mikä rajoitti yksittäisen tiedoston koon pienemmillään ainoastaan 256 GB:n kokoiseksi. Tiedostojärjestelmän blokki on pienin yksikkö [37], joka siitä luetaan tai kirjoitetaan. Virtuaalikoneen allokoidessa keskusmuitinsa kapasiteetin verran levytilaa käyttöönsä, kuten luvussa 2.2 todettiin virtuaalikoneen varaavan levyjärjestelmästä keskusmuistikapasiteettinsa verran tilaa, asettaisi jo tiedostojärjestelmänä VMFS3 rajoituksia virtuaalikoneiden keskusmuistikapasiteetille. Esimerkiksi vSphere 5.5 tukee 1 TB:n virtuaalikeskusmuistilla varustettuja virtuaalikoneita. Näitä käyttävät esimerkiksi SAP:n HANA [38].

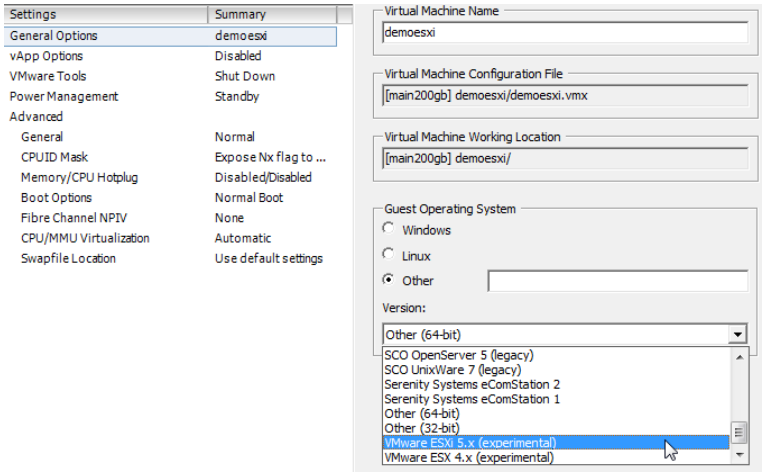
5 Käytännön toteutus

5.1 Insinöörityön käytännön osuus

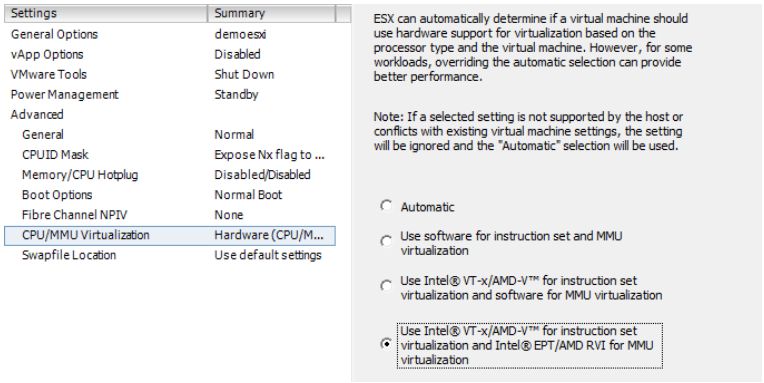
Käytännön osuus insinöörityössä suoritettiin kahdessa osassa: aluksi suoritettiin pilottitestaus, jossa järjestelmä rakennettiin ja testattiin soveltuvin osin pienemmässä testiympäristössä. Pilottivaiheessa testattiin erilaisia sovellusvaihtoehtoja muun muassa iSCSI-verkkolevyjen, verkkokonfiguraatioiden suhteen. Pilottivaiheen jälkeen toimiva ympäristö saaavutettiin perinteisen iteraatiokehitysmallin kuvan 26 tapaan. Virtuaaliympäristöstä johtuen sekä pilottitestaus että tuotantoversio tehtiin modifioiduilla virtuaalikoneilla. Tämä tapahtui käyttämällä sisäkkäisesti virtualisoituja virtuaalikoneita. Myöskin pilottiympäristön testaus tapahtui sisäkkäisesti virtualisoiduin virtuaalikonein VMware Workstationilla.



Kuva 26. Iteraatiokehitysmalli sovellettuna etäresurssiympäristön kehitykseen.



Kuva 27. VMware Workstationin käyttöjärjestelmäversion valintanäkymä virtuaalikoneen luomisen jälkeen.



Kuva 28. Käyttöjärjestelmän valinnan yhteydessä on sallittava laitteistotason käskykantojen toimivuus ja läpinäkyvyys virtuaalikoneelle.

Sisäikkäisten virtualisointikerrosten luonti

Sekä pilottiosiossa että lopullisessa versiossa työ toteutettiin sisäkkäisiä virtualisointikerroksia käyttäen. Sisäkkäiset virtualisoinikerrokset mahdollistettiin luomalla virtuaalikone ja se tyyppimääriteltiin ESXi-yhteensopivaksi jo luontivaiheessa käyttämällä Other (64-bit) -aihiota ja jälkikäteen valiten oikea kuvan 27 versio ja lisäämällä sille erityinen VT-x/EPT:n tai AMD-V/RVI:n kuvan 28 salliva valinta ominaisuuksiin. Lisäksi asetettiin ESXi:n päällä pyörivän virtuaalikoneen käytettäville fyysisen palvelimen virtuaalikytkimelle ja -porteille erityinen Promicious-valinta. Virtuaalikoneen luomisen jälkeen konfiguraatiotiedostoon lisättiin rivi "vhv.allow = true"ja hypervisorin asennus toimitettiin alusta loppuun aivan kuten se suoritettaisi suoraan laitteistolle virtuaalikoneen sijasta.

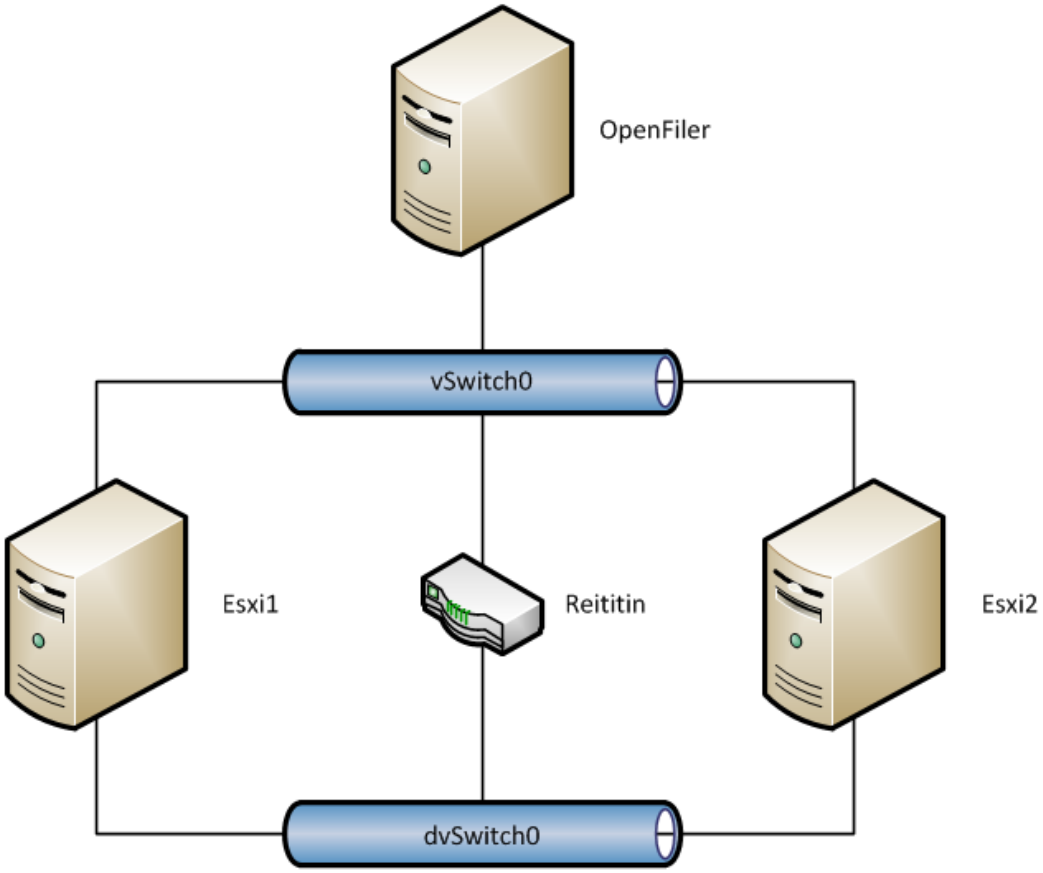
5.2 Pilottiympäristö

Pilottitestaus suoritettiin kuluttajaympäristössä kuluttajamarkkinoilta löytyvillä laitteilla. Poikkeuksena oli natiivisti kahdeksan ytimen prosessori (Intel Xeon E5 QA90 ES), joka on erityinen Engineering Sample, jossa oli tarpeelliset VT-d ja VT-x ominaisudet enabloituina toisin kuin vastaavissa normaalisti myynnissä olevissa malleissa kyseisellä ajanjaksolla olevissa versioissa. Lopullinen kokoonpano koostui 64-bittisestä Windows 7 käyttöjärjestelmästä, jossa virtuaaliympäristöä ajettiin virtuaalisesti sisäkkäisesti virtualisoidussa moodissa VMware Workstation 8 -virtualisointiohjelmistossa, edellä mainitusta Intel Xeon E5 QA90 -prosessorista, Asus Sabertooth X79 -emolevystä, 56 gigatavusta keskusmuistia sekä 256 gigatavun SSD-levystä. Kiintolevyn flash-pohjaisuutta perusteltiin usean virtuaalikoneen yhtäaikaisen levyn käytön intensiivisyydellä. Tämän seikan ja sisäkkäisen virtualisoinnin myötä perinteinen mekaaninen ja RAID-teknologialla nopeuttamaton kiintolevyteknologia ei olisi ollut mielekästä ympäristön verkkaisuuden vuoksi. Empiirinen kokemus oli osoittanut, että suurin haaste yli yhden virtuaalikoneen ajamisessa yhdellä työasemalla on levyn IOPS-kyky pullonkaulana (Input / Output Operations Per Second), jonka vuoksi käyttöön valittiin SSD-pohjainen kiintolevy. Intelin X79 -piirisarjaan pohjautuva ratkaisu valittiin pilottitestausympäristöksi sen mahdollistaman perinteistä suuremman muistikapasiteetin hallinnan vuoksi. Kaksikanavaisen muistiohjaimen sijaan X79-piirisarjalliset LGA2011-kantaiset prosessorit ovat varustettuja nelikanavaisella muistiohjaimella, jotka mahdollistavat kaksinkertaisen muistikapasiteetin hallinnan kanavakohtaisen muistikampojen lukumäärän pysyessä vakiona. Jokaiseen muistikanavaan laitettiin kaksi muistikampaa: kolmen kanavan osalta käytettin 8 GB:n kampoja ja yhden osalta kahta 4 GB:n muistikampaa, jolloin saatiin kokonaismuistikapasiteetiksi 56 GB.

Pilottitestaus aloitettiin luomalla taulukon 3 mukaiset virtuaalikonekokonaisuudet. Kokonaisuuteen kuului kaksi virtuaalista ESX-virtualisointipalvelinta kahdeksalla virtuaaliytimellä, suoralla siltauksella kahden verkkokortin voimin fyysiseen verkkoon tietokoneen verkkokortin kautta sekä allokoimalla ensimmäiselle virtuaalikoneelle 24 GB ja toiselle 16 GB. Paikallista levytilaa jaettiin riittävästi hypervisorin asentamista varten, muttei enempää, sillä kaikki sisältö tulee olemaan iSCSI-yhteyden päässä verkossa. Samassa verkossa, sivun 52 kuvan 29 mukaisesti, käytetään kahta verkkokorttia, koska toinen verk-

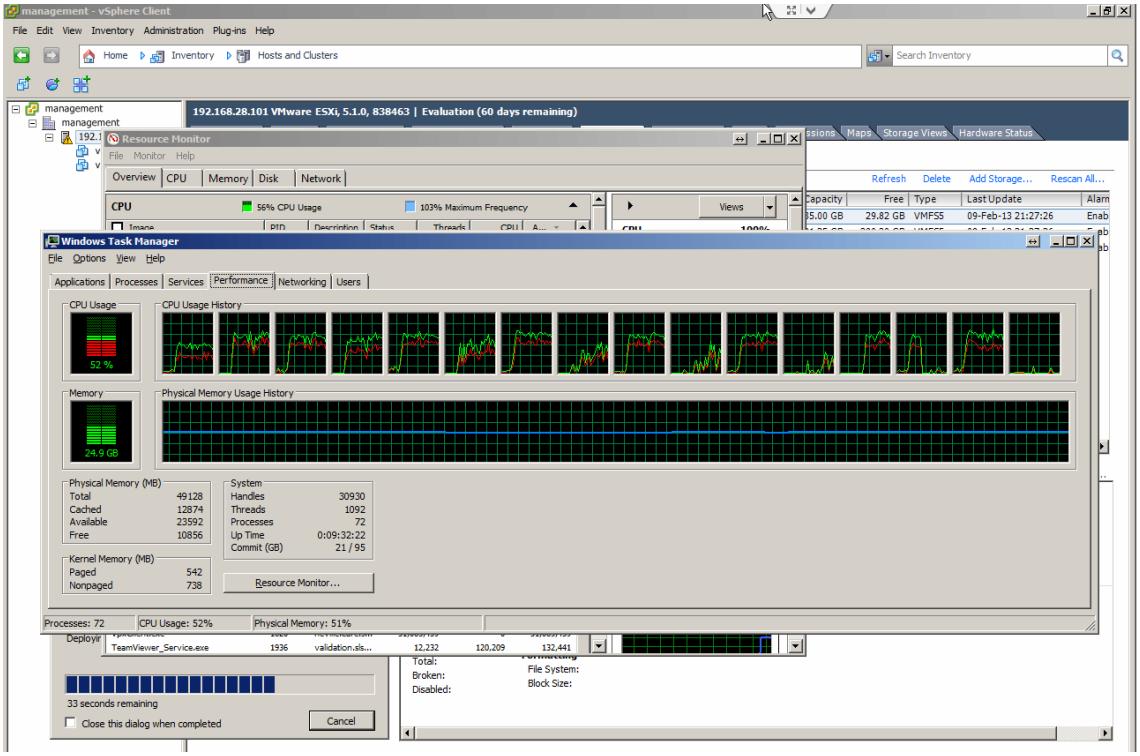
Taulukko 3. Virtuaalikoneiden laitteistoresurssit pilottiympäristössä.

	Esxi01	Esxi02	Openfiler
Prosessoriytimet (kpl)	4	4	2
Keskusmuisti (GB)	24	16	2
Kiintolevytila (GB)	40	40	8 + 200
Verkko 1	vSwitch0	vSwitch0	vSwitch0
Verkko 2	dvSwitch0	dvSwitch0	-



Kuva 29. Pilottiverkon rakenne.

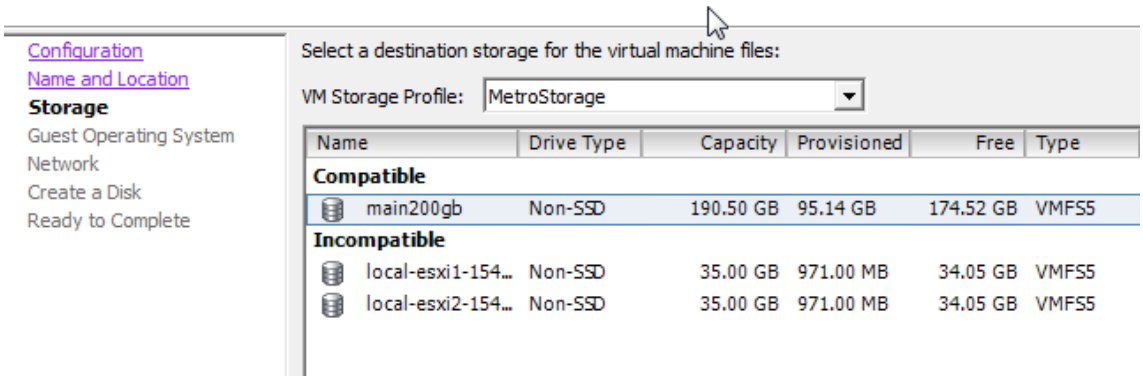
kokortti on asetettava yksinomaan hajautetun virtuaalikytkimen käytettäväksi, jossa etäresurssissa luotavien ja siellä pyörivien virtuaalikoneiden verkko sijaitsevat. Seuraavaksi luotiin Openfiler 2 GB:n keskusmuistilla ja kahdella ytimellä varustettuna sekä sillaten se suoraan samoihin verkkoihin aiempien koneiden kanssa ja allokoiden sille 200 GB kapasiteettia nopealta SSD-levyltä. Kuva 30 näyttää asennusvaiheessa kuormituksen jakautumisen useamman prosessoriytimen kesken.



Kuva 30. Pilottiympäristön kuormitustestausta sisäkkäisillä virtualisointikerroksilla.

Pilottiympäristön resurssi-infrastruktuuri

Openfilerin konfiguroinnin ja iSCSI:n toiminnallisten edellytysten muodostamisen jälkeen virtualisointipalvelimet konfiguroitiin käyttämään iSCSI-toiminnallisuutta Openfileriltä. Tämän jälkeen vCenter-hallintapalvelin tuotiin suoraan vSphere Clientin kautta. Tuomisen jälkeen vCenterin perusasetukset annettiin erityisen selainpohjaisen käyttöliittymän kautta ottamalla yhteys porttiin 5480 HTTPS-yhteyden välityksellä. Seuraavaksi vSphere Client ohjattiin vCenteriä kohti ja aloitettiin hajautetun kytkimen (dvSwitch0) alustus. Virtualisointipalvelimista luotiin palvelinrypäs testiympäristön datacenteriin. dvSwitch0:lle osoitettiin molemmilta virtualisointipalvelimilta yksi verkkokortti. Paikallisen vSwitch0:n Management Network -porttiryhmälle lisättiin tuki vMotionille vMotionin toiminnan saavuttamiseksi. Ryppääseen lisättiin tuetuiksi ominaisuudeksi DRS, sen ollessa edellytys vCloudille. Yhden ryppään integroitua mallia päätettiin käyttää tuotujen virtuaalikoneiden verkkorakenteiden mallien rajoitusten vuoksi verkkokorttien osalta. Virtuaalikoneissa oli staattinen määrä verkkokortteja ja esimerkiksi käsin verkkokorttia lisättäessä, vCenterin web-käyttöliittymä hajosi käyttökelvottomaksi. Käytännössä siis virtuaalikoneisiin saatiin lisättyä verkkokortti, mutta tämä rikkoi muun muassa selaimella käytettävän käyttöliittymän. Ryppään luonnin



Kuva 31. Luotaessa virtuaalikonetta, tallennusprofiili rajoittaa käytettävissä olevia levyvarantoja.

ja konfiguroinnin jälkeen luotiin tallennusprofiili etäresurssissa tehtävien virtuaalikoneiden puitteiksi.

Alustavan hallintainfrastruktuurin pystyttämisen ja sen hallintaryppään luomisen jälkeen siirryttiin etäresurssi-infrastruktuurin tuomiseen ympäristöön. Kaikki aloitettiin vShield Managerin pystyttämisellä, konfiguroinnilla sekä liittämällä vCenter siihen. vShield Manageriin otettiin vCenterin tapaisesti yhteys HTTPS-yhteyden yli porttiin 5480 ja yhdistetään vCenter siihen. Tämän jälkeen DRS:n avulla etäresurssiryväs siirrettiin yhdelle virtuaalipalvelimelle toisen palvelimen Edgen vCenter pluginin infrastruktuuri-integraation vuoksi. Asennuksen aikana palvelin ajetaan erityiseen huoltotilaan, jossa siihen vCenterin toimesta asennetaan erityinen vib-paketti. Asennuksen onnistumisen havaittiin muun muassa vSphere Clienttiin ilmestyvästä liitännäisestä.

vShield Managerista luotiin virtuaaliverkko, jonka reititintä, nimipalvelinta ja palomuuria Edge hallitsee. vShield Managerissa myös tässä vaiheessa pitäisi luoda VXLAN-verkon ryppään palvelinten välille, mikä edellyttää jumbokehysten konfiguroimista toimiviksi. Useamman hetken tutkimisen jälkeen huomattiin, ettei VMware Workstation tue VXLAN-verkkoja perinteistä suurempien jumbokehysten tuen puutteen vuoksi eikä niitä siten saada toimiviksi testiympäristössä eikä myöskään testiympäristön reititin tukenut niitä. Verkon yksilöintiin käytetyt Segment ID -kentät kuitenkin ovat konfiguroitavissa ja käytettävissä. Huomattiin, ettei vCloud Directorin ryppääseen tuotua versiota ole mahdollista konfiguroida esimerkiksi verkkokorttien, joita on vain kaksi kappaletta, tai verkkossa esiintyvän työ-

aseman nimen osalta, jolloin koko ympäristö ei juuri hyötynyt omasta DNS-palvelimista. Kaksi verkkokorttia on minimivaatimuksena, mutta oikea esimerkillinen eriytetty ympäristö vaatisi vähintään neljä verkkokorttia: kaksi ulkoverkkoon ja kaksi sisäverkkoon, jossa muu virtualisointi-infrastruktuuri sijaitsisi.

Tuotantoversiossa siis olisi tarkoitus saada ulkoverkosta yhteys ainoastaan vShield Managerin Edge-tyyppiseen virtuaalikoneeseen sekä vCloud Directorin kahteen verkkoliittymään. Proof of Casen luonteesta johtuen tähän ei kuitenkaan päädytty ja lopullinenkin versio päätettin toteuttaa edellä olevasti yksinkertaisena verkkona. vShield Managerin valmistuttua siirryttiin vCloud Directoriin, johon yhdistettiin vShield Manager sekä vCenter siihen kuuluviine virtualisointipalvelimineen. vCloud Directorissa luotiin virtuaalinen palveluntarjoaja, joka sai käyttöönsä vCenterin kautta virtualisointiapalvelimet ja tallennusprofiilinsa mukaisen iSCSI-levykapasiteetin. Seuraavaksi luotiin virtuaalinen palveluntarjoaja ja sille virtuaalinen tietotekonekeskus. Tämän yhteydessä vCenter loi sekä DRS-poolin että dvSwitch0:lle porttiryhmän vCenteriin. Seuraavaksi virtuaaliselle palveluntarjoajalle annettiin ulospäin näkyvästä verkosta verkkoallokaatio IP-osoitteista, joita Edge-virtuaalikoneet tai virtuaalikoneet yleensä suoraan saavat varata vCloud Directorin kautta. Lopuksi luotiin virtuaalinen organisaatio, jonka tietokonekeskusta virtuaalinen palveluntarjoaja ajaa virtuaalisessa tietokonekeskuksessaan ja joka saa myös oman DRS-poolinsa erillään Edge-virtuaalikoneista. Organisaation datacenterille määriteltiin minimirajat, joita sen virtuaalikoneiden on noudatettava muun muassa käynnissä olemisen ja resurssien riittävyyden suhteen. Tässä vaiheessa valittiin myös resurssiallokaatiomalli, joka vaikuttaisi vCenter Chargebackin käyttöön ja suorasti kaikkiin muihin ryppään käyttäjiin varaamalla tai olemalla varaamatta resursseja. Tarvittaessa olisi ollut myös mahdollista rajoittaa verkkoliikenteen lähteää/tulevaa kaistaa verkkokohtaiseti. Valmiin organisaation kanssa seuraavaksi luotiin virtuaalikatalogi virtuaalikoneista, joita organisaation jäsenet voivat varata ja käyttää hyödykseen. Etäresurssiin virtuaalikoneita luotiin sekä tyhjästä liikkeelle lähtien että suoraan tuomalla niitä valmiina. Jotta virtuaalikoneista saataisiin etäresurssikelpoisia, oli perinteisen asennusten ja VMwaren eritystyökalujen konfigurointien lisäksi virtuaalikoneille sallittava vieraskustomointisääntö, jonka sallimana vCloud Director säätää eritoten verkkoasetukset kuntoon.

Taulukko 4. Fyysisen laitteistotason näkökulmasta virtuaalikoneiden koostumukset.

Virtuaalikone	ESXi01	ESXi02	Openfiler
Prosessoriytimet (kpl)	4	4	2
Keskusmuisti (GB)	24	24	2
Kiintolevykapasiteetti (GB)	40	40	8 + 200
Verkkokortti 1	vCloud	vCloud	vCloud
Verkkokortti 2	vCloud	vCloud	-

5.3 Toteutusvaiheen koejärjestelmä

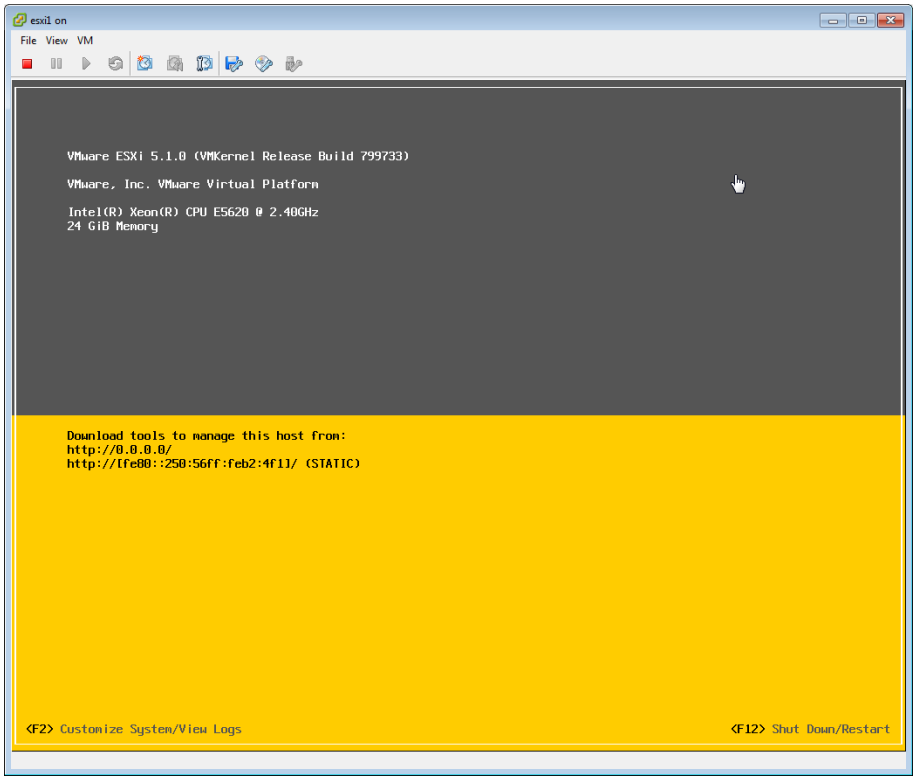
Pilottijärjestelmän toteuttamisen ja sen osa-alueisiin tutustumisen jälkeen siirryttiin luomaan insinöörityön varsinaista järjestelmää. Pilotin jälkeinen toteutusvaiheen koejärjestelmä toteutettiin täysin virtuaalisena hyväksikäyttäen sisäkkäistä virtualisointia. Fyysisessä laitteistossa ajettiin vSphere-ympäristössä kolmea virtuaalikonetta, joista kaksi kykenivät sisäkkäiseen virtualisointiin ja kolmas toimi jaetun iSCSI-levykapasiteetin palvelimena.

5.3.1 Openfiler

Jaetuksi levypalvelimeksi ympäristöön asennettiin alan jakelu Openfiler, jolle lisättiin hypervisoreiden käyttöön 200 GB:n kiintolevyosio asennuslevyn lisäksi. Välittömän asennuksen ja peruskonfigurointien jälkeen Openfilerin säätämistä jatkettiin sen web-käyttöliittymän kautta. Seuraavaksi palvelimen iSCSI-toimivuuden kannalta kriittiset iSCSI Target- ja iSCSI Initiator -prosessit käynnistettiin ja lisättiin automaattisesti käynnistyvyiksi palvelimen käynnistymisen yhteydessä. Tämän jälkeen iSCSI-kohteelle luotiin nimi, jonka perusteella se myöhemmin tunnistetaan. Tyhjälle 200GB:n osiolle luotiin tyhjä osio, joka lisättiin vcloud-nimiseen levykokonaisuuteen edelleen allokoitavaksi. Levykokonaisuus liitettiin vcloud-levyryhmään. Koko levykapasiteetti allokoitiin eteenpäin iSCSI-yhteyden päähän sopivaksi ja koko ryhmän levykapasiteetti on käytössä. Seuraavaksi kyseinen rypäs osoitettiin ensimmäiseksi jaettavaksi LUN-kokonaisuudeksi. Optionaalisen tunnistautumisen lisäksi Openfileriin määritettiin verkot, joista voidaan toiveikkaasti olla yhteyksissä levykapasiteettia tavoitellessa. Tämä tehtiin määrittelemällä verkkosegmentti, joka myöhemmin lisättiin sallittujen verkkoavaruuksien listoille Openfileriltä.

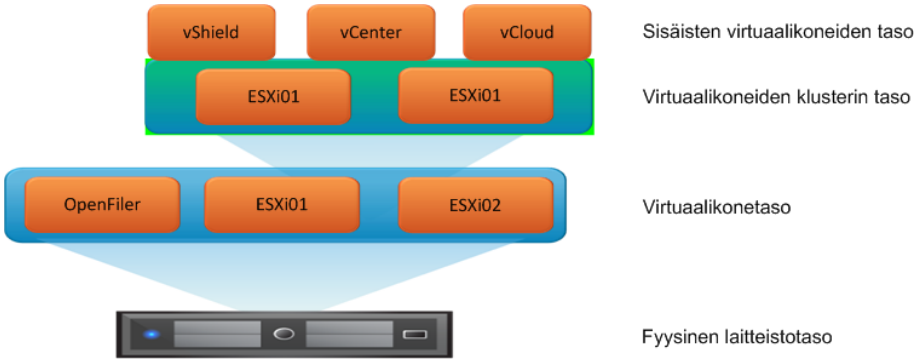
5.3.2 Virtualisoidut ESXi-virtualisointipalvelimet

Virtuaaliset ESXi-virtuaalikoneet luotiin sekä hallinta- että resurssiryppäille ja ne on varustettu 24 gigatavun keskusmuistilla, neljällä virtuaaliprosessoriytimellä sekä molemmilla on kaksi verkkokorttia samassa verkossa. Toinen verkkokortti on tarkoitettu hallintaverkkokortiksi ja toinen hajautetun kytkimen osapuoleksi. Virtuaalikoneet valmisteltiin aiemmin kuvailtujen sisäkkäisten virtualisointikerrosten edellytysten mukaisesti. Normaalien asennusrutiinien jälkeen eteen avautuvassa kuvan 32 konsolinäkymässä näkyy palvelimen olennaismmat tiedot.



Kuva 32. Konsolinäkymä virtuaalisesta ESXi-palvelimesta.

Aluksi palvelimille asetettiin oikeat osoiteasetukset nimipalvelimen, yhdyskäytävän ja oman osoitteistuksen osalta. Palvelinta siirryttiin konfiguroimaan Windowsissa ajettavan VMware vSphere Infrastructure Clientin kautta (VI client). Ensimmäisellä yhteyskerralla hyväksyttiin SSL-varmenne. Palvelimet kytkettiin käyttämään ulkoista aikapalvelinta kellojen ajastamiseksi. Sisäisille virtualisointipalvelimille lisättiin ohjelmistopohjainen iSCSI-sovitin. Seuraavaksi sovittimelle osoitettiin haettavaksi kohteeksi IP-osoite, josta Openfiler levykapasiteettiaan tarjoaa. Osoite skannattiin saavutettavissa olevista I/O-laitteista ja havait-



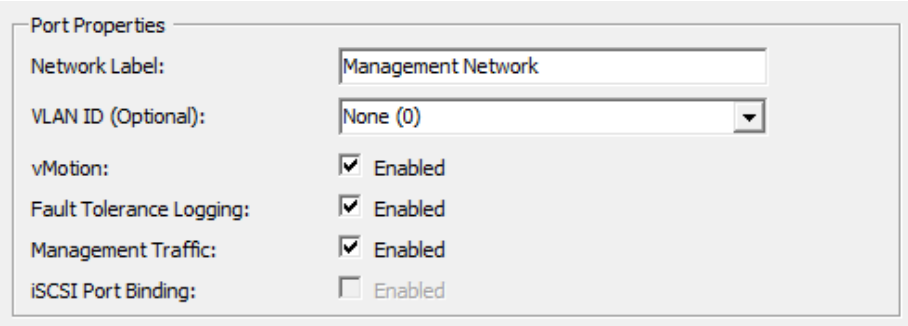
Kuva 33. Insinöörityön toteutusvaiheen järjestelmän rakenne.

tiin saavutettavaksi. Kyseiselle iSCSI-levylle tehtiin VMFS5-muotoinen tiedostojärjestelmä. Levy ilmestyi alustettuna ja kapasiteetti käytössä molemmille ESXi-palvelimille automaattisesti.

5.3.3 Toteutusvaiheen koejärjestelmän hallintaryppään konfigurointi

Seuraavaksi tuotiin toiselle palvelimelle vCenter-hallintaohjelmisto valmiina OVF-paketoituna. vCenter appliance on valmispaketoinnin suhteen siten harvinainen, ettei tuomisvaiheessa virtuaalikoneelle anneta IP-osoitteistustietoja, vaan ne annettiin myöhemmin konsolin kautta käynnistymisen jälkeen. Tuomisen loputtua ja virtuaalikoneen käynnistyttyä ajettiin konfigurointiskripti verkkoyhteyksien kuntoonsaamiseksi. Tämän jälkeen siirryttiin suorittamaan alkeiskonfigurointi selaimen kautta valiten oletusasetukset. Alkeiskonfiguroinnissa hyväksyttiin VMwaren koekäytön ohjeistuksen [39] mukaisesti oletusarvot valittavien vCenterin tietokannan sekä laajan kirjautumisvalinnan (SSO, Single Sign On) osalta. Tämän jälkeen yhdistettiin aiemmin konfiguroidut ESXi-virtualisointipalvelimet vCenter-hallintapalvelimelle luotuun datacenterin ryppääseen ja ryppäästä konfiguroitiin DRS-tuettu.

Ohjeistuksesta ja parhaista mahdollisista toimenpiteistä [40] poiketen vMotioniin käytettiin hallintaverkkokorttia ja ympäristön luonteesta johtuen voitiin näin myös tehdä. Toimiakseen vMotion edellyttää porttiryhmän nimien olevan identtisiä kaikilta siihen osallistuvilta palvelimilta. Virtualisointipalvelimien toinen verkkokortti liitettiin seuraavaksi luotavaan hajautettiin dvSwitch-virtuaalikytkimeen. Käytännössä olisimahdollista konfiguroida



Kuva 34. VI clientin vMotion-teknologian käyttöönottoasetukset hallintaverkon osalta.

dvSwitch ainoastaan yhdellä verkkokortilla siten, että verkkokortti nostetaan virtuaalisen kytkimen uplink-portiksi. Tämä kuitenkin jätettiin käytännön syistä toteuttamatta. Lopuksi virtuaaliympäristöön luodut levyprofiilit liitettiin Openfileriltä näkyvä kapasiteetti niihin.

5.3.4 vCloud Director ja vShield

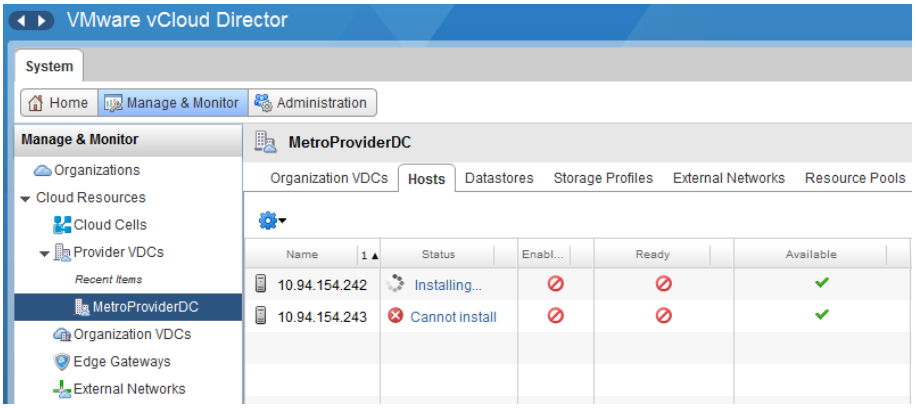
Tallennusprofiilin luomisen jälkeen tuotiin ympäristöön vCloud Director. Tämän yhteydessä annettiin myös oikeat verkkoasetukset virtuaalikoneelle jo ennen sen käynnistymistä. vCloud Directorin ohjeistuksen mukaisesti ja laajempien koekäyttöjärjestelmien rakentamiseen tähtäävien ohjeiden [39] vastaisesti vCloud Directorin verkkokortit kytkettiin samaan verkkoon verkon rakenteen vuoksi. Ennen vCloud Directorin yksityiskohtaista konfigurointia tuotiin ympäristöön kuitenkin vielä vShield Manager. Tämän tuomisen yhteydessä ei annettu verkon asetuksia vaan ne konfiguroitiin tuomisen jälkeen. vShield sijoitettiin samaan verkkoon vCloud Directorin kanssa. Tämän jälkeen päästiin käsiksi vShieldin web-selaimen kautta hallittavaan Flash-pohjaiseen hallintapaneeliin. Aluksi vShieldiin yhdistettiin vCenter-palvelin. Seuraavaksi tuotiin vCenterin varmenne järjestelmään ja tarkistettiin vShieldillä tuodun moduulin toiminta sekä Solutions and Applications -kategoriassa että datacenter- ja cluster-tasoille Hosts and Clusters -näkymässä. VXLAN-valmistelut aloitettiin datacenter-tason uudella verkon virtualisointia käsittelevällä välilehdellä, jossa alustettiin ryppään ominaisuuksia VXLAN-toiminnallisuutta ajatellen. Valmistelun varjolla hajautetulle virtuaaliselle kytkimelle luotiin vmk-virtuaaliverkkosovittimet, joille syötettiin osoiteasetukset ryppään hallintasovelluksen kautta. Tämän jälkeen Multicast- ja Segment ID -asetukset asetettiin asianmukaisiksi. vShieldin ylläpitämän verkkopuo-

len valmistuttua siirryttiin vCloud Directorin web-selainpohjaiseen käyttöliittymään, josta asennusprosessia jatkettiin. vCloud Directorin web-käyttöliittymässä aluksi syötettiin lisenssiavain ja järjestelmähallinnan kannalta erityiset tiedot järjestelmänvalvojan ja järjestelmän nimen osalta. Perusasetusten jälkeen järjestelmään kirjauduttiin aiemmin luoduilla järjestelmävalvojan tunnuksilla. Jatkokonfiguraatio koostui kahdesta pääkohdasta ja niiden seitsemästä alavaiheesta:

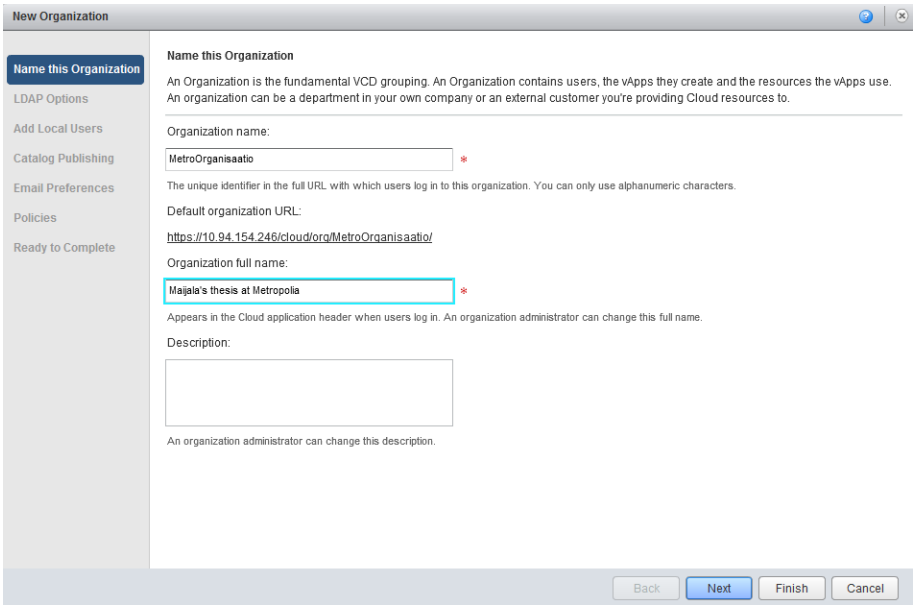
- Järjestelmän resurssien alustus
  - vCenterin yhdistäminen vShield Manageriin
  - Provider Virtual Datacentereiden luominen
  - Ulkoisen verkon osoitteistuksen määrittely
  - Virtuaalikoneille jaettavan verkkopoolin määrittely
- Provisioitujen etäresurssiresurssien organisaatiokohtainen allokointi
  - Organisaation luonti
  - Resurssien ositus organisaatiolle
  - Katalogin tuominen organisaatioon käytettäväksi

VMwaren koekäyttöohjeistuksen [39] mukaisesti palveluntarjoajan etäresurssidatacenterin kohdalla yhdistettäisiin kaksi rypästä yhdeksi, mikä jätettiin tässä kohtaa tekemättä. Liitettäessä vCenter vCloud Directorin palveluntarjoajan resurssiryppään hallintapalvelimeksi luotiin vCenteriin System vDC DRS -ryhmä vCloud Directorin ajettavalle kuormalle sekä otettiin käyttöön aiemmin luotu Storage Profile -levyprofiili. Ohjeistuksesta poiketen käytettäessä ainoastaan kahta ESXi-palvelinta täytyi ne tuoda palveluntarjoajan ryppääseen mukautetusti yksitellen pakettien asentamisen järjestelyn ja kuormantasauksen vuoksi. Tämä tapahtui peruen etäresurssiliitännäisten pakettien asennus molemmilta palvelimilta, siirtämällä kuorma täysin toiselle palvelimelle ja lisättiin jäljelle jäävä tyhjä palvelin kuvan 35 mukaisesti etäresurssiin. Tämän tapahduttua siirrettiin kaikki kuorma toiselle palvelimelle ja lisäämällä kuormattomaksi tehty palvelin etäresurssiin palvelimeksi.

Organisaation luomisen yhteydessä vCloud Director osoittaa organisaatiolle oman osoitetun alasivuston sivun 61 kuvan 36 mukaan emosivuston alta. Tätä kautta ei-ylläpidolliset hallinnolliset toimet ovat suoritettavissa ilman kokonaisvaltaisempaa käsitystä etäresurssipalvelujen edellyttämistä toiminnallisista kokonaisuuksista. Tämän yhteydessä myös



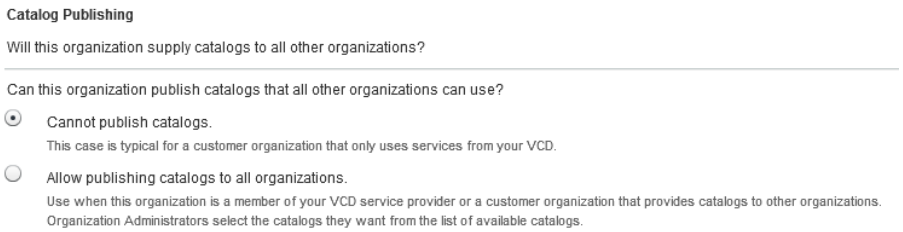
Kuva 35. Palvelimet on tuotava yksitellen etäresurssin käyttöön.



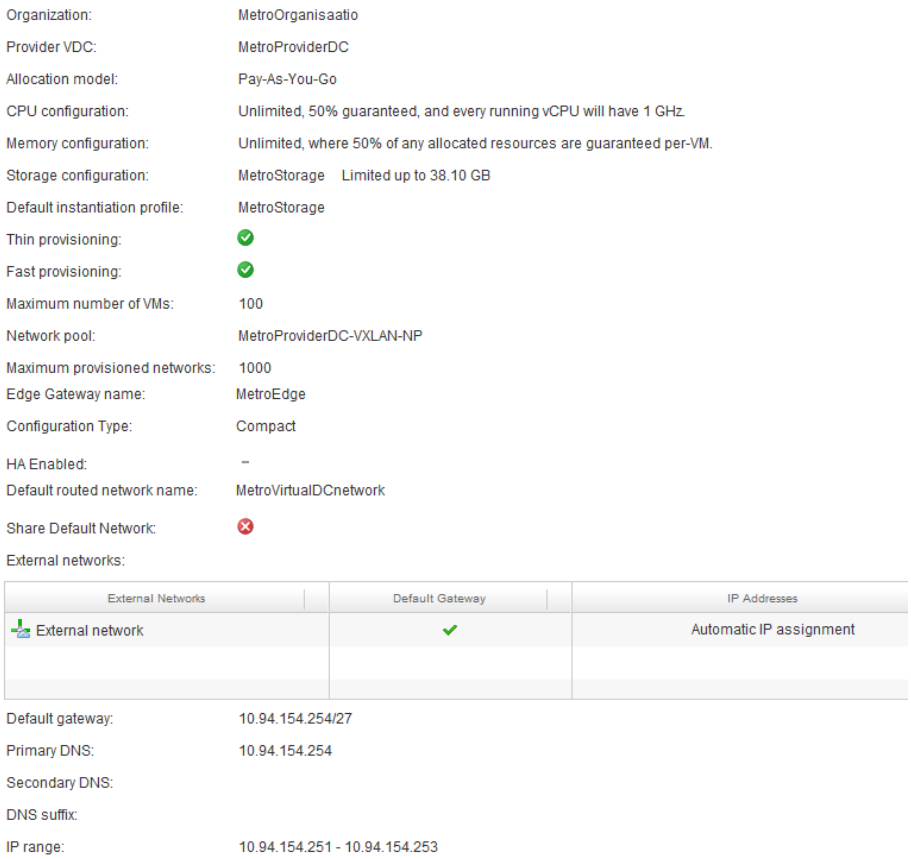
Kuva 36. Organisaation sivusto löytyy organosatorisen hallintakäyttöliittymän alasivustona.

luotiin organisaation oma ylläpitojärjestelmänvalvoja sekä luotavan organisaation ja muiden organisaatioiden väliset, kuvan 37 katalogijakosuhteet. Lisäksi tässä yhteydessä määriteltiin myös yleiset raja-arvot etäresurssikuorman käynnissäpysymiselle sekä katalogien päivitettävyydelle ja niiden kuluttamille laitteistoresursseille.

Organisaation luonnin jälkeen tuotiin organisaatiolle virtuaalinen palvelinsali. Virtuaalisen palvelinsalin resurssien kohdalta konfiguroidaan varustasot ja luvatut tasot niin prosessoriytimien, keskusmuistikapasiteetin kuin kiintolevykapasiteetin ja verkkoyhteyksien suhteen. Kiintolevytilan suhteen valittiin myös oletusarvoinen tallennusprofiili ja vShield Ed-



Kuva 37. Organisaatiolla on mahdollista olla yksityisiä tai jaettuja katalogeja.



Kuva 38. Virtuaalisen palvelinsalin ominaisuudet lopullisessa yhteenvetonäkymässä.

gelle osoitettiin ulkoisia IP-osoitteita käytettäväksi osoitteenmuunnoksissa ja palomuuritoiminnoissa. Lopussa vahvistettiin vielä tiedot yhteenvetonäkymästä, kuten kuva 38 kertoo. Yhteenvetonäkymässä on mahdoton erottaa virtuaalikoneille jaettavien julkisten IP-osoitteiden ja vShield Edgen osoitettujen IP-osoitteiden eroja.

5.3.5 Katalogin luominen ja käyttö ympäristöjen luomisessa.

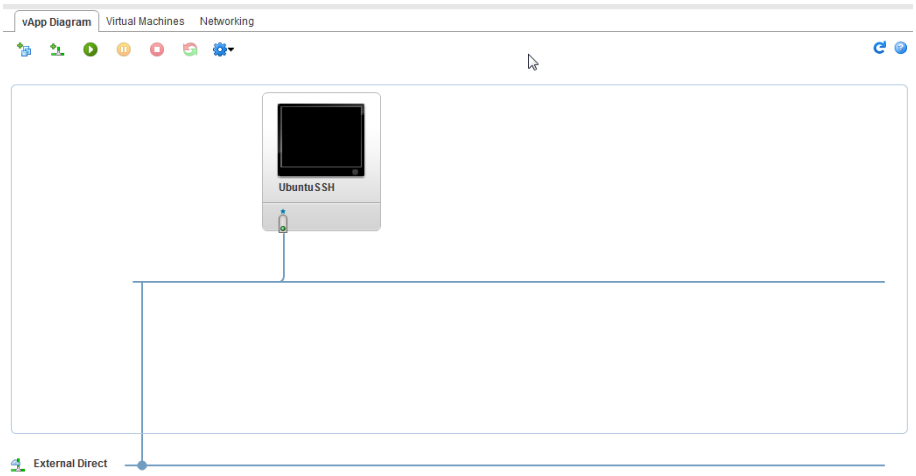
Seuraavaksi luotiin organisaatiolle katalogi, josta sen ylläpitäjät ja käyttäjät voivat tuoda sisältöä. Seuraavaksi siirryttiin katalogiin lisäämään sisältöä organisaation saataville. Levykuvien siirtäminen etäresurssiin tapahtui Javalla tehdyllä asiakasohjelmalla. Valmiiden virtuaalikoneiden tuominen vSphere-ympäristöstä käytti Flashia. Ympäristössä rakennettiin myös Ubuntun 12.04-versiollinen virtuaalikone. Tätä Ubuntu-pohjaista virtuaalikonetta on käytetty jokaisessa ympäristössä. Seuraavaksi siirryttiin My Cloud -näkymään. Tämä sama näkymä seuraa, kun organisaation käyttäjät haluavat itse käsitellä etäresurssipalvelun tarjoomaa siirryttäessä kuvan 36 osoitteen kautta. vApps-valinnan alla luotiin kolme erilaista vApp-virtualisointikokonaisuutta.

Rakennetut ympäristöt

Ensimmäinen ympäristö koostui suorasta siltauksesta ulkoiseen verkkoon, seuraava Edgen kautta osoitteenmuunnoksen kautta toteutettavan yhteyden avulla ja kolmas Edgen takana olleen toisen Edgen kautta tapahtuvan osoitteenmuunnoksen kautta.

Ympäristö 1

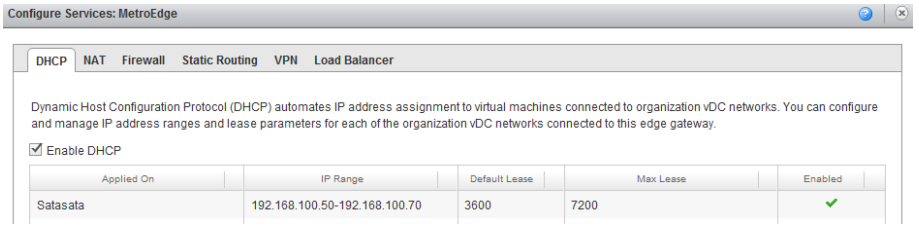
Ympäristö 1 luotiin rakentamalla vApp-kokonaisuus, joka sillattiin suoraan kuvan 39 ulkoiseen VM network -verkkoon, josta on suora yhteys VM network -verkkoon ilman Edgen osallisuutta.



Kuva 39. Ensimmäinen ympäristö sisälsi suoran verkkoyhteyden.

Ympäristö 2

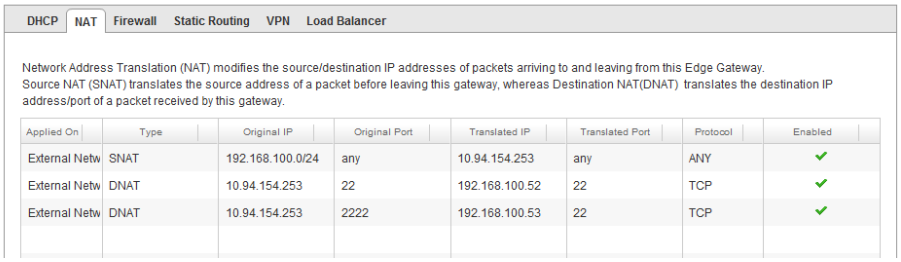
Ympäristö 2 luotiin lisäämällä kaksi virtuaalikonetta vApp-kokonaisuuteen ja luomalla niille oma sisäinen Satasata-niminen verkko. Virtuaalikoneet asetettiin käyttämään Edgen DHCP-palvelinta kuvan 40 asetuksin sekä MetroEdgelle luotiin säännöstö kuvan 41 mukaisen PAT-tyyppisen osoitteenmunnoksen mahdollistamiseksi. Säännöissä ohjattiin Edgen osoitteenmunnoksen suorittavan osoitteen portti 22 Alfalle ja portti 2222 Betalle. Säännösten luomisen yhteydessä voitiin havaita Edgen lukinneen ulkoisen IP-osoitteen itselleen sivulla 65 esitettävän kuvan 42 näkymästä.



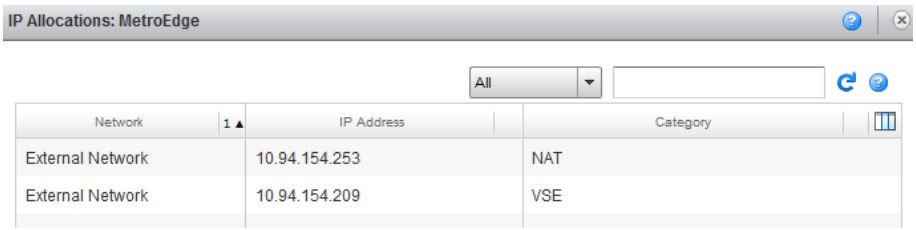
Kuva 40. MetroEdgen DHCP-osoiteavaruuden määrittely.

Ympäristö 3

Ympäristö 3 toteutettiin luomalla vApp-kokonaisuus siten, että luotiin uusi organisaation sisäinen verkko 2.1 fwd -nimellä siten, että sen liikenne kulki Ympäristö 2:n verkon ja



Kuva 41. Porttiosoitteenmuunnossäännöstöt Edgellä.



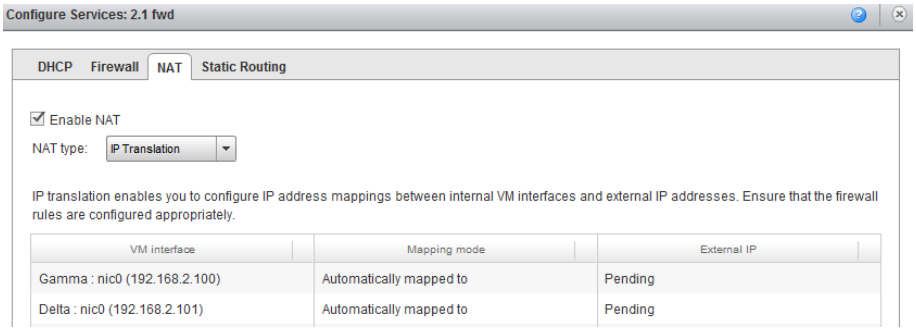
Kuva 42. Edge allokoi käyttöönsä ulkoisen IP-osoitteen osoitteenmuunnosta varten.

samalla Edgen osoitteenmuunnoksen kautta. Ympäristö 3 koostuu kahdesta virtuaalikoneesta ja niille luodusta sisäisestä verkosta, jolle asetettiin staattinen IP-osoiteavaruusvaranto kuvan 43 mukaisesti. Verkon Edgelle määriteltiin dynaaminen osoitteenmuunnos näille kahdelle virtuaalikoneelle kuvan 44 asetuksin. Ympäristö käynnistettäessä osoitteenmuutosten osoitteistus määräytyy ulkoisesti Satasata-verkon staattisten IP-osoitteiden varannon perusteella, kuten kuvasta 50 sivulta 70 voidaan nähdä varsinaisen toiminnallisuuden kartoituksen osiosta.

Kuva 43. 2.1 fwd -verkon osoiteasetukset.

Taulukko 5. Yhteenveto vApp-ympäristöjen ominaisuuksista.

Porttiryhmän nimi	Virtuaalikoneet	Osoiteavaruus	Edget	Osoittenmuunnos	Yhteys internettiin
VM network	UbuntuSSH	10.94.154.0/24	-	-	Suora yhteys
Satasata	Alfa ja Beta	192.168.100.0/24	MetroEdge	Porttimuunnos	MetroEdgen kautta
2.1 fwd	Gamma ja Delta	192.168.2.0/24	2.1 fwd ja MetroEdge	Dynaaminen	2.1 fwd ja Metroedgen kautta



Kuva 44. 2.1 fwd -verkon osoitteenmuunnosasetukset yhdestä yhteen.

5.4 Lopullisen ympäristön toimivuus ja ominaisuudet

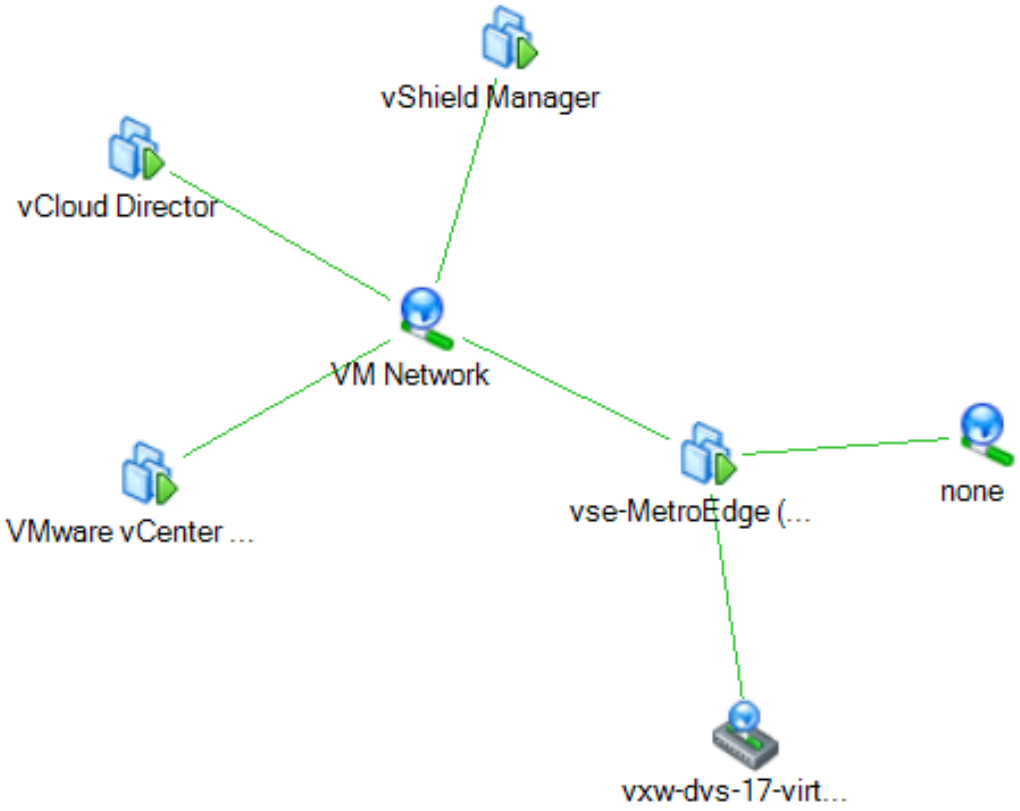
Kuvan 45 sisäkkäisesti virtualisoitu ympäristö virtuaalikoneineen näyttää loppuen lopuksi kuvan 46 kaltaiselta laitteisto mukaan lukien. Tähän ympäristöön rakennettiin kolme verkkokonfiguraatioita suoraan ulkoverkkoon sillatusti sekä kahdentyyppisellä osoitteenmuunnoksella varustetusti. Näistä ensimmäinen osoitteenmuunnos oli yhden suhde moneen -tyyppinen, jossa Edgen kaksi porttia uudelleenohjattiin edelleen Edgen takana oleville virtuaalikoneille. Kolmannessa verkkokonfiguraatiossa porttikohtaisen ohjaamisen sijaan ohjaus tapahtui osoitetasolla osoitteenmuunnoksen olleen tyyppiä yhden suhde yhteen, jossa Edgen takana olleille koneille osoitettiin Edgen julkiselta puolelta oma osoite.

Ympäristö 1

Ensimmäisessä verkkokonfiguraatiossa etäresurssissa ajaettava virtuaalikone sillattiin suoraan julkiseen verkkoon kuvan 47 mukaisesti. Verkkorakennenäkymässä ensimmäinen ympäristö nähdään kuvassa 48.

Ympäristö 2

Porttikohtaisen osoitteenmuutoksen ympäristö liittyi asianomaiseen hajautetun kytkimen porttiryhmään, joka puolestaan kytkeytyi Edgeen. Porttiohjaus toimi asianmukaisesti. Ul-

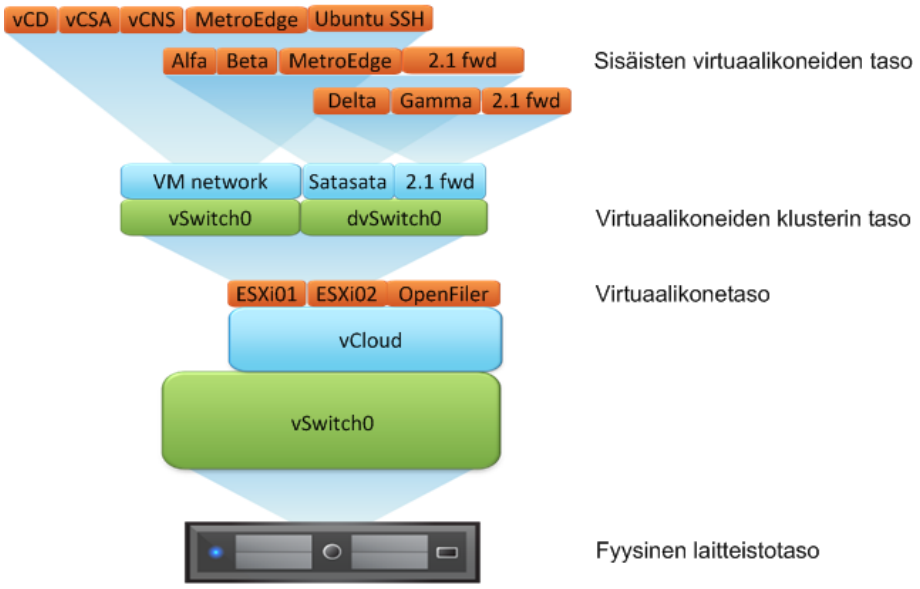


Kuva 45. Etäresurssiympäristö ainoastaan infrastruktuuria ylläpitävien virtuaalikoneiden kanssa toteutusvaiheen koejärjestelmässä.

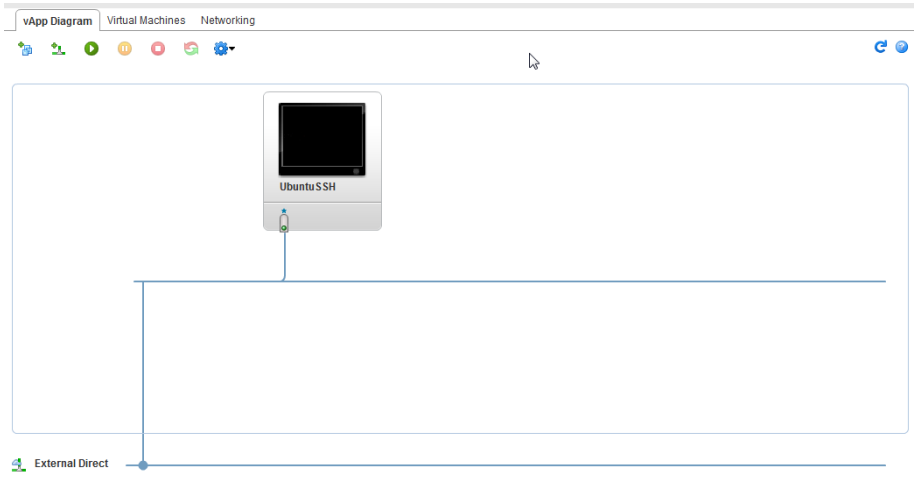
koverkosta tavoitellen Edgen osoitteenmuunnoksesta vastaavasta portista 22 vastasi Alfa ja portista 2222 saatiin yhteys Betaan porttiohjaukseen perustuvan osoitteenmuunnoksen avulla.

Ympäristö 3 - osoitekohtainen porttimuutos ympäristö 2 kautta

Lopuksi luotiin oma virtuaalikokoelma omalla sisäisellä verkollaan, joka käytti ympäristön 2 verkkoa ja oli sitä kautta yhteydessä ulkoisiin palveluihin. Oma paikallisen tason verkko päättyi Edgeen, joka yhdistyi organisaatiotasolla hajautetun kytkimen kautta toiseen aiemmin luotuun Edgeen. Ympäristö käynnistettäessä osoitteenmuutosten osoitteistus määräytyi ulkoisesti Satasata-verkon staattisten IP-osoitteiden varannon perusteella, kuten kuvasta 50 nähdään. Virtuaalikoneet Gamma ja Delta saivat osoitteensa omalta Edgel-

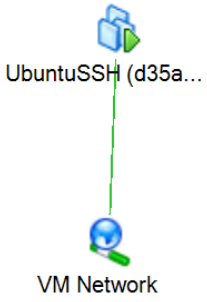


Kuva 46. Toteutusvaiheen koejärjestelmän verkkorakennekuvaus.

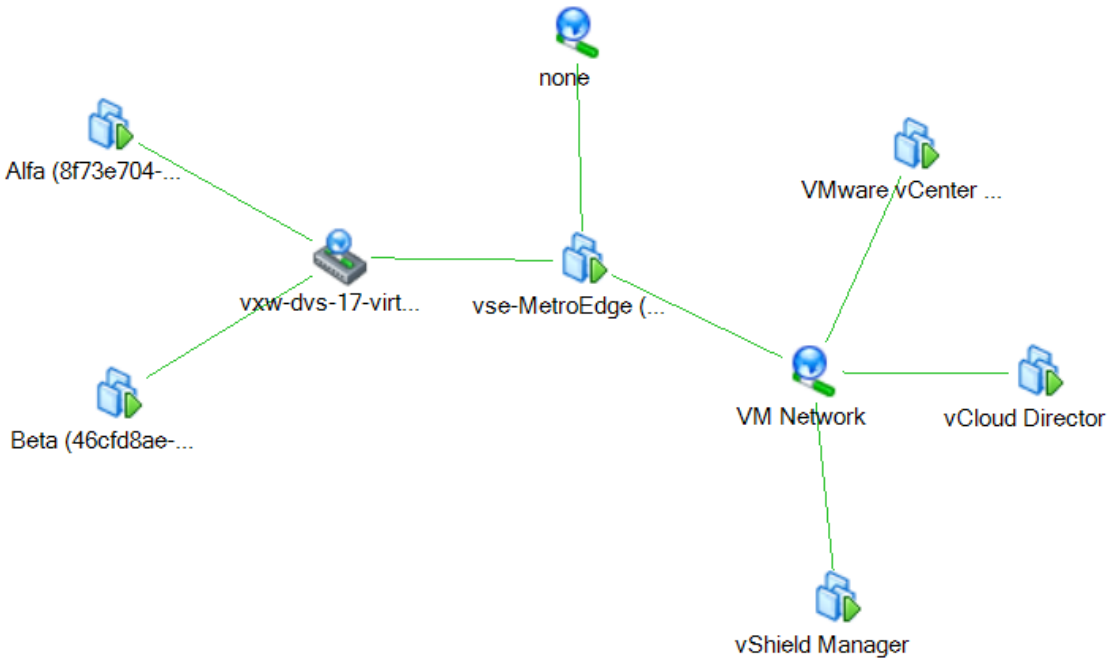


Kuva 47. Ensimmäinen ympäristö sisälsi suoran verkkoyhteyden.

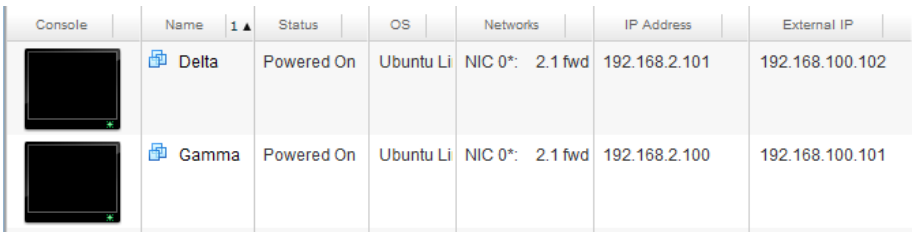
tään, ja niihin oli mahdollista ottaa yhteys Satasata-verkosta sivun 71 kuvan 51 tyyppisesti.



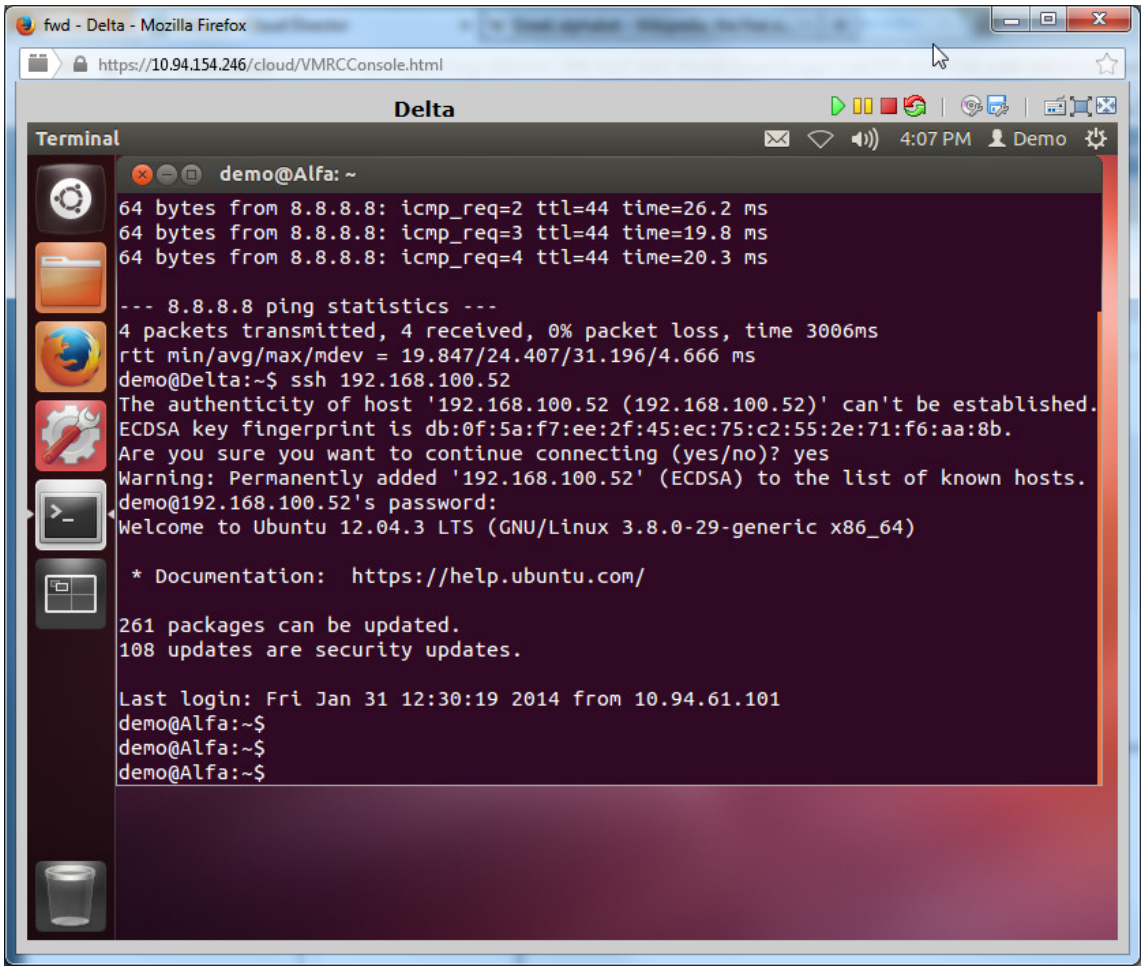
Kuva 48. Ympäristö 1 virtuaalikone suoraan sillattuna ulkoverkkoon ilman Edgeä.



Kuva 49. Porttiohjatun osoitteenmuutoksen takana olevat virtuaalikoneet.



Kuva 50. Dynaaminen osoitteenmuunnos allokoi saavutettavissa olevasta osoiteavaruudesta osoitteet virtuaalikoneille.



Kuva 51. Deltalta saadaan muodostettua SSH-yhteys Alfalle.

6 Yhteenveto

Insinöörityössä toteutettiin pilottivaiheen jälkeen IaaS-tasoinen etäresurssipalvelu käyttäen vCloud Suite -tuotetta sisäisesti virtualisoituna vSphere-ympäristössä Dell R710-palvelimella. Tähän ympäristöön tuotiin etäresurssipalvelun mahdollistavat virtuaalikoneet ja niiden avulla rakennettiin kolme erilaista ympäristöä. Ensimmäinen ympäristö koostui suorasta yhteydestä ulkoiseen VM network -tason verkkoon. Toinen käsitti osoitteenmuunnoksen Edgen kautta porttikohtaisesti sisäänpäin avauttuna. Kolmannessa ympäristössä toteutettiin toisen ympäristön kautta toimiva liikennöinti dynaamisen osoitteenmuunnoksen kautta.

Tulokset ja onnistumiset

Työn pilottivaiheessa onnistuttiin etäresurssin rakentamisessa työasematasolla sekä myöhemmin palvelimella sisäkkäisissä virtualisointikerroksissa. Pilottivaiheessa yhdistetty infrastruktuuri toteutettiin myös toimivasti varsinaisessa toteutusvaiheessa, jossa virtuaalisille virtuaalipalvelimiin liitettiin virtuaalikoneet. Kaikki kolme luotua ympäristöä havaittiin toimiviksi sekä saavutettaviksi. Haasteellisena voidaan myös pitää tapaa erotella IaaS-palvelussa ajettavien Edgen ja virtuaalikoneiden välisien osoitteiden saatavuuksia ja osoituksia. Lopulta tämä kuitenkin osoittautui täysin loogiseksi osoitteiden jäsentelyksi.

Ongelmat ja haasteet

vCloud Suite ollessa VMwaren tuotevalikoimassa jo vuodesta 2010 on valitettavaa havaita heikkouksia palomuurin osalta. Esimerkiksi useampaa sisäkkäistä Edgeä käyttävä kolmas ympäristö otti hyvinkin oman aikansa ja hieman enemmänkin käynnistyessään

ja konfiguroituessaan ilman ulkoisia merkkejä minkään tapahtumisesta. Osaltaan syynä voidaan pitää sisäkkäisiä virtualisointikerroksia niiden kuormittavuuden vuoksi.

Talousnäkökulmat

Työssä tarkasteltiin virtualisoinnin teknisen toteutuksen lisäksi etäresurssipalveluiden hinnoitterakenteita ja allokaatiomalleja, jotka ovat omiaan sekä tehostamaan palvelujen tuotantoa että laskemaan niiden ostamisen kustannuksia.

Kehityskohteet

Sisäisesti virtualisoituun IaaS-palveluun voitaisiin esimerkiksi lisätä kuormantasauspalvelimet. Lisäksi mahdollisuus siirtää järjestelmä lennosta oikealle sisäisesti virtualisoimattomalle laitteistolle ryppäiden ominaisuuksia muokkaamalla olisi mielenkiintoinen kehityssuunta.

Lähteet

1 VMware Secure Virtualization for Datacenter Security. 2014. Verkkodokumentti. VMware. http://www.vmware.com/products/datacenter-virtualization/vsphere/scale-security.html Luettu 15.1.2014.

2 William Lam. How to Enable Nested ESXi & Other Hypervisors in vSphere 5.1. 29.8.2012. Verkkodokumentti. Virtuallyghetto http://www.virtuallyghetto.com/2012/08/how-to-enable-nested-esxi-other.html Luettu 2.6.2014.

3 Software and Hardware Techniques for x86 Virtualization. Luotu 8.6.2006. Päivitetty 9.8.2011. Verkkodokumentti. VMware. http://www.vmware.com/files/pdf/software_hardware_tech_x86_virt.pdf. Luettu 27.5.2014.

4 Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau. Operating Systems: Three Easy Pieces - Virtual Machine Monitors 101. 26.5.2014. Verkkodokumentti. University of Wisconsin-Madison. http://pages.cs.wisc.edu/~remzi/OSTEP/vmm-intro.pdf Luettu 30.5.2014.

5 Haltsonen Seppo, Esko T. Rautanen. Tietokonetekniikka. Edita. 2008. sivut 134-148.

6 Virtualization Technique System Virtualization Memory Virtualization. Michigan Technological University. 10.8.2010. Verkkodokumentti. National Tsing Hua University. http://www.cs.nthu.edu.tw/~ychung/slides/Virtualization/VM-Lecture-2-2-SystemVirtualizationMemory.pptx Luettu 30.5.2014.

7 The Basics of Page Faults. 10.6.2008. Verkkodokumentti. Microsoft. http://blogs.technet.com/b/askperf/archive/2008/06/10/the-basics-of-page-faults.aspx Luettu 2.6.2014.

8 Scott Devine. E6998 - Virtual Machines Lecture 3 Memory Virtualization. 2013. Verkkodokumentti. VMware. http://www.cs.columbia.edu/~nieh/teaching/e6998_s08/ Luettu 2.6.2014.

9 vSphere Availability ESXi 5.1 vCenter Server 5.1. Luotu 11.7.2012. Muokattu 23.7.2012. Verkkodokumentti. VMware. http://pubs.vmware.com/vsphere-51/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-51-availability-guide.pdf Luettu 9.6.2014.

10 Virtualizing Business-Critical Applications on VMware vSphere. 26.1.2010. Verkkodokumentti. VMware. http://www.vmware.com/files/pdf/VMW_10Q1_

WP_vSPHERE_USLET_EN_R6_proof.pdf Luettu 11.2.2014.

11 How Virtualization is Key to Managing Risk. 2013. Verkkodokumentti. Virtualizationreview. http://virtualizationreview.com/whitepapers/2013/06/vmware-how-virtualization-is-key-to-managing-risk/asset.aspx Luettu 13.6.2013.

12 Metrics and Measurements, White Paper #49-PUE. 02.10.2012. Verkkodokumentti. The Green Grid. http://www.thegreengrid.org/en/Global/Content/white-papers/WP49-PUEAComprehensiveExaminationoftheMetric Luettu 17.2.2014.

13 Case Study #5 - Chiller System Optimization. 13.12.2013. Verkkodokumentti. The Green Grid. http://www.thegreengrid.org/Global/Content/case-studies/CS5-Chiller_System_Optimization Luettu 17.2.2014.

14 vSphere vMotion: Live Migration of Virtual Machines. 2014. Verkkodokumentti. VMware. http://www.vmware.com/products/datacenter-virtualization/vsphere/vmotion.html Luettu 17.2.2014.

15 Host Power Management in VMware vSphere 5. Luotu 10.8.2011. Muokattu 12.8.2011. Verkkodokumentti. VMware. https://www.vmware.com/files/pdf/hpm-perf-vsphere5.pdf Luettu 25.5.2013.

16 ACPI (Advanced Configuration and Power Interface) 5.0. HP, Intel, Microsoft, Phoenix Technologies, Toshiba. 6.12.2011. Verkkodokumentti. acpi.info http://acpi.info/spec50.htm Luettu 17.2.2014.

17 Rebecca Grider. Power Management and Performance in VMware vSphere® 5.1 and 5.5 Performance Study TECHNICAL WHITE PAPER. 18.3.2014. Verkkodokumentti. VMware. http://blogs.vmware.com/performance/2014/03/power-management-performance-vmware-vsphere-5-1-5-5.html Luettu 17.5.2014.

18 Open Virtualization Format Specification. 12.1.2010. Verkkodokumentti. Distributed Management Task Force. http://dmtf.org/sites/default/files/standards/documents/DSP0243_1.1.0.pdf Luettu 17.2.2014.

19 DMTF Accepts New Format for Portable Virtual Machines from Virtualization Leaders. 10.9.2007. Verkkodokumentti. Distributed Management Task Force. http://www.dmtf.org/news/pr/2007/9/dmtf-accepts-new-format-portable-virtual-machines-virtualization-leaders Luettu 17.2.2014.

20 Aidan Finn. Beware of Windows Server and System Center Update Rollups. 2012. Verkkodokumentti. Aidan Finn http://www.aidanfinn.com/?p=15444 Luettu 17.2.2014.

21 Matthias Luft. From Hypervisors to Clouds or How Traditional Security Controls Fail. 21.10.2012. Verkkodokumentti. ERNW. https://www.ernw.de/download/ERNW_DCVI-HypervisorsToClouds.pdf Luettu 17.06.2013.

22 Crisis virus attempts to infect virtual machines running on VMware Workstation or Player using legitimate functionality (2033939). Luotu 20.8.2012. Päivitetty 29.7.2014. Verkkodokumentti. VMware. http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2033939 Luettu 17.2.2014.

23 Kostya Kortchinsky. A VMware Gues to Host Escape Story, Black Hat USA 2009. 2009. Verkkodokumentti. Black Hat. http://www.blackhat.com/presentations/bh-usa-09/KORTCHINSKY/BHUSA09-Kortchinsky-Cloudburst-SLIDES.pdf Luettu 10.12.2013.

24 Nelson Elhage. Virtunoid: A KVM Guest! Host privilege Escalation exploit, Black Hat USA 2011. 2011. Verkkodokumentti. Black Hat. https://media.blackhat.com/bh-us-11/Elhage/BH_US_11_Elhage_Virtunoid_WP.pdf Luettu 10.12.2013.

25 Arrigo Triulzi. Project Maux Mk.II, “I 0wn the NIC, now I want a shell!”. 2008. Verkkodokumentti. Alchemist Owl. http://www.alchemistowl.org/arrigo/Papers/Arrigo-Triulzi-PACSEC08-Project-Maux-II.pdf Luettu 10.12.2013.

26 VMware vSphere 5.5 Documentation Center, How Fault Tolerance Works. 2014. Verkkodokumentti. VMware. http://pubs.vmware.com/vsphere-55/index.jsp#com.vmware.vsphere.avail.doc/GUID-623812E6-D253-4FBC-B3E1-6FBFDF82ED21.html Luettu 9.6.2014.

27 Peter Mell and Timothy Grance. Special Publication 800-145, The NIST Definition of Cloud Computing, Recommendations of the National Institute of Standards and Technology. Luotu 2011. Muokattu 27.4.2012. Verkkodokumentti. National Institute of Standards and Technology. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf Luettu 7.8.2014.

28 Russ Walsh and Navarasu Dhanasekar. Cloud Computing - Technical & Business Aspects. 10.1.2013. Verkkodokumentti. San Jose State University. http://www.cob.sjsu.edu/nellen_a/CloudComputing1-10-13_RW_ND.pdf Luettu 5.6.2014.

29 VMware vCloud Director Resource Allocation Models. 27.8.2012. Verkkodokumentti. VMware. http://www.vmware.com/files/pdf/techpaper/vCloud_Director_Resource_Allocation-USLET.pdf Luettu 12.8.2013.

30 VMware vCloud Architecting a vCloud, version 1.6 Technical Whitepaper. 2010. Verkkodokumentti. VMware. https://www.vmware.com/files/pdf/VMware-Architecting-vCloud-WP.pdf Luettu 21.5.2013.

31 VMware vCloud Implementation Example Public vCloud Service Provider. 3.11.2010. Verkkodokumentti. VMware. https://www.vmware.com/files/pdf/VMware-vCloud-Implementation-Example-ServiceProvider.pdf Luettu 26.4.2013.

32 VMware vCloud® Architecture Toolkit, Architecting a VMware vCloud, Version 2.0.1. 11.11.2011. Verkkodokumentti. VMware. http://www.vmware.com/files/pdf/vcat/Architecting-VMware-vCloud.pdf Luettu 25.6.2013.

33 M. Mahalingam, D. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, C. Wright. Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks. Verkkodokumentti. rfc-editor.org. http://www.rfc-editor.org/rfc/rfc7348.txt Luettu 26.8.2014.

34 VMware vCloud Networking Poster. 2012. Verkkodokumentti. VMware. http://blogs.vmware.com/vsphere/2012/09/vmware-vcloud-networking-poster.html Luettu 24.4.2013.

35 Rawlinson Rivera. PunchingClouds, vCloud Director 5.1 VXLAN configuration. 9.9.2012. Verkkodokumentti. Punching Clouds. http://www.punchingclouds.com/2012/09/09/vcloud-director-5-1-vxlan-configuration/ Luettu 24.3.2013.

36 Valtteri Maijala. Installing and basic configuration to ESXi 4.1 and FreeNAS 8.0 (part1). 14.10.2011. Verkkodokumentti. Wordpress. https://virtualloop.wordpress.com/2011/10/14/installing-and-basic-configuration-to-esxi-4-1-and-freenas-8-0-part1/ Luettu 24.3.2013.

37 Chang Captain SK. PHYSICAL STRUCTURES. 2002. Verkkodokumentti. University of Pittsburgh. http://people.cs.pitt.edu/~chang/156/08struct.html Luettu 17.5.2014.

38 Bob Goldsand. SAP HANA Now Supported on VMware vSphere 5.5 for Production Scenarios. 6.5.2014. Verkkodokumentti. VMware. http://blogs.vmware.com/vsphere/2014/05/sap-hana-now-supported-vmware-vsphere-5-5-production.html Luettu 17.5.2014.

39 VMware vCloud Director 5.1 Evaluation Guide, Technical Whitepaper, v 1.0. Päivitetty lokakuu 2012. Verkkodokumentti. VMware. http://www.vmware.com/files/pdf/products/vCloud/VMware-vCloud-Director-51-Evaluation-Guide.pdf Luettu 2.12.2012.

40 ESXi Configuration Guide. 28.6.2012. Verkkodokumentti. VMware. http://www.vmware.com/pdf/vsphere4/r41/vsp_41_esxi_server_config.pdf Luettu 17.5.2014.