

FOSS -työasemavirtualisointi ja etä- käyttö

Teemu Pölkki

Opinnäytetyö

Marraskuu 2020

Liiketalouden ala

Tradenomi (AMK), tietojenkäsittelyn tutkinto-ohjelma

Tekijä(t) Pölkki, Teemu	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Marraskuu 2020
	Sivumäärä 47	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi FOSS -työasemavirtualisointi ja etäkäyttö		
Tutkinto-ohjelma Tietojenkäsittelyn tutkinto-ohjelma		
Työn ohjaaja(t) Jarkko Immonen		
Toimeksiantaja(t) -		
Tiivistelmä <p>Etätyöasemavirtualisointi on tärkeä osa etätyöskentelyä ja modernia ICT-infrastruktuuria. Etätyöasemavirtualisoinnin avulla yrityksen työntekijät voivat työskennellä päätelaitteella kuin päätelaitteella käyttäen kuitenkin koko ajan samaa työasemaa.</p> <p>Tutkimuksessa tarkasteltiin virtualisointia, sen historiaa ja käytännön toimintaa virtuaalityöasemainfrastruktuurin näkökulmasta. Normaalisissa työasemakäytössä olevat tietokoneet ovat kalliita uusia, ja niiden elinkaari on lyhyempi kuin kevyiden asiakaspäätteiden elinkaari. Virtuaalityöasemainfrastruktuuri voi olla vastaus ongelmaan.</p> <p>Tutkimuksella oli epävirallinen nimettömänä pysyttelevä toimeksiantaja, jolla oli halu ja tarve virtualisoida käytössä olevia työasemia kustannusten säästämiseksi ja etätyön mahdollistamiseksi. Tämän tavoitteen täyttämiseksi tutkittiin virtualisointia, virtuaalityöasemainfrastruktuuria ja siihen liittyviä teknologioita. Tietoperustan kartuttamiseksi materiaalia kerättiin virtualisoinnin hyödyistä, haasteista sekä haitoista.</p> <p>Ennen nykytila-analyysia etsittiin sopivimpia vaihtoehtoja, joista valittiin kolme valmista olemassa olevaa ratkaisua. Ratkaisuvaihtoehdoista etsittiin tietoa, niiden toimintaa tutkittiin sekä analysoitiin, mutta ne todettiin puutteellisiksi. Tämän takia päätettiin kehittää prototyyppi uudesta ratkaisuvaihtoehdosta.</p> <p>Kehitystutkimuksen tuloksena syntyi prototyyppi virtuaalityöasemainfrastruktuurin osasta. Prototyyppillä pystyy käyttämään virtualisoituja työasemia etänä lähiverkon ylitse. Tuloksista hyötyvät henkilöt ja organisaatiot, jotka pohtivat virtualisointia osana ICT-infrastruktuuriaan sekä saman aihepiirin opinnäytetyöntekijät.</p>		
Avainsanat (asiasanat) Virtualisointi, Linux, Hypervisor, QEMU, KVM		
Muut tiedot		

Description

Author(s) Pölkki, Teemu	Type of publication Bachelor's thesis	Date November 2020 Language of publication: Finnish
	Number of pages 47	Permission for web publication: x
Title of publication FOSS desktop virtualization and remote use		
Degree programme Bachelor's degree programme in Business Information Technology		
Supervisor(s) Immonen, Jarkko		
Assigned by -		
Abstract <p>Workstation virtualization is an important part of telecommuting and modern ICT infrastructure. With the help of workstation virtualization, the organizations personnel can work on any client device they want, but always use the same workstation.</p> <p>The study examined virtualization, its history, and practical operation from the perspective of virtual workstation infrastructure. Regular computers used as workstations are expensive new and have a shorter life cycle than thin clients.</p> <p>The study had an anonymous client with a desire and need to virtualize existing workstations to save costs and enable telecommuting. To meet this goal, virtualization and related technologies were researched. To increase the knowledge base, material was collected on the advantages, challenges, and disadvantages of virtualization.</p> <p>Prior to the current state analysis, the most suitable alternatives were sought, from which three ready-made existing solutions were selected. Information was gathered from these three solutions, their operation was researched and analysed but the solutions did not fit the use case of the anonymous client. For this reason, it was decided to develop a prototype of a new solution.</p> <p>As a result of the research, a prototype was created. The prototype is a small part of virtual desktop infrastructure and it can be used to access virtualized workstations remotely over a local area network. The results will benefit individuals and organizations considering virtualization as part of their ICT infrastructure, as well as students working on their thesis around the same topics.</p>		
Keywords/tags (subjects) Virtualization, Linux, Hypervisor, QEMU, KVM		
Miscellaneous		

Sisältö

1	Johdanto	8
2	Tutkimusasetelma	9
2.1	Toimeksiantaja	9
2.2	Kehitystyön tavoitteet ja rajaukset	10
2.3	Tutkimuskysymykset- ja menetelmät	12
2.4	Kehitystutkimuksen käytännön toteutus	13
3	Virtualisointi	13
3.1	Virtualisoinnin määritelmä	14
3.2	Virtualisoinnin hyödyt	16
3.3	Virtualisoinnin haasteet ja haitat	18
3.4	Virtuaalinen työasemainfrastruktuuri	19
3.5	Virtuaalisen työasemainfrastruktuurin osat	21
4	Etätyöasemavirtualisoinnin toteutus	22
4.1	Alustava selvitystyö	23
4.2	Prototyypin kehitys	24
4.2.1	Isäntäkoneen asentaminen ja konfigurointi	25
4.2.2	Asiakaskoneen asentaminen ja konfigurointi	34
4.2.3	Virtualisoituun etätyöasemaan yhdistäminen	35
4.3	Prototyypin toiminnan validointi	37
5	Pohdinta	41
5.1	Kehitystyön tulokset ja johtopäätökset	41
5.2	Luotettavuus	42
	Lähteet	44

Kuviot

Kuvio 1. Etätyöläisten määrä koronaviruspandemian aikana	10
--	----

Kuvio 2. Esimerkkidiagrammi virtualisoinnin arkkitehtuurista.....	15
Kuvio 3. Hypervisorin sijainti virtualisoinnissa	16
Kuvio 4. Eri työasematyyppien hyviä ja huonoja puolia.....	21
Kuvio 5. Yksinkertaistettu VDI arkkitehtuuri	22
Kuvio 6. Rufus ohjelmisto konfiguroituna	26
Kuvio 7. Ubuntu Server 20.10 asennus, OpenSSH valittuna	27
Kuvio 8. Ubuntu server 20.10 asennettuna, sisäänkirjautumista vailla	27
Kuvio 9. virt-managerin, libvirtin ja Qemu/KVM:n suhteet toisiinsa	28
Kuvio 10. Virtualisointi löytyy, ja KVM kiihdytystä voi käyttää	29
Kuvio 11. libvirtd toiminnassa	30
Kuvio 12. Weston asennettuna	30
Kuvio 13. Waypipen arkkitehtuuri.....	31
Kuvio 14. Waypipen lähdekoodin hakeminen.....	32
Kuvio 15. Waypipen asennus onnistui.....	33
Kuvio 16. Virtuaalikoneessa ajettavan käyttöjärjestelmä asennusohjelmisto käynnistymässä.....	34
Kuvio 17. Prototyypin yksinkertaistettu arkkitehtuuri	37
Kuvio 18. Isäntäkoneen resurssien käyttö, kun vieraskoneella ei tehdä mitään .	38
Kuvio 19. Asiakaskoneen resurssien käyttö, kun vieraskoneella ei tehdä mitään	38
Kuvio 20. Verkonkäyttö, kun vieraskoneella ei tehdä mitään.....	39
Kuvio 21. Isäntäkoneen resurssien käyttö videokatselun aikana.....	39
Kuvio 22. Asiakaskoneen resurssien käyttö videokatselun aikana	40
Kuvio 23. Verkon käyttö, kun YouTuben videosoitin ei ole koko ruudulla.....	41

Käsitteet ja lyhenteet

ARM

Advanced RISC Machines. Prosessorin käskykanta, jota käytetään paljon mobiililaitteissa ja kevyissä tietokoneissa

BIOS

Lyhenne sanoista Basic Input/Output System. Ohjelma tietokoneen emolevyllä, lataa käyttöjärjestelmän käynnistyksenlataajan keskusmuistiin ja käynnistää sen.

Guest/vieras

Virtualisoitu tietokone tai palvelin.

Host/isäntä

Palvelin tai tietokone, jolla vieraskonetta suoritetaan

IP-osoite

Internetin protokollaosoite. Numerosarja, (esimerkiksi 127.0.0.1) jota käytetään tunnistamaan verkkosovitin tietoverkossa.

Kernel

Käyttöjärjestelmän ydin. Toimii tulkkina käyttöjärjestelmän ja laitteiston välillä.

KVM

Kernel-based Virtual Machine. Hypervisor -ohjelmisto, joka on rakennettu sisään Linux -kerneliin.

UEFI

Lyhenne sanoista Unified Extensible Firmware Interface. Toimii samanlaisessa roolissa kuin BIOS, ja on sen korvaaja

USB

Universal Serial Bus. Protokolla oheislaitteiden liittämiseksi sitä tukeviin laitteisiin.

VDI

Lyhenne termistä Virtual Desktop Infrastructure, virtualisoitu työasemainfrastruktuuri. Virtualisointistrategia, jossa suurin osa työasemista on virtualisoitu ja saatavilla etätyöpöytäyhteyden avulla

x86

x86 ja sen laajennukset (mm. x86_64) ovat prosessorin käskykantoja. Käytössä pääosin työasematietokoneissa, ja pelikonsoleissa.

1 Johdanto

ICT-tekniikan kehittyessä, pienentyessä ja halventuessa on tarpeen miettiä mahdollisuuksia kehittää olemassa olevaa tietojärjestelmäinfrastruktuuria. Tässä opinnäytetyössä palataan taaksepäin ajassa, jolloin suurtietokoneet ja kevyet asiakaspäätteet olivat uusinta uutta. Tutkimustyön ajankohta on hedelmällinen useista syistä. AMD on julkaissut uudet EPYC-palvelinprosessorit, ja yhden prosessoriytimen hinta laski huomattavasti. Tämä vaikuttaa huomattavasti "suurtietokoneen" hintaan, ja mahdollistaa jopa usean kymmenen vieraskoneen virtualisoinnin.

Toinen tutkimuksen mahdollistava ajankohtainen asia on Raspberry Pi Foundationin julkaisema ja kehittämä Raspberry Pi 4. Raspberry Pi 4 on yhden piirilevyn tietokone, joka on halpa, tehokas sekä virrankulutukseltaan konservatiivinen.

Kolmantena tutkimuksen ajankohtaisuuteen vaikuttavana asiana on nopeiden tietoverkkojen ja verkkolaitteiden hintojen lasku sekä yleistymisen. Graafisen käyttöliittymän käyttäminen tietoverkon yli vaatii paljon kaistaa. Nopeat gigabitin yhteydet mahdollistavat usean graafisen käyttöliittymän käyttämisen verkon yli.

Vaikka virtuaaliseen työasemainfrastruktuuriin perustuvia ratkaisuja on ollut olemassa jo kauan, ovat ne yleensä suljettua lähdekoodia ja kalliita käyttää. Kehitystutkimuksen tuloksena syntyvä prototyyppi toivottavasti avaa portteja ilmaisille vaihtoehdoille ja jatkokehitykselle asian saralla.

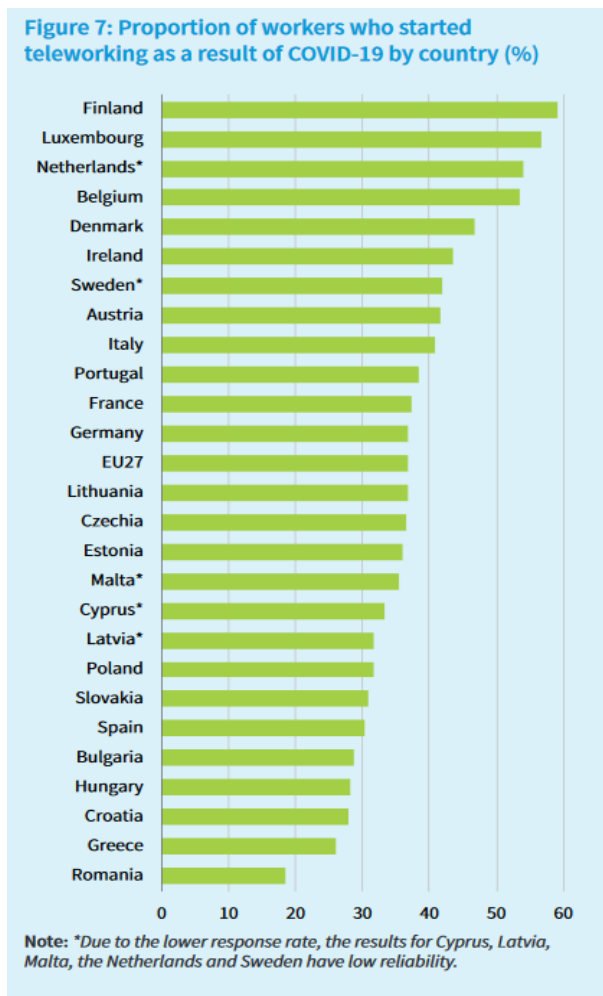
2 Tutkimusasetelma

Tässä luvussa kuvataan tutkimuksen lähtöasettelun keskeiset asiat, kuten tutkimuksen tavoitteet, rajaukset, taustat, sekä työkalut. Luvussa esitellään myös tutkimuksen tutkimuskysymykset sekä annetaan lukijalle selkeä kuva kehitys- ja tutkimusprosessista.

2.1 Toimeksiantaja

Työ tehdään nimettömänä pysyttelevän toimeksiantajan epävirallisesta pyynnöstä. Toimeksiantajalla on tarve keventää IT-infrastruktuuria sekä helpottaa etätyön tekemistä. Suomessa etätyötä tekevien määrä kasvoi vuosina 2013–2018 12 prosenttiyksikköä (Sutela, 2020). Sutela (2020) viittaa blogikirjoituksessaan Tilastokeskuksen (2018) tutkimukseen, jonka mukaan vuonna 2018 28 prosenttia palkansaajista tekee etätyötä, ja 16 prosentilla olisi ollut mahdollista tehdä etätyötä, mutta he eivät syystä tai toisesta sitä tehneet.

Etätyön helppo mahdollistaminen muodostui ajankohtaiseksi keväällä 2020 maailmaa, mukaan lukien Suomea, riepotelleen COVID-19 pandemian takia. Pandemian takia noin 60 % Suomen työntekijöistä alkoi etätyöskentelemään (Living, working and COVID-19, 2019)



Kuvio 1. Etätyöläisten määrä koronaviruspandemian aikana (Living, working and COVID-19, 2019)

Kehitystutkimusta tehdään myös allekirjoittaneen omien yrityshaaveiden takia. Muodostuvasta prototyypistä olisi tarkoitus jatkokehittää opinnäytetyön valmistumisen jälkeen oikea toimiva myytävä tuote tai palvelu, jolla voi tehdä varteenotettavaa liiketoimintaa.

2.2 Kehitystyön tavoitteet ja rajaukset

Tutkimuksen aiheeksi on rajattu laajasta kokonaisuudesta kapeahko kehitystyö, jonka pääaiheena on selvittää kuinka modernilla IT-laitteistolla, voidaan rakentaa vanhaa suurtietokone-kevyt asiakastietokone -arkkitehtuuria jäljittelevä moderni tietojärjestelmä ja kehittää siitä toimiva prototyyppi. Joitain toimivia ratkaisuja on jo

olemassa, mutta ne eivät sopineet kriteereiltään nimettömänä pysyvän toimeksiantajan käyttöön kalliin hinnan ja suljetun lähdekoodin takia. Nimettömänä pysyttelevä toimeksiantaja toivoo myös kovasti, että järjestelmä rakentuisi kokonaan vapaista ohjelmistoista.

Richard Stallmanin Free Software Foundation määrittelee vapaan ohjelmiston tarkoittavan, että käyttäjällä on vapaus käyttää, kopioida, jakaa, opiskella, muuttaa ja parantaa ohjelmistoa. Vapaa ohjelmisto ei siis tarkoita ilmaista ohjelmistoa, vaikka vapaat ohjelmistot ovat ilmaisia käyttää. (What is free software? 2019).

Esimerkkeinä toimivasta ratkaisusta on VMWare Horizon ja Windows Virtual Desktops. VMWaren ratkaisu mahdollista hankkia virtuaalisena työasemainfrastruktuurina (VDI) tai työasema palveluna (DaaS) mallilla (VMWare Horizon N.d.). Windows Virtual Desktops on puhdas työasema palveluna ratkaisu. (Windows Virtual Desktop, N.d.)

Asiakaskoneeksi kehitystutkimuksessa rajattiin Raspberry Pi 4. Valinta tehtiin, koska Raspberry Pi 4 on Piltchin (2020) arvostelun mukaan yksi parhaista yhden piirilevyn tietokoneista ja paras omassa hintaluokassaan. Raspberry Pi 4 tukee 3940x2160 (4K) resoluutiota 60Hz virkistystaajudella yhdellä näytöllä ja 30Hz virkistystaajudella kahdella näytöllä. 1920x1080 (FullHD) resoluutiolla taasen kahta 60Hz virkistystaajudella operoivaa näyttöä. Myös 1920x1080 videotoisto on täysin sulavaa. (Piltch, 2020). Laitteesta löytyy myös tuki gigabitin langalliselle yhteydelle, ja se operoi Pilchin (2020) mukaan maksimissaan 943Mbit/s nopeudella. Hyvä, nopea videotoisto sekä nopeat verkkoyhteydet ovat tärkeitä ominaisuus etätyöasemavirtualisointia ajatellessa.

Opinnäytetyön kehitysosassa kehitetään järjestelmä. Kehitystyössä tärkeä osa on selvittää mitä ohjelmistoja, konfiguraatioita sekä laitteita työasemavirtualisoinnin sulavasti toteuttava tietojärjestelmä vaatii toimiakseen. Työssä ei keskitytä tutkimaan matalan tason protokollia tai ohjelmointirajapintoihin vaan toimivan ratkaisun rakentamiseen ja tähän vaadittaviin toimenpiteisiin.

Kehittämistutkimuksen perimmäisenä tavoitteena on tutkia mahdollisuuksia siirtää

yrittäjien työasemat virtuaalisiksi ja etäkäytettäväksi. Ratkaisun olisi oltava kustannustehokas, toimiva sekä tietoturvallinen. Tavoitteena ei kuitenkaan ole rakentaa koko IT-järjestelmää, vaan toimiva prototyyppi, jonka pohjalta järjestelmää voidaan jatkokehittää ja mahdollisesti tulevaisuudessa kaupallistaa.

Teoriaosassa käsitellään virtualisointia kokonaisuutena, painottaen kuitenkin työasemavirtualisointia sekä sen eri osa alueita. Teoriaa on virtualisoinnin historiasta, hyvistä ja huonoista puolista sekä sen toimintaperiaatteista.

2.3 Tutkimuskysymykset- ja menetelmät

Opinnäytetyössä haetaan vastauksia seuraaviin kysymyksiin:

1. Mitä virtualisointi tarkoittaa?
2. Mitä eri osia työasemavirtualisoinnissa on?
3. Millainen ratkaisuvaihtoehto ratkaisee yrityksen X ongelman?

Kahteen ensimmäiseen kysymykseen vastataan teoreettisessa viitekehityksessä ja kolmanteen tutkimusosassa.

Kanasen (2015, 47) mukaan kehittämistutkimus alkaa kuten kaikki muukin tutkimustyö. Perehtymällä tutkittavaan ilmiöön, ja jo aiemmin ilmiöstä kirjoitettuun materiaaliin. Kirjallisuuskatsaus kuuluu Kanasen (mts, 47) mukaan jokaiseen tieteellisen opinnäytetyöhön. Tietoa voi kerätä kirjallisuudesta, ja nykypäivänä enenevässä määrin internetistä. Teoriatieto voi olla tietokirjallisuutta, aihetta sivuavia raportteja, tutkimuksia malleja tai teorioita. (Mts, 47).

Tässä opinnäytetyössä on tarkoituksenaan kehittää vaihtoehtoinen etätyöasemavirtualisointisovelluksen prototyyppi, minkä vuoksi opinnäytetyö on kehittämistutkimus. Työssä on selkeä uusi aihe ja tarve, jota täytyy olemassa olevien ratkaisujen puuttumiseksi lähteä kehittämään. Kehittämistutkimus onkin tutkimustyötä, jonka olennaisimpia osia on ongelmien poistaminen ja muutoksien aikaansaaminen (mts, 2015, 9–11). Bister (2019, 43) kirjoittaa kirjassaan, että

opinnäytetyön kokoisissa kehittämisprojekteissa suurien ja monimutkaisten kehittämishaasteiden pilkkominen pienempiin osiin on keskeinen menettelytapa.

2.4 Kehitystutkimuksen käytännön toteutus

Tutkimuksen alussa perehdytään virtualisointiin, työasemavirtualisointiin ja näitä aiheita sivuavaan kirjallisuuteen sekä verkkolähteisiin. Teoriapohjan keräämisen jälkeen suoritetaan nykytilan kartoitus, eli käydään läpi jo olemassa olevia järjestelmiä ja tarkastellaan niitä analyyttisellä silmällä verraten niitä samalla nimettömänä pysyvän toimeksiantajan tarpeisiin.

Teoriapohjan keräämisen ja nykytilan kartoituksen jälkeen suoritettavassa varsinaisessa kehitystyössä pyritään kehittämään toimiva etätyöasemavirtualisoinnin mahdollistava järjestelmän prototyyppi. Pohdintaosiossa tarkastellaan tehtyä työtä analyyttisesti kriittisellä silmällä. Saatuja tuloksia verrataan tutkimuskysymyksiin ja tarkastellaan, onko tutkimuskysymyksiin vastattu. Samassa kappaleessa pohditaan myös tutkimuksen validiteettia ja analysoidaan jatkokehityskohteita ja kehitystyön tuloksen puutteita.

3 Virtualisointi

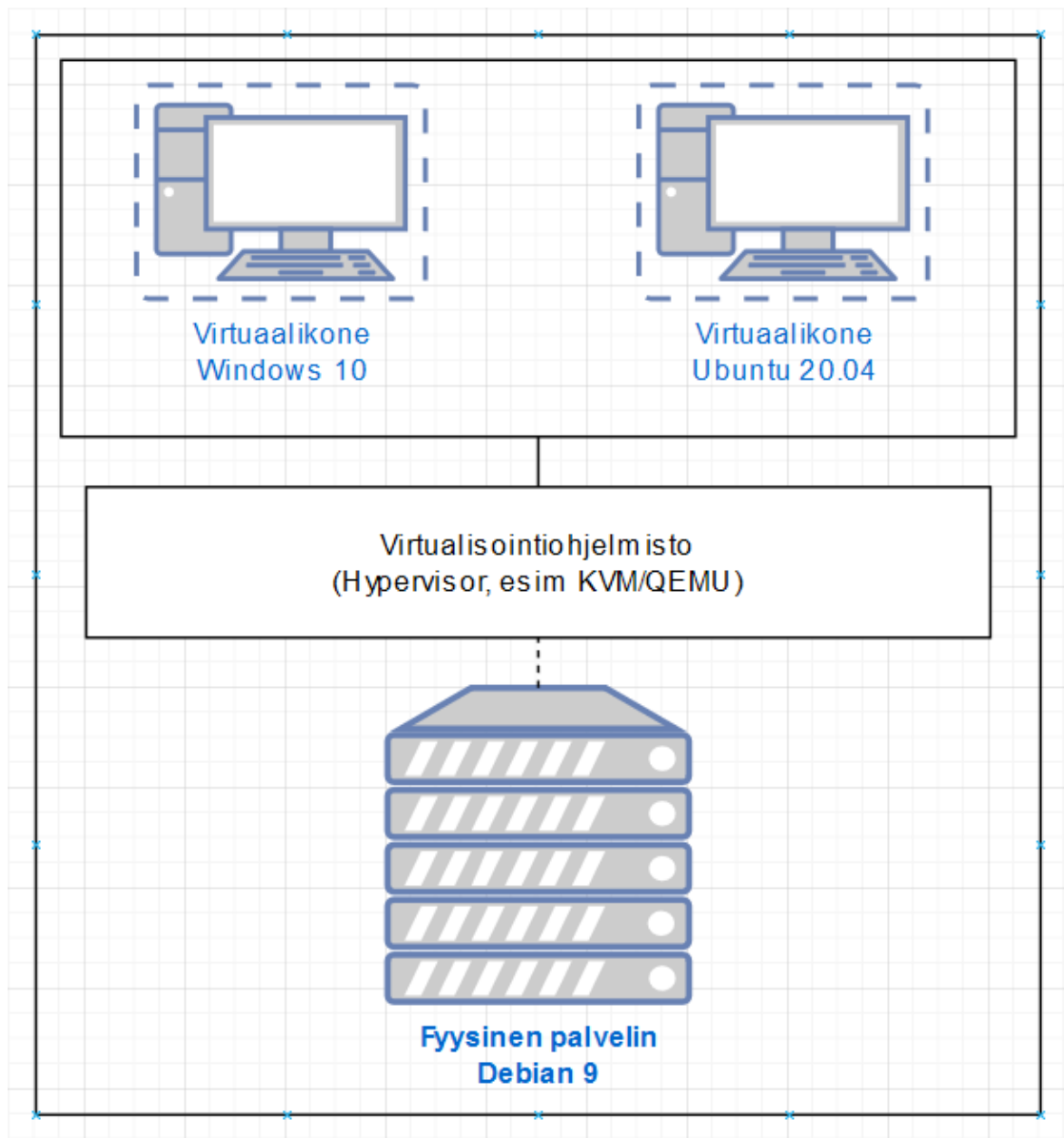
”Filosofiassa virtuaalinen tarkoittaa ’jotain, joka ei ole todellinen’ tietojenkäsittelytieteessä virtuaalinen tarkoittaa ’laitteisto, joka ei ole todellinen’ (Chiramal, Mukhedkar & Vettahu 2016, 1)

Viimeisen kuuden vuosikymmenen aikana tietojenkäsittely on kulkenut monien avaintrendien läpi. Portnoyn (2016) mukaan kuusi- ja seitsenkymmenluvulla keskustietokoneet jylläsivät konesaleissa, 80- ja 90-luvuilla pöytä tietokoneet ja kannettavat tietokoneet tekivät tuloaan, ja kaksituhattluvun taitteessa internet mullisti tietojenkäsittelyn täysin. Hän väittää myös, että elämme tällä hetkellä keskellä uutta tietojenkäsittelyn mallia muuttavaa avaintrendiä. Virtualisointia. (Portnoy 2016, 1–2)

3.1 Virtualisoinnin määritelmä

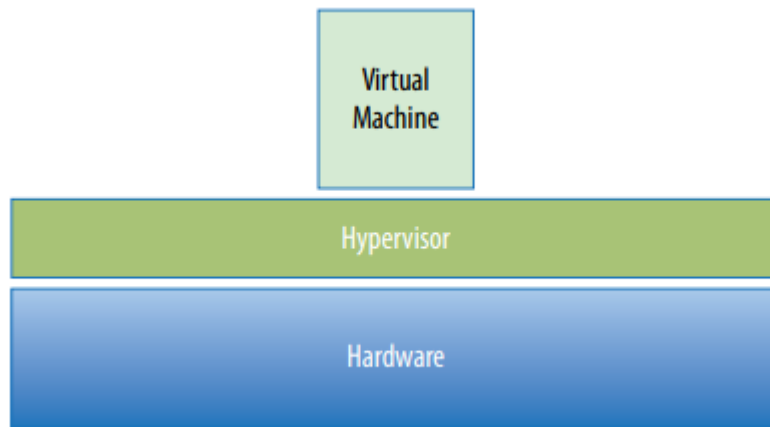
Yleisesti ottaen tietotekniikassa, kuten myös tämän opinnäytetyön kontekstissa, virtualisointi tarkoittaa menetelmää, jolla oikeita olemassa olevia fyysisiä resursseja voidaan hallinnoida loogisina resursseina. Virtualisoinnin avulla voidaan käyttää palvelimen resurssit tehokkaammin jakamalla niitä useammalle käyttöjärjestelmälle samalla pitäen virtuaalikoneet erotettuina toisissaan. Goldenin (2009) mukaan organisaatiot voivat nostaa fyysisten resurssien käyttöastetta dramaattisesti ottamalla virtualisoinnin käyttöön, Portnoy (2016) on teoksessaan samaa mieltä ja lisää "Virtualisointi on häiritsevä teknologia, joka rikkoo olemassa olevan nykytilan siitä kuinka fyysisiä tietokoneita ja palvelimia käsitellään ja kuinka budjetit määritellään." (Golden 2009, 2–3; Portnoy 2016, 2–11).

Yksinkertaistettuna virtualisointitekniikan avulla voit ajaa tietokoneella, jossa on Debian 9 käyttöjärjestelmä, kahta eri tietokonetta, joissa toisessa on Ubuntu Linux 20.04 ja toisessa Windows 10 (Golden 2009, 1).



Kuvio 2. Esimerkkidiagrammi virtualisoinnin arkkitehtuurista

Virtualisointi vaatii isäntäkoneelle asennetun *hypervisor-ohjelmiston*, joka sijaitsee fyysisen laitteiston ja vieraskoneen (tai koneiden) välissä. Hypervisorin tehtävänä virtualisoinnissa on luoda fyysisistä resursseista (kuten esimerkiksi massamuisti, prosessori, näytönohjain, äänikortit ja verkkokortit) loogisia kokonaisuuksia, joita se jakaa vierastietokoneiden käyttöön. Voidaankin siis sanoa, että hypervisor on resurssien tuomari. (Golden 2016, 21–22). Chirammalin, Mukhedkarin ja Vettathun (2016, 13) mukaan hypervisor huolehtii myös virtuaalikoneiden elinajan aikaisesta hallinnasta, sekä niiden siirtämisestä fyysiseltä laitteelta toiselle.



Kuvio 3. Hypervisorin sijainti virtualisoinnissa (Golden 2016)

Aikaisemmin hypervisorista on käytetty myös termiä virtuaalikonemanageri (virtual machine manager), mutta Goldenin (2016, 21) mukaan se on pudonnut pois muodista, ja hypervisor on terminä käytännössä korvannut sen. Chirammalin, Mukhedkarin ja Vettathun (2016, 13) teoksessa termejä käytetään kuitenkin keskenään ristiin rastiin.

Tietokoneessa, jossa hypervisor -ohjelmistoa ei ole asennettu, käyttöjärjestelmä kommunikoi suoraan laitteiston kanssa. Käyttöjärjestelmän lähettämät I/O pyynnöt menevät suoraan laitteistolle, esimerkiksi muistia allokoimassa pyyntö tehdään suoraan fyysisestä muistista. (Golden 2016, 23).

3.2 Virtualisoinnin hyödyt

Chirammalin, Mukhedkarin ja Vettathun (2016, 3–5) teoksessa seitsemän hyvää puolta virtualisoinnin käytöstä. Tässä aliluvussa on avattu niistä neljä opinnäytetyön kannalta merkityksellisintä.

Palvelinten yhdistäminen

Palvelinvirtualisoinnissa fyysisen isäntätietokoneen resurssit jaetaan usean toisistaan eristetyn vieraskäyttöjärjestelmäinstanssin käyttöön. Portnoyn (2016, 16–17) mukaan yleensä jokainen isäntäkoneelle asennettu vieraskäyttöjärjestelmä tuottaa vain yhden palvelun, tai palvelukokonaisuuden. Palvelinvirtualisoinnin avulla

isäntätietokoneen resurssien käyttöaste kasvaa ja täten isäntätietokoneen investointikustannukset ”maksavat itsensä takaisin” nopeammin. Tämä johtuu siitä, että ohjelmistot ja palvelut harvemmin käyttävät kaiken käytettävissä olevan suorituskyvyn, ja virtualisoinnin avulla käyttöaste kasvaa. (Mts, 16–17; Golden 2009, 3–4). Chirammal, Mukhedkar ja Vettathu (2016, 3) lisäävät että fyysisten palvelimien vähentyminen vapauttaa lattiapinta-alaa, sekä aiheuttaa virrankulutuksen vähenemistä aiheuttaen rahallisia säästöjä.

Palveluiden eristäminen

Järjestelmien eriyttämisessä on useita hyviä puolia. Infrastruktuurista tulee vikasietoisempi, koska yhden virtuaalipalvelimen vikatila ei aseta koko infrastruktuuria vikatilaan (Portnoy 2016, 16–17). Palvelujen eristäminen estää myös yhteensopivuusongelmia eri sovellusten ja sovelluskokonaisuuksien välillä (Chirammal, Mukhedkar & Vettathu 2016, 4).

Nopeampi tietokoneiden ja palvelimien provisiointi

Chirammalin, Mukhedkarin ja Vettathun (2016, 4) mukaan virtualisoidun tietokoneen käyttöönotto on huomattavasti nopeampaa kuin fyysisen tietokoneen käyttöönotto. Virtuaalikoneen luominen ja käynnistäminen voidaan automatisoida, jonka avulla virtuaalikone voidaan luoda lyhyessä ajassa.

Hätätilanteesta palautuminen

Esimerkiksi myrskystä, tulvasta, tulipalosta tai muusta hätätilanteesta palautuminen on nopeaa. Vierasjärjestelmistä voi ottaa tietyin aikamäärein tilannekuvan, johon tallennetaan vierasjärjestelmän sen hetkinen tila. Nämä tilannekuvat voi nopeasti siirtää toimivalle isäntäkoneelle, tai palauttaa vierasjärjestelmän toimivaan tilaan. Tilannekuvat sallivat myös vierasjärjestelmien siirtämisen eri isäntäkoneiden välillä. (Chirammal, Mukhedkar & Vettathu 2016, 4). Portnoyn (2016, 14) mukaan normaalin datakeskuksen uudelleenrakentamiseen olisi mennyt päiviä tai jopa viikkoja, mutta virtualisointitekniikoiden avulla tämä voi tapahtua tunneissa tai minuuteissa.

3.3 Virtualisoinnin haasteet ja haitat

Korkeat implementointikustannukset

Virtualisoinnin avulla voi säästää rahaa pitkällä aikavälillä, mutta alkusijoitus on suuri. Virtualisointiohjelmistolisenssien hinnat ovat kalliita, jos organisaatio ei käytä avoimen lähdekoodin työkaluja. Kustannuksia tulee myös uusittavasta infrastruktuurista. Järeät tehokkaat isäntäkoneet ovat kalliita, ja niiden rakentamiseen sekä asentamiseen kuluu sekä aikaa että rahaa. (Virtualization 2.0 - Pros and Cons, N.d; Gaille, 2017)

Nopea skaalautuminen on haastavaa

Virtuaalikoneella skaalaaminen on nopeaa ja helppoa, paitsi jos mennään isäntäkoneen suorituskyvyn rajojen ulkopuolelle. Virtualisoinnissa pitää ottaa huomioon sovellukset, tietoturva, sekä massamuistin määrä. Tämä johtaa kustannusongelmiin ja pitkiin hankinta-aikoihin. (Taylor, N.d.)

Mahdolliset suorituskykyongelmat

Taylor (N.d.) nostaa esiin artikkelissaan mahdolliset suorituskykyongelmat. Koska virtualisoinnissa isäntäkoneen resurssit on jaettu vieraskoneiden käyttöön, voi ongelmaksi muodostua se, että yksi vieraskone käyttää liikaa resursseja "varastaen" suoritinaikaa muilta vieraskoneilta. Taylor (N.d.) ja Gaille (2017) molemmat nostavat esiin ongelman. Vieraskoneen luominen on liian helppoa. Tämä ongelma voi manifestoitua suorituskykyongelmina, koska isäntäkoneella ajetaan liian monta vieraskonetta. Se voi myös johtaa tietojen katoamiseen ja odottamattomiin palvelukatkoksiin.

Oppimiskynnys

Gaille (2017) kertoo artikkelissaan, että virtualisoidun ICT-ympäristön ylläpitäminen ja implementoiminen vaatii ylläpitäjiltä ja henkilökunnalta paljon osaamista virtualisoinnin saralta. Karlin (2016) mukaan virtualisoinnin saralla kokemattomat IT-ammattilaiset saattavat olla negatiivinen aspekti. Hänen mukaansa on olemassa mahdollisuus, että kokematon järjestelmäasiantuntija konfiguroi järjestelmän väärin hukaten näin palvelinkapasiteettiä ja sähköä.

3.4 Virtuaalinen työasemainfrastruktuuri

“Client and application virtualization is a simple concept: decoupling the operating system or application from the end-user computing device. Essentially, the computing takes place on a server in the data centre and presents a desktop or application to the user.” (Coutinho, Schaapman & Temin 2010, 8)

Työasemavirtualisoinnilla tarkoitetaan järjestelmää, joka eriyttää työpöydän laitteesta, jolla sitä käytetään (IBM Cloud Education, 2019). Tämä tarkoittaa sitä, että työpöydän tarjoavaa käyttöjärjestelmää ajetaan vieraskoneessa, isäntäkoneen päällä, virtualisoinnina. Isäntäkone on yleensä tehokas palvelin, jossa on paljon keskusmuistia, kymmenillä ytimillä varustettu prosessori, satoja tai tuhansia gigatavuja massamuistia sekä nopea verkkoyhteys. (What is Desktop Virtualization? N.d.) Työasemavirtualisointia voi pitää eräänlaisena asiakas - palvelin mallin muotona, koska virtualisoitu työasema sijaitsee keskitetyssä palvelimessa, eikä päätietokoneessa, jolla sitä käytetään. (Beal. V, N.d.).

VDI (virtual desktop infrastructure) on virtualisointistrategia, jossa suuri osa yrityksen työasemista on virtualisoitu ja niitä pääsee käyttämään joko lähiverkosta tai ulkoverkon kautta VPN-yhteyden avulla. (Coutinho, N., Schaapman, P., Temin, T. 2010, 8). Coutinho, Schaapman ja Temin (2010, 8–10) kertovat teoksessaan virtuaalityöasemainfrastruktuurissa työaseman koostuvan yhdestä tai useammasta levykuvasta, jotka sisältävät käyttöjärjestelmän, sovellukset, käyttäjän tiedostot sekä asetukset. Tämä kyseinen levykuva ladataan palvelimelle, virtualisoidaan ja suoritetaan hypervisorin avulla.

Työasemavirtualisoinnin hyvät puolet ovat osaltaan samoja kuin virtualisoinnissa yleensä. Virtuaalisia työasemia voi luoda ja käynnistää minuuteissa yhden käyttöliittymän kautta. Tämä nopeuttaa muun muassa uusien työntekijöiden sisäänottoprosessia. Ylläpito on myös helpompaa. Virtualisointiohjelmisto luo virtuaalityöasemista automaattisesti levykuvia, jotka varmuuskopioidaan. Koska virtuaalityöasemia ajetaan keskitetyllä palvelimella, ja käytetään thin clientillä, virrankulutus pienenee. (Top 10 key benefits of Desktop Virtualization, N.d.)

Adams (2016) nostaa esiin myös tietoturvan. Parannettu tietoturva johtuu siitä, että kaikki tieto on keskitetyllä palvelimella eikä käyttäjän päätelaitteella. Kellyn (2017) mukaan virtualisoitu työasema nostaa työntekijöiden tuottavuutta. Samaa työasemaa voi käyttää kaikkialta, kunhan saatavilla on nopea internetti, jolloin tiedot löytyvät aina samasta paikasta. Samoin Kellyn (2017) mukaan tuottavuus ei laske, vaikka päätelaite, jolla työasemaa käytetään, menisi rikki koska toisen päätelaitteen käyttöönotto on nopea toimenpide.

Kuten kaikissa järjestelmissä, myös työasemavirtualisoinnissa on huonot puolet ja haasteet. Madden (2018) nostaa esimerkeiksi neljä asiaa. Oheislaitteet, varsinkin USB 3.2 gen 1 jonka teoreettinen tiedonsiirtonopeus on 5 gigabittiä sekunnissa, vaativat paljon kaistaa toimiakseen oikein. Seuraava haaste on tuki usealle korkearesoluutioiselle näytölle. Vaikka teknologia on kehittynyt ei toimivuus silti ole varmaa. Kolmantena ongelmana on vaatimus verkkoyhteydelle. Jos verkkoyhteyttä ei ole, etätyöasemavirtualisointi ei toimi. Neljäs haaste on multimedia ja tuki av-laitteille. Laitteiden uudelleenohjaus ja sekä niiden käyttäminen Skypessä, Microsoft Teamsissa ja Slackissa.

Työasema palveluna

Konseptin tasolla ”työasema palveluna” (DaaS, Desktop as a Service) on hyvin samantyyppinen virtuaalisen työasemainfrastruktuurin kanssa. DaaS -palvelu onkin käytännössä virtuaalityöasemainfrastruktuuri, mutta pilvipalveluna. Tämä tarkoittaa sitä, että organisaation ei tarvitse omistaa omaa IT-infrastruktuuria. (What is Desktop as a Service (DaaS)?, N.d). ”Työasema palveluna” palvelussa tilaaja ostaa palveluntarjoajalta tilauspohjaisen palvelun. Palveluntarjoaja toimittaa tilaajalle käytön ja tilauksen mukaan virtuaalityöpöytiä, joihin tilaaja voi itse määrittää haluamansa käyttöjärjestelmät ja asennettavat sovellukset. (Whitney, 2020a).

Whitneyn (2020a) mukaan DaaS palvelussa ideana on ulkoistaa tilaajayrityksen IT-osasto, tukihenkilökuntaa ja datakeskuksen resursseja siirtämällä vastuu DaaS palvelun tuottajalle. Tilaajayrityksen IT-osaston vastuulle jää etätyöpöytien konfigurointi ja hallinnoiminen.

Pros and Cons of Desktop Types



Kuvio 4. Eri työasematyyppien hyviä ja huonoja puolia. (Whitner, 2020)

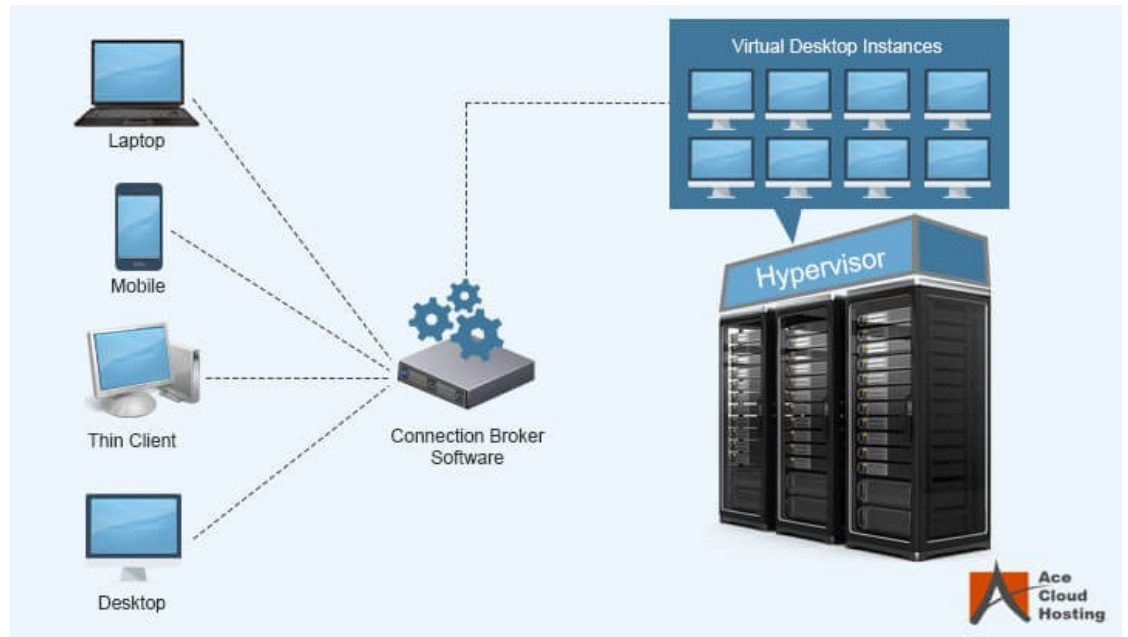
3.5 Virtuaalisen työasemainfrastruktuurin osaset

Virtualisointi vaatii prosessoritason tuen isäntäpalvelimelta. Palvelimessa tulee olla riittävä määrä prosessoriytimiä, keskusmuistia ja massamuistia sujuvan toiminnan varmistamiseksi. Schimscheimer (N.d.) suosittelee teoksessaan, että jokaiselle Windows 7 64-bit käyttöjärjestelmää ajavalle vieraskoneinstanssille olisi varattu kaksi virtuaaliydintä ja neljä gigatavua keskusmuistia.

Hypervisor -ohjelmistoa ajetaan isäntäkoneella. Hypervisor käyttää komponentteja suoraan ja jakaa niitä ja virtualisoiduille käyttöjärjestelmille tarpeen mukaan. Hypervisor -ohjelmistoja on usealta eri organisaatiolta. Esimerkkeinä Linux -kernelin mukana tuleva *KVM*, Microsoftin *Hyper-V*, *Xen*, ja VMWaren *ESXi*. (What is a hypervisor? N.d.)

Yhteydenvälittäjä (Connection broker) -ohjelmisto on vastuussa käyttäjien yhdistämisestä virtuaalityöasemainstansseihin. Connection brokeria käytetään myös käyttäjien autentikointiin. Yhteydenvälittäjän rooli on myös päivittää

virtuaalityöasemien tilaa, joko aktiiviseksi tai passiiviseksi riippuen siitä onko sillä aktiivinen yhteys vai ei. (What Is VDI And How Does It Work? N,d)



Kuvio 5. Yksinkertaistettu VDI arkkitehtuuri (What Is VDI And How Does It Work? N,d)

Kuten kuviossa 5. demonstroidaan, VDI-järjestelmään voi yhdistää lähes millä tahansa päätelaitteella. Yleisimmin kuitenkin käytössä on niin sanottuja kevyitä asiakaspäätteitä (thin client). Kevyt asiakaspääte on yleensä pieni, vähätehoinen ja, vähän virtaa kuluttava tietokone, jossa ei ole ollenkaan omaa massamuistia. Vaihtoehtoja kevyisiin asiakaspäätteisiin löytyy useita. x86 prosessoriarkkitehtuuriin perustuvista järjestelmistä vaihtoehtoja on muun muassa Intelin NUC ja Compute Stick laitteet ja tavanomaiset työpöytäkoneet. ARM arkkitehtuuriin perustuvista taasen Raspberry Pi on suosittu vaihtoehto. (What is a Thin Client? N.d.)

4 Etätyöasemavirtualisoinnin toteutus

Virtuaalityöasemainfrastruktuuri on jo vakaa teknologia ja se on implementoitu sekä käytössä useassa olemassa olevassa yrityksessä. Tässä luvussa käsitellään aliluvuttain alustava selvitystyö, jossa on nykytilan analyysi, prototyypin kehitys, jossa kuvataan

kehitystoimet, joita prototyypin eteen ja validointi, jossa kuvataan prototyypin toimintaa ja esitellään avainlukuja.

4.1 Alustava selvitystyö

Prototyypin kehityksessä lähdetään liikenteelle tilanteesta, jossa olemassa on olemassa kaupallisia ratkaisuja (muun muassa Citrixiltä ja VMWarelta), mutta mitään ilmaista avoimen lähdekoodin valmista ratkaisua ei ole löytynyt. Alustavan selvitystyön perusteella nykytila-analyysiin kolme eri järjestelmää, jotka ovat kaikki suosittuja ja paljon käytettyjä. Valittuja VDI/DaaS palveluita ovat, VMWare Horizon, Windows Virtual Desktop ja Red Hat Virtualization

VMWarella on tuote nimeltä VMWare Horizon, joka on joko kuukausimaksullinen tai kertamaksullinen VDI palvelu, josta on tarjolla myös ilmainen selaimessa rajoitettu kokeiluversio. VMWare Horizon Standard (N.d) palvelua myydään lisenssipohjaisesti ja lisenssejä joutuu ostamaan niin monta, kuin samanaikaisia käyttäjiä on. Yhden vuoden 24/7/365 saatavilla olevalla tukijaksolla, yhden lisenssin hinta on 3219 Yhdysvaltojen dollaria. Kymmenen yhtäaikaisen käyttäjän hinta olisi 31 160 Yhdysvaltojen dollaria. VMWare Horizonin infrastruktuuriin voi asentaa minkä tahansa vieraskäyttöjärjestelmän, ja asiakasohjelmistot tukevat useimpia mobiililaitteita, käyttöjärjestelmiä ja verkkoselaimia. VMWare Horizon (N.d.).

Microsoftin DaaS -palvelu on nimeltään Windows Virtual Desktop. Palvelu tukee pelkästään Microsoftin Windows 10, Windows Server ja Windows 7 käyttöjärjestelmiä. Palvelun asiakasohjelmiston saa useimmille mobiililaitteille, Windows ja macOS käyttöjärjestelmille ja verkkoselaimelle. (Windows Virtual Desktop, N.d.). Microsoft ei suoraan listaa palvelun hintaa verkkosivuillaan, mutta palvelu sisältyy Microsoft 365 Business Premium (N.d.) lisenssiin, joka maksaa 20 Yhdysvaltojen dollaria kuukaudessa.

Red Hat Virtualization on Red Hatin VDI -palvelu. Se perustuu Red Hatin omaan Linux distribuutioon, Red Hat Enterprise Linuxiin (RHEL) sekä KVM hypervisorin ja oVirt hallintatyökaluun. Red Hat Virtualizationin avulla voi virtualisoida Microsoftin

Windows käyttöjärjestelmiä sekä RHEL ja SUSE Linux distribuutioita. VDI-palvelun hinta on 24/7/365 tuella 1499 Yhdysvaltojen dollaria/vuosi per hypervisor/prosessorikanta.

Kaikki nykytilan analyysiin päätyneet vaihtoehdot olivat maksullisia ja ominaisuuksiltaan puutteellisia joillain tavoin puutteellisia. Red Hat Virtualization olisi ollut avoimen lähdekoodin ideologialtaan hyvä ratkaisu, niin kuin nimettömän pysyttelevän toimeksiantajan tahtotila oli, mutta sekin oli maksullinen. Koska määritettyjä ominaisuuksia ei täytetty, päätettiin lähteä kehittämään järjestelmästä prototyyppiä.

4.2 Prototyypin kehitys

Kehitystyön tarkoituksena on luoda prototyyppi etätyöasemavirtualisointiin käytettävästä järjestelmästä. Prototyypin toiminallisuudet rajattiin yhden työpöydän virtualisoimiseen ja sen sujuvaan käyttöön asiakaskoneella. Testiympäristö rakennettiin opinnäytetyön tekijän työhuoneeseen. Prototyypin toteutukseen käytettiin Raspberry Pi 4 pientietokonetta (asiakaskone), tehokasta kannettavaa tietokonetta (isäntäkone) sekä tarvittavia oheislaitteita ja ohjelmistoja. Muutamia tarvittuja ohjelmistoja olivat: KVM, Qemu, Weston ja Waypipe. Työhuoneesta löytyy 100MBit/s symmetrinen internetyhteys sekä 1Gbit/s lähiverkko.

Isäntäkoneeksi prototyypin kehityksessä valikoitui Asuksen ROG Zephyrus G14 GA401 kannettava tietokone

- Prosessori: AMD Ryzen 4800HS (8 ydintä, 16 säiettä)
- Näytönohjain 1: AMD Vega 7
- Näytönohjain 2: Nvidia GTX 1660ti MaxQ
- Keskusmuisti: 16gb DDR4 3200MHz
- Massamuisti: Samsung 970 Evo nVME 250gb SSD

Tietokone valittiin koska siinä on nopea prosessori kahdeksalla ytimellä ja kuudellatoista säikeellä, paljon nopeaa keskusmuistia sekä kaksi näytönohjainta.

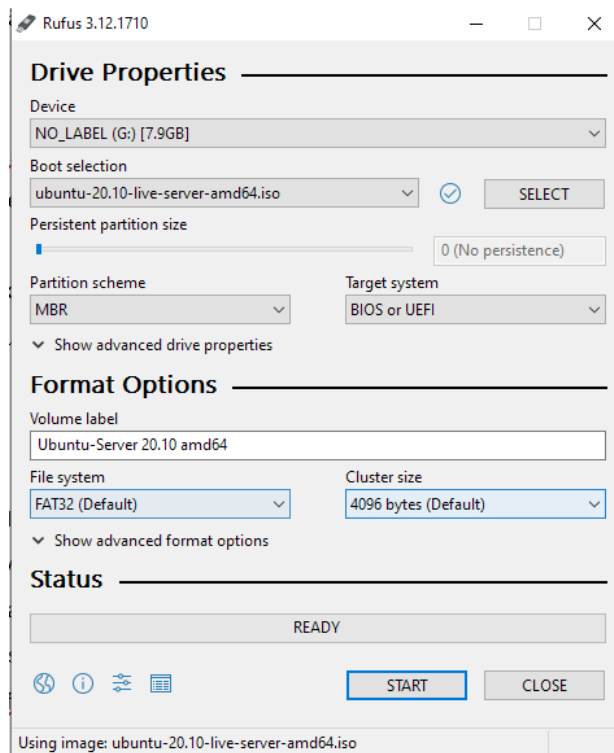
Käyttöjärjestelmäksi isäntäkoneeseen valikoitui Ubuntu Server 20.10, Linux 5.8 kernelillä. Valinta kohdistui siihen siksi, että se oli opinnäytetyön tekohetkellä Debian pohjainen serverikäyttöjärjestelmä ja siinä oli vakiona uudempi kernel versio kuin muissa vaihtoehdoissa. Toisena osatekijänä oli se, että käyttöjärjestelmään ei ole asennettu mitään ylimääräistä, joka tietysti säästää resursseja. Käyttöjärjestelmään ei ole myöskään asennettu ikkunointimanageria tai työpöytäympäristöä, joten ne täytyy itse asentaa.

Ubuntu Server 20.10 on myös ensimmäinen versio, joka tukee Raspberry Pi 4 pientietokonetta (Ubuntu 20.10 on Raspberry Pi delivers the full Linux desktop and micro clouds, 2020), ja se valittiin käyttöjärjestelmäksi asiakaskoneelle. Etuna tässä on se, että isäntäkone ja asiakaskone voivat molemmat käyttää samaa käyttöjärjestelmää, tämä vähentää järjestelmän kompleksisuutta ja helpottaa konfigurointia.

4.2.1 Isäntäkoneen asentaminen ja konfigurointi

Käyttöjärjestelmän asennus

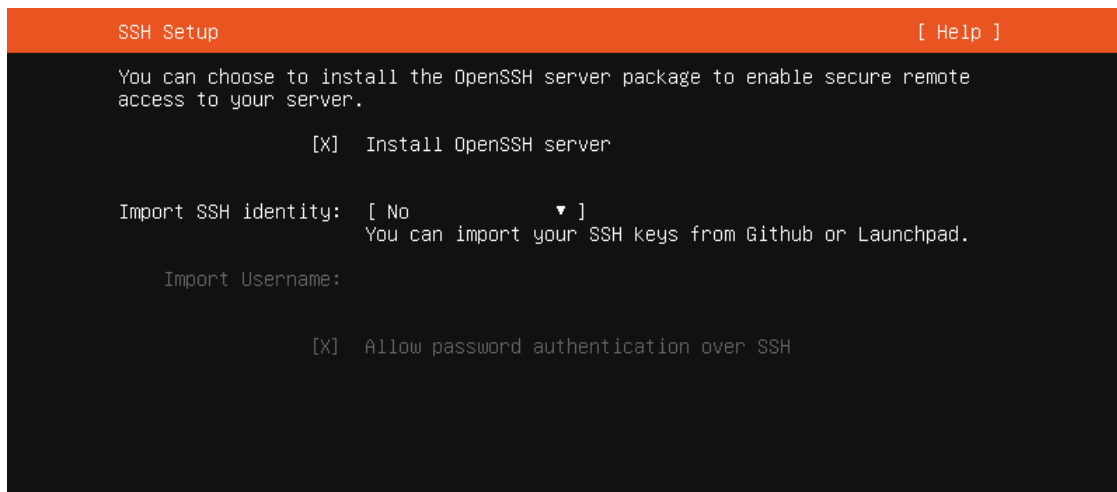
Käyttöjärjestelmien asennus koneille oli suoraviivainen prosessi. Isäntäkoneelle käyttöjärjestelmä asennettiin lataamalla Ubuntu Server 20.10 asennusmedia ISO-tiedostona Ubuntun verkkosivuilta, josta luotiin käynnistettävä USB-asema Rufus ohjelmistolla. Rufus on ohjelmisto, jota käytetään USB asennusmedioiden luomiseen. Rufus osaa luoda käynnistettävän USB-aseman ISO tiedostoista (Windows, Linux, jne). (Rufus, N.d.)



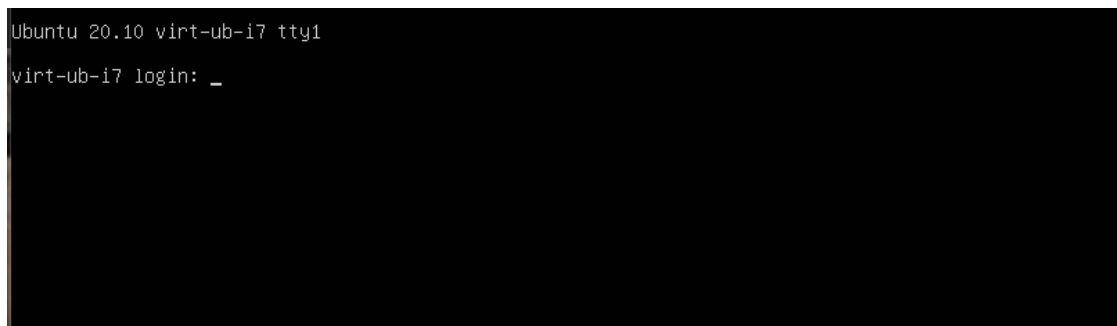
Kuvio 6. Rufus ohjelmisto konfiguroituna

USB-asemalta käynnistäminen vaatii tietokoneen BIOS tai UEFI-asetusten muuttamista. Se tapahtuu eri tietokoneilla eri tavalla. Prototyypissä käytettävän ROG Zephyrus G14 -tietokoneella UEFI-asetuksiin pääsee joko käyttöjärjestelmästä, tai painamalla käynnistyksen yhteydessä F2 näppäintä. UEFI-asetuksista pitää valita USB-asema, jolta tietokone halutaan käynnistää.

Asennuksen aikana asennusohjelmisto kysyy käyttäjältä muun muassa kieltä, verkkoasetuksia sekä kiintolevyn osioimiseen liittyviä asetuksia. Vakioasetukset olivat prototyypille suotuisia, ja muutoksia ei OpenSSH -palvelimen asennusta lukuun ottamatta (ks. Kuvio 6.). Asennusohjelman suoritettua toimenpiteensä, pitää tietokone käynnistää uudelleen. Uudelleenkäynnistyksen jälkeen tietokone tervehtii kauniilla mustavalkoisella terminaalilla ja sisäänkirjautumisruudulla (ks. Kuvio 7.).



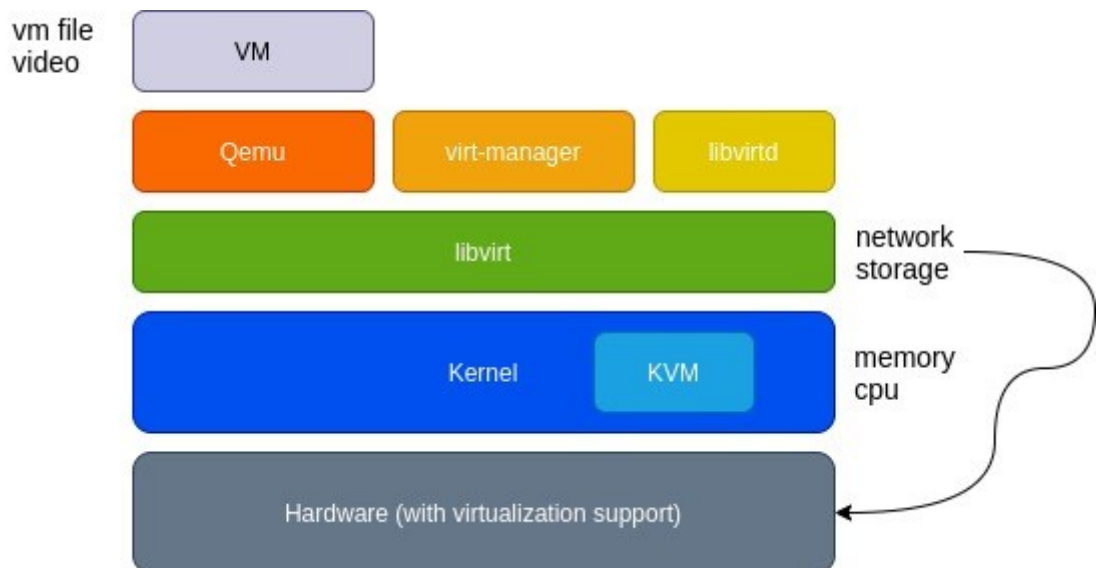
Kuvio 7. Ubuntu Server 20.10 asennus, OpenSSH valittuna



Kuvio 8. Ubuntu server 20.10 asennettuna, sisäänkirjautumista vailla

Tarvittavien ohjelmistojen asennus

Tutkimusosan ja selvitystyön perusteella prototyyppiin valikoitui hypervisoriksi KVM ja virtualisointiteknologiaksi sen kanssa QEMU. Virtualisoinnin ja virtuaalikoneiden hallintaohjelmaksi tuli luonnostaan Virtual Machine Manager (virt-manager). Virt-manager on käyttöliittymäpohjainen hallintaohjelmisto, jolla virtuaalikoneita voi hallita ja se käyttää *libvirt* kirjastoa.



Kuvio 9. virt-managerin, libvirtin ja Qemu/KVM:n suhteet toisiinsa (Portier, 2020)

Ennen virtualisointiteknologioiden asentamista, kannattaa tarkastaa, että laite tukee virtualisointia. Terminaalissa voi ajaa komennon `lscpu |grep Virtualization`.

Ensimmäinen osa `lscpu` listaa kaikki prosessorin ominaisuudet ja jälkimmäinen osa `|grep Virtualization` etsii ominaisuuksista `Virtualization` kenttää. Jos virtualisointi on tuettu palauttaa komento AMD järjestelmällä tekstin `AMD-V` (ks. Kuvio 10) ja Intel järjestelmällä tekstin `VT-x`. Jos prosessori tukee virtualisointia, seuraavana kannattaa tarkastaa onko virtualisointi päällä. Tähän tarvitaan ohjelmisto, nimeltä `cpu-checker` jonka saa asennettua komennolla `sudo apt install cpu-checker`.

Asennuksen jälkeen tulee ajaa komento `sudo kvm-ok`, jos virtualisointi on käytössä, tulostaa ohjelmisto `KVM Accelleration can be used` (Ks. Kuvio 10).

```

super@zephyrus:~$ lscpu |grep Virtualization
Virtualization:          AMD-V
super@zephyrus:~$ kvm-ok
INFO: /dev/kvm exists
KVM acceleration can be used
super@zephyrus:~$

```

Kuvio 10. Virtualisointi löytyy, ja KVM kiihdytystä voi käyttää

Virtualisointiympäristön luomiseen tarvitsee asentaa useampisovelluspaketti ja niiden riippuvuudet. Asennuskomento on *"sudo apt install qemu-kvm virt-manager libvirt-daemon-system virtinst bridge-utils"*.

- **qemu-kvm:** tarjoaa ja luo laitteistoemuloinnin KVM hypervisorille
- **virt-manager:** käyttöliittymätyökalu, jolla voi hallinoida virtuaalikoneita
- **libvirt-daemon-system:** libvirt palvelu/daemon, virt-managerin riippuvuus
- **virtinst:** komentorivityökalu virtuaalikoneiden luo
- **bridge-utils:** komentorivityökaluja, joilla voi luoda verkkoja virtuaalikoneiden välille ja linkittää niitä ulkoverkkoon

Asennusprosessin jälkeen nykyinen käyttäjä pitää lisätä libvirt ryhmään, jotta käyttäjä saa oikeuden hallinoida virtuaalikoneita. Tämä tapahtuu komennolla *"sudo usermod -a -G libvirt käyttäjänimi"*. Käyttäjänimi korvataan käytössä olevan käyttäjän käyttäjänimellä. Prototyypin tapauksessa komento olisi *"sudo usermod -a -G libvirt super"*. Käyttäjän täytyy kirjautua ulos, tai tietokone tulee käynnistää uudelleen, että käyttäjäryhmään lisääminen tulee voimaan. Asennuksen toimivuuden voi testata tässä vaiheessa käynnistämällä virt-manager ohjelmiston, tai ajamalla komennon *"systemctl status libvirtd"*.

```

super@zephyrus:~$ systemctl status libvirt
● libvirt.service - Virtualization daemon
   Loaded: loaded (/lib/systemd/system/libvirt.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2020-11-07 19:50:01 UTC; 2h 52min ago
     TriggeredBy: ● libvirt.socket
                  ● libvirt-admin.socket
                  ● libvirt-ro.socket
        Docs: man:libvirt(8)
              https://libvirt.org
    Main PID: 1067 (libvirt)
      Tasks: 21 (limit: 32768)
     Memory: 44.0M
    CGroup: /system.slice/libvirt.service
            └─1067 /usr/sbin/libvirt
              └─1191 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefil
                └─1192 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefil

```

Kuvio 11. libvirt toiminnassa

Ikkunointiprotokollaksi valittiin Wayland, koska valitsemamme etätyöpöydän mahdollistava työkalu, Waypipe, toimii ainoastaan Waylandin avulla.

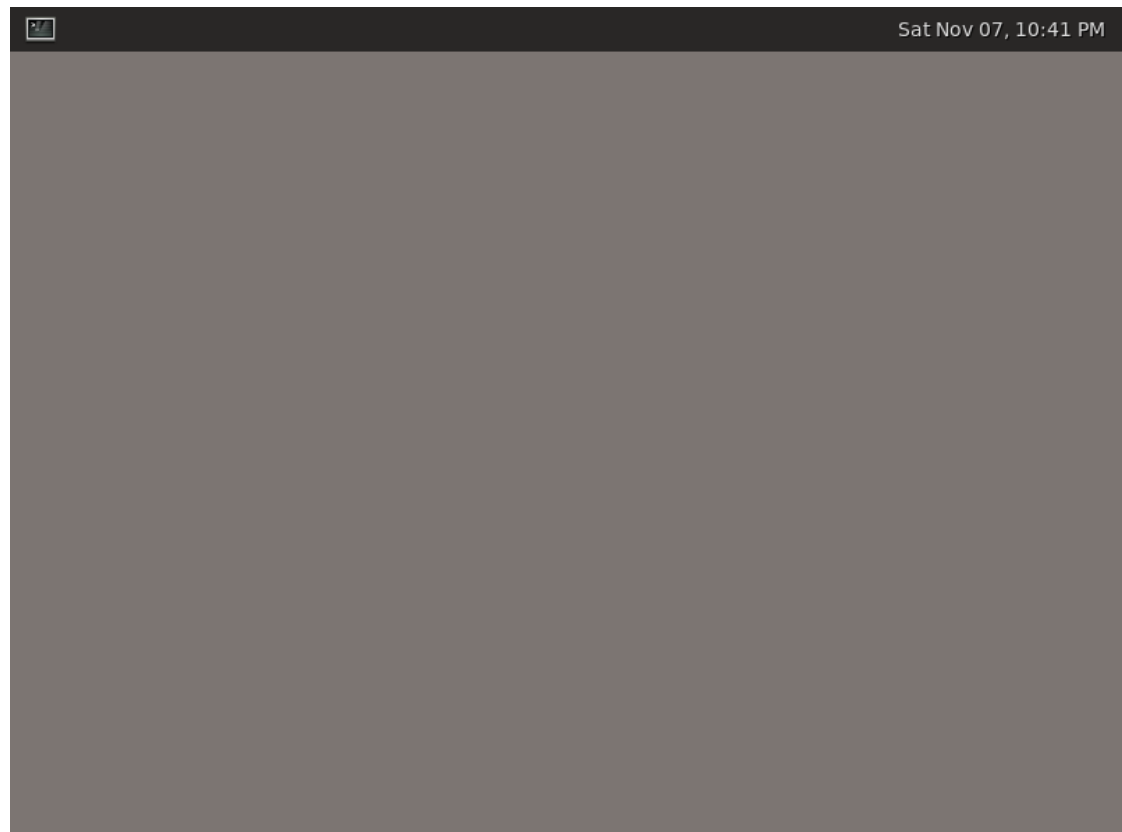
Ikkunointimanageriksi valittiin Waylandin referenssi-implemентаatio, Weston.

Westonin ja Waylandin asentamiseen riittää yksi komento. *"sudo apt install weston"*.

Tämä asentaa Westonin, ja riippuvuudet, joissa on muun muassa libwayland.

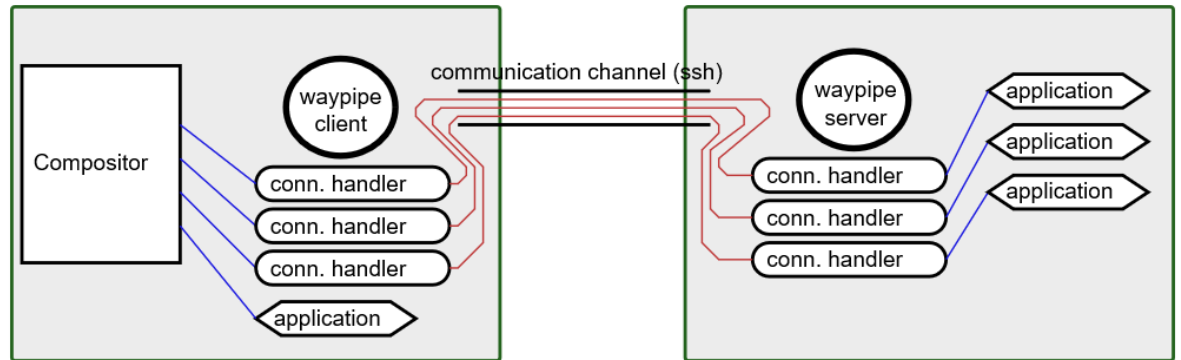
Westonin toimivuuden voi testata ajamalla terminaalissa komennon *"weston"*.

Komento käynnistää graafisen käyttöliittymän. (README.md, N.d.).



Kuvio 12. Weston asennettuna

Seuraavana vaiheena oli Waypiperin kääntäminen lähdekoodista ja asentaminen. Waypipe on ohjelmisto, joka toimii prototyypissä etätyöpöytäkäytön mahdollistajana. Waypipe toimii samaan aikaan asiakas- ja serveriohjelmistona. Severi-instanssia ajetaan isäntäkoneella (eli työasemalla, josta halutaan kuva) ja asiakasinstanssia asiakaskoneella (eli työasemalla, jossa kuva halutaan näyttää).



Kuvio 13. Waypiperin arkkitehtuuri. (Stoeckl, 2019)

Isäntäkoneella ajettava Waypipe -instanssi esittää olevansa Wayland compositor ja tarjoaa käyttöön Unix socketin johon asiakaskoneella ajettava Waypipe -instanssi voidaan yhdistää. Kun tämä tapahtuu Waypiperin isäntäkoneen instanssi luo uuden prosessin, jonka tehtävä on kääntää Wayland protokollan viestit ja lähettää ne asiakaskoneen Waypipe -instanssille. Waypipe päivittää ja lähettää protokollansa avulla vain ne muistialueet, jotka ovat muuttuneet edellisestä päivityksestä. Tämä tehdään kaistan säästämiseksi. (Stoeckl, 2019).

Waypipe täytyy kääntää lähdekoodista, koska se ei ole saatavilla Ubuntu Server 20.10:n pakettivarastoista. Ensimmäinen vaihe tästä prosessista on luoda kansio Waypiperin lähdekooditiedostoille ja kloonata uusin versio Waypiperin lähdekoodista gitlab palvelusta. Tämä tapahtuu komennoilla `"mkdir ~/build && cd ~/build"` ja `"git clone git@gitlab.freedesktop.org:mstoeckl/waypipe.git"`.

```

super@virt-ub-17:~$ mkdir ~/build && cd ~/build
super@virt-ub-17:~/build$ git clone https://gitlab.freedesktop.org/mstoeckl/waypipe.git
Cloning into 'waypipe'...
remote: Enumerating objects: 130, done.
remote: Counting objects: 100% (130/130), done.
remote: Compressing objects: 100% (130/130), done.
remote: Total 1896 (delta 73), reused 0 (delta 0), pack-reused 1766
Receiving objects: 100% (1896/1896), 824.66 KiB | 1.16 MiB/s, done.
Resolving deltas: 100% (1426/1426), done.
super@virt-ub-17:~/build$

```

Kuvio 14. Waypipen lähdekoodin hakeminen

Ennen hankitun lähdekoodin kääntämistä, täytyy asentaa kääntämiseen tarvittavat työkalut sekä riippuvuudet. Stoeckl (N.d.) listaa riippuvuuksiksi seuraavat sovellukset:

Kääntämistä varten tarvittavat työkalut:

- **meson:** (versio, ≥ 0.47 . riippuvuuksilla: ninja, pkg-config, python3)
- **C kääntäjä**

Muut riippuvuudet:

- **liblz4:** nopeaan pakkaukseen, versio: $\geq 1.7.0$
- **libzstd:** hitaampaan pakkaukseen, versio: $\geq 0.4.6$
- **libgbm:** (tuo tuen ohjelmille, jotka käyttävät OpenGL apia DMA buffereiden kautta)
- **libdrm:** ks. libgbm
- **ffmpeg:** ≥ 3.1 , tarvitsee avcodec/avutil/swscale for häviöllistä pakkaamista varten
- **libva:** rautatason enkooderi ja dekooderituki
- **scdoc:** man -sovelluksen ohjesivujen luomiseen
- **ssh:** ajon aikainen ympäristö, OpenSSH ≥ 6.7 , UNIX sockettien uudelleenohjaamista varten
- **libx264:** ffmpeg:n ajon aikainen ympäristö, ohjelmistopohjaiseen videoiden enkoodaamiseen ja dekoodaamiseen

Stoecklin listauksesta puuttuvat seuraavat paketit, joita tarvittiin Waypipen kääntämiseen:

- **wayland-protocols:** sisältää uusia ominaisuuksia Wayland protokollaan
- **pkg-config:** työkalu, jota käytetään sovelluksia ja kirjastoja käännettäessä
- **systemtap-sdt-dev:** työkalu, jolla saa tietoja ajettavasta sovelluksesta

Ohjelmistot asennetaan komennolla *"sudo apt install meson wayland-protocols liblz4-dev libzstd-dev libgbm-dev libdrm-dev ffmpeg libva-dev scdoc ssh libx264-dev pkg-config cmake systemtap-sdt-dev libavcodec-dev libavutil-dev libswscale-dev"*.

Prototyypissä käytettävällä isäntäkoneella pakettien asennus vie kiintolevytilaa 485MB. Waypipe käännetään ja asennetaan komentosarjalla. ”*meson waypipe build-waypipe && sudo ninja -C build-waypipe install*”. Komentosarjan ensimmäinen osa tarkastaa tyydyttyvätkö Waypipen riippuvuudet, ja valmistelee koodin käännettäväksi ”build-waypipe” kansioon. Jälkimmäinen osa suorittaa Waypipen asennuksen ”build-waypipe” kansioista. Waypipen asennuksen toimivuuden voi validoida komennolla ”*waypipe*”, joka toimiessaan palauttaa ohjeet sovelluksen käyttöön.

```
Usage: waypipe [options] mode ...
A proxy for Wayland protocol applications.
Example: waypipe ssh user@server weston-terminal

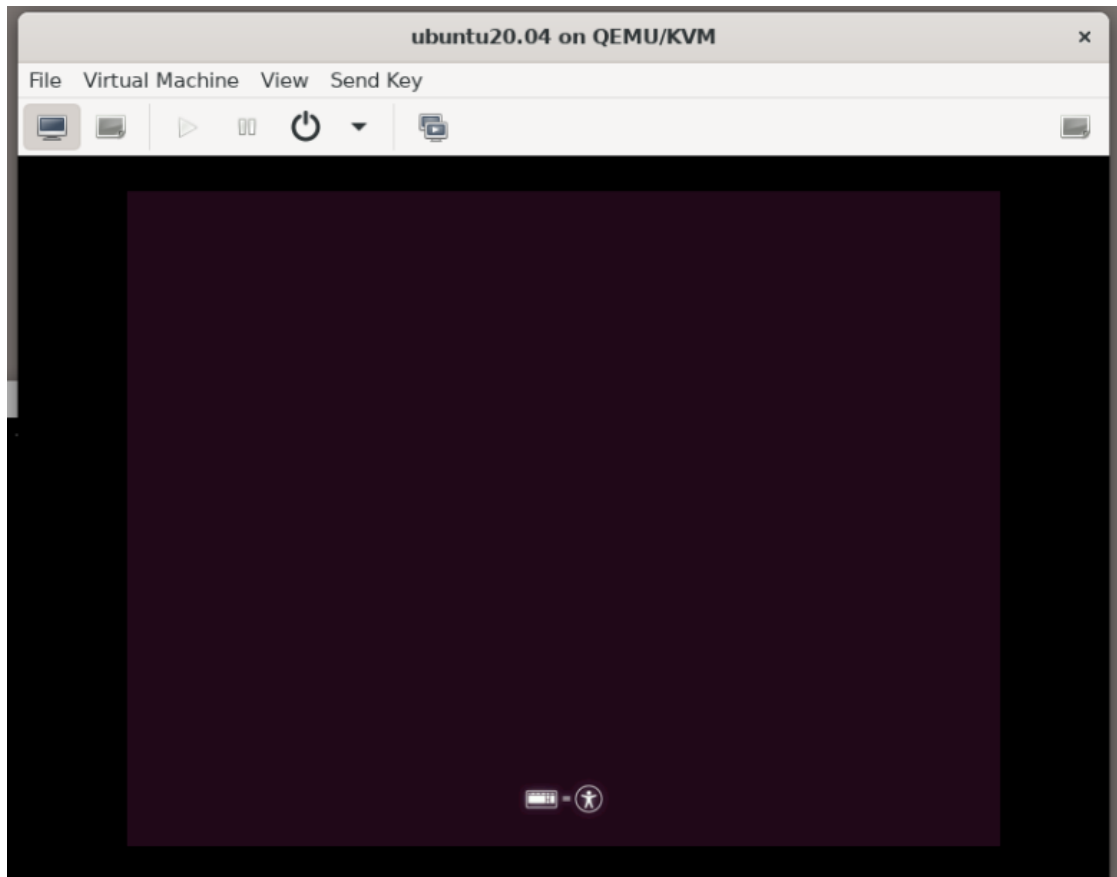
Modes:
  ssh [...]   Wrap an ssh invocation to run waypipe on both ends of the
              connection, and automatically forward Wayland applications.
  server CMD  Run remotely to invoke CMD and forward application data through
              a socket to a matching 'waypipe client' instance.
  client      Run locally to create a Unix socket to which 'waypipe server'
              instances can connect.
  recon C T   Reconnect a 'waypipe server' instance. Writes the new Unix
              socket path T to the control pipe C.
  bench B     Given a connection bandwidth B in MB/sec, estimate the best
              compression level used to send data
```

Kuvio 15. Waypipen asennus onnistui

Virtuaalikoneen luominen

Tarvittavien ohjelmistojen asentamisen jälkeen jäljellä isäntäkoneella on enää virtuaalikoneen luominen. Virtuaalikoneen käyttöjärjestelmäksi valittiin Ubuntu 20.04.1 LTS. Koska virtuaalikone luodaan graafiseen käyttöliittymään perustuvalla virt-manager ohjelmistolla, täytyy Weston käynnistää ensin. Westonin saa käynnistettyä komennolla ”*weston*”. Westonin käynnistyttyä avataan virt-manager sovellus, kirjoittamalla terminaaliin ”*virt-manager*”.

Virtual Machine Managerista luotiin uusi virtuaalikone, paikallisesta asennusmediasta. Virtuaalikoneelle annettiin 4096 megatavua keskusmuistia, neljä prosessoriydintä, sekä 25 gigatavua tallennustilaa ja vfio ohjauksella isäntäkoneen toinen näytönohjain, Nvidia GTX 1660ti MaxQ. Virtuaalikoneen käyttöjärjestelmän asennus suoritettiin graafisen käyttöliittymän kautta asennusohjelmiston antamalla vakioasetuksilla.



Kuvio 16. Virtuaalikoneessa ajettavan käyttöjärjestelmä asennusohjelmisto käynnistymässä

4.2.2 Asiakaskoneen asentaminen ja konfigurointi

Käyttöjärjestelmän asennus ja konfigurointi

Raspberry Pi 4:ään asennettiin Ubuntu Server 20.10 samoista syistä, kun isäntäkoneeseen. Asennusprosessin loppu on hyvin samantyyppinen kuin luvussa 4.2.1. mutta pieniä eroavaisuuksia löytyy.

Raspberry Pi 4 käynnistetään sd-kortilta Raspberry Pi:n mukana tulevaan käyttöjärjestelmään, ja päivitetään komentosarjalla *"sudo apt update && sudo apt full-upgrade -y && sudo rpi-update."* Komentosarja päivittää käyttöjärjestelmään uusimmat paketit ja päivittää käyttöjärjestelmän kernein sekä näytönohjaimen firmwären. Seuraavaksi pitää suorittaa komento *"sudo rpi-eeprom-update -d -a"*, joka asentaa uusimman käynnistyksenlataajan (bootloader). Uusin käynnistyksenlataaja tarvitaan USB-asemalta käynnistämiseksi. USB-käynnistyksen päälle laittamiseksi pitää

käynnistää raspi-config ohjelma, komennolla *"sudo raspi-config"* ja valita asetuksista käynnistysjärjestyksesi "B1 USB Boot".

Tämän jälkeen luodaan mahdollisimman nopealle USB-asetalle levykuva Ubuntu Server 20.10 käyttöjärjestelmästä. Tämä tapahtuu käyttäen komentoa *"sudo dd if=X of=Y bs=1M status=progress"* missä X on polku levykuvaan, prototyyppijärjestelmän tapauksessa *"ubuntu-20.10-preinstalled-desktop-arm64+raspi.img"* ja Y on käytettävän USB-aseman laitetunnus. Prototyyppijärjestelmän tapauksessa */dev/sdb*. Näiden toimien jälkeen asennus tehtiin luvussa 4.2.1 kuvatulla tavalla.

Raspberry Pi:ssä ei ole BIOS tai UEFI järjestelmää kuten x86 pohjaisissa tietokoneissa, tämän takia asetusmuutokset tehdään *"/boot/firmware/config.txt"* tiedostoon. Config.txt tiedostosta lisättiin grafiikkamuistin määräksi 512 megatavua. Muistin lisäys parantaa grafiikkapiirin suorituskykyä niissä tilanteissa, joissa käytettävien näyttöjen resoluutio on suuri. (Memory options in config.txt, N.d.; Config.txt, N.d.).

Tarvittavien ohjelmistojen asennus

Asiakastietokoneeseen asennettiin Weston ja Waypipe. Asennusprosessi ja käyttöönotto kuvataan luvussa 4.2.1.

4.2.3 Virtualisoituun etätyöasemaan yhdistäminen

Etätyöasemaan yhdistäminen aloitetaan hakemalla isäntäkoneen IP-osoite. IP-osoitteen saa Ubuntu Server 20.10 käyttöjärjestelmässä komennolla *"ip a"*. Prototyyppijärjestelmän tapauksessa isäntäkoneen IP-osoite on 10.0.0.33. Kun IP-osoite on selvitetty, täytyy varmistaa, että virtuaalikone on päällä ja selvittää virtuaalikoneen identifikaattorinumero. Tämän voi tehdä komennolla *"virsh list -all"*. Komento tulostaa luotujen virtuaalikoneiden tiedot. Listatuista tiedoista selviää virtuaalikoneen identifikaattorinumero, nimi sekä tila.

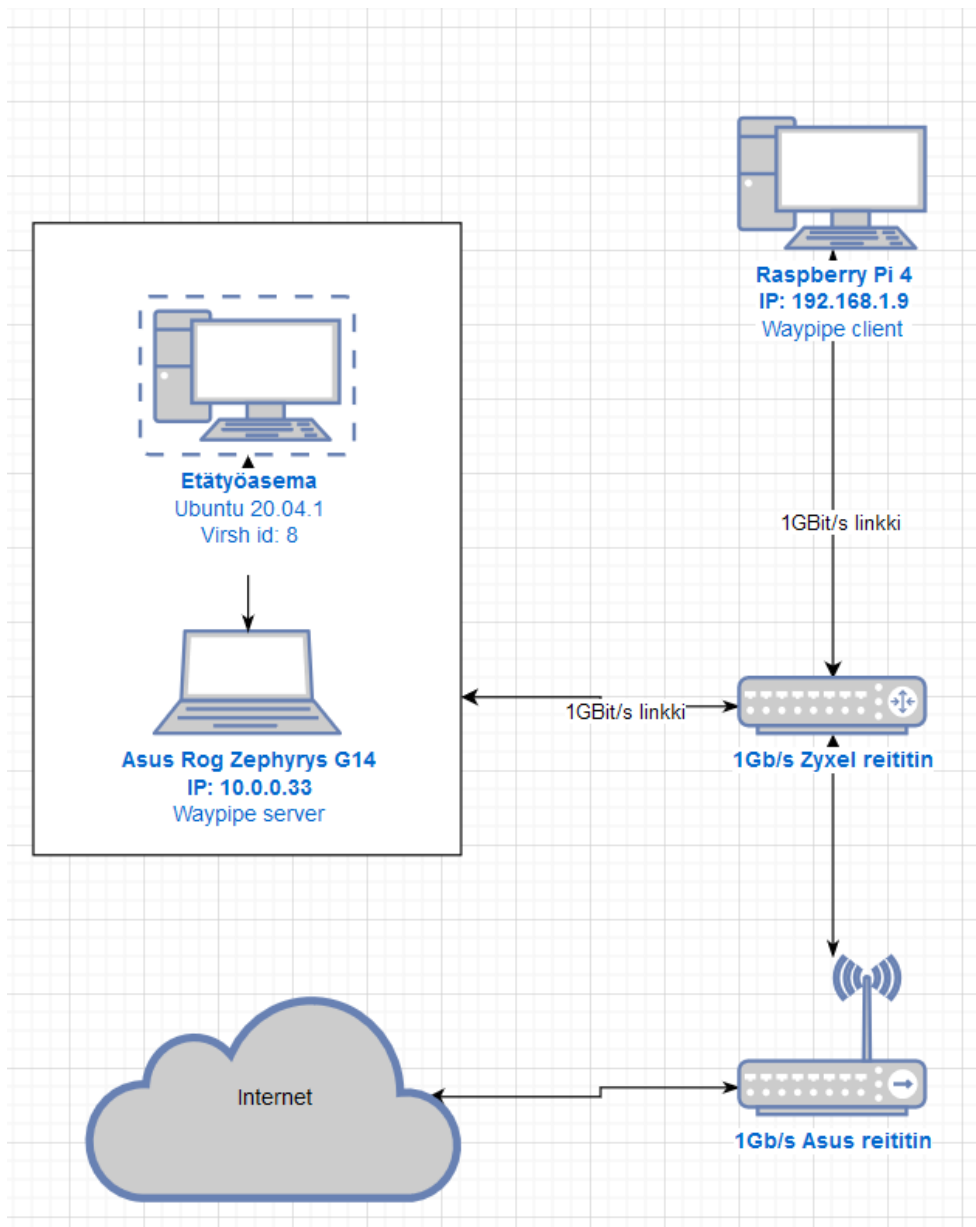
Waypipe tukee videon pakkausta ja siitä löytyy suorituskykytesti, jonka avulla voidaan selvittää mitä videonpakkausta kannattaa käyttää. Suorituskykytesti pitää ajaa työasemalla, johon yhdistetään, prototyypin tapauksessa isäntäkoneella.

Suorituskykytesti ajetaan komennolla *"waypipe bench B --threads N"*, jossa B on toivottu bittivirran kaistankäyttö sekunnissa megabiteissa ja N pakkaukseen käytettävien prosessorisäikeiden määrä. Prototyypin tapauksessa päädyttiin B=125, koska prototyyppijärjestelmän käytössä on 1Gbit/s yhteys, ja N=4, koska käytössä on nopea 16 säikeinen prosessori. Suorituskykytesti suosittelee, että isäntäkoneella käytettäisiin zstd pakkausta, arvolla 2. (README.md, 2020).

Yhdistäminen etätyöasemaan asiakaskoneelta tapahtuu komennolla: *"waypipe --threads 4 --compress zstd=2 ssh super@10.0.0.33 env LIBVIRT_DEFAULT_URI=qemu:///system DISPLAY=:0 virt-viewer -a --id 8"*

Komento paloihin pilkottuna:

- **waypipe**: käynnistää Waypipe asiakasohjelman
- **--threads 4**: määrittää pakkauksen käyttöön isäntäkoneella neljä säiettä
- **--compress zstd=2** käyttää videon pakkaamiseen zstd algoritmia pakkausarvolla 2
- **ssh super@10.0.0.33**: ottaa ssh protokollan yli yhteyden IP osoitteeseen 10.0.0.33 käyttäjänä super
- **env LIBVIRT_DEFAULT_URI=qemu:///system**: asettaa ympäristömuuttujan LIBVIRT_DEFAULT_URI arvoon qemu:///system
- **DISPLAY=:0**: käyttää näyttöä 0
- **virt-viewer -a --id 8**: käynnistää ja ohjaa virt-viewer ohjelmiston, identifikaattorilla 8 (halutun virtuaalityöaseman id), piirrettäväksi Waypipe asiakasohjelmistossa



Kuvio 17. Prototyypin yksinkertaistettu arkkitehtuuri

4.3 Prototyypin toiminnan validointi

Prototyyppiä testattiin 1920x1080 resoluutiolla ja 60Hz:n virkistystaajuudella. Käytössä oli langaton hiiri ja näppäimistö. Järjestelmää testattiin katsomalla YouTubea Karpo Official kanavalta ”Verottajan sauna” videota Firefox selaimella, tekstidokumentin kirjoittamisella LibreOffice Writer ohjelmistoa käyttäen sekä terminaalityökentelyllä käyttäen Ubuntun Terminal sovellusta. Huomion arvoista on, että hiiren

kursori piirretään edelleen asiakaskoneella, joten viivettä sen liikuttelussa ei ole normaalia syöttöviivettä enempää.

Prototyyppijärjestelmän ollessa päällä niin että vierastyöasema on toimettona työpöydällä, isäntäkoneen sekä asiakaskoneen resurssienkäyttö on hyvin pientä. Verkon käyttöaste on myös matala. (ks. Kuviot 18, 19, 20). Verkonkäytön graafeja tulkittaessa kannattaa huomioida, että liikenne, joka kulkee isäntäkoneelta asiakaskoneelle, mitataan "outgoing" kohdasta.

```

super@zephyrus: ~
1  [ 1.3%] 5 [||] 2.0%] 9 [ 0.7%] 13 [ 0.0%]
2  [ 0.0%] 6 [ 0.0%] 10 [ 0.7%] 14 [ 0.7%]
3  [ 6.1%] 7 [||] 7.3%] 11 [||] 15 [||] 1.3%]
4  [ 2.0%] 8 [ 0.0%] 12 [ 0.0%] 16 [ 0.0%]
Mem [|||||] 8.92G/15.1G
Swp [ 0K/4.00G]
Tasks: 54, 161 thr; 2 running
Load average: 0.66 0.48 0.21
Uptime: 00:08:25

  PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
 1343 super      20   0 1048M 139M 49068 R 10.0 0.9   0:03.49 /usr/bin/python3 /usr/share/virt-manage
1309 super      20   0 1830M 85972 59584 S  7.3 0.5   0:05.30 weston
1408 libvirt-q  20   0 12.2G 8389M 51148 S  2.0 54.4 0:53.61 /usr/bin/qemu-system-x86_64 -name guest
499  root       19  -1 123M 58436 57188 S  2.0 0.4   0:02.84 /lib/systemd/systemd-journald
1641 super      20   0 37236 21520 13824 S  2.0 0.1   0:00.68 /usr/bin/weston-terminal
875  syslog    20   0 215M 5340 4032 S  0.7 0.0   0:00.99 /usr/sbin/rsyslogd -n -iNONE
1640 super      20   0 8296 4360 3384 R  0.7 0.0   0:00.31 htop
1632 super      20   0 35152 19600 13380 S  0.7 0.1   0:00.53 /usr/bin/weston-terminal
1335 super      20   0 39536 23724 13424 S  0.7 0.2   0:01.60 /usr/bin/weston-terminal
1028 root        20   0 1371M 32904 21888 S  0.7 0.2   0:00.75 /usr/sbin/libvirtd
1445 libvirt-q  20   0 12.2G 8389M 51148 S  0.7 54.4 0:11.65 /usr/bin/qemu-system-x86_64 -name guest
1310 super      20   0 1830M 85972 59584 S  0.7 0.5   0:00.36 weston
913  syslog    20   0 215M 5340 4032 S  0.0 0.0   0:00.56 /usr/sbin/rsyslogd -n -iNONE
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice - F8Nice + F9Kill F10Quit

```

Kuvio 18. Isäntäkoneen resurssien käyttö, kun vieraskoneella ei tehdä mitään

```

ubuntu@ubuntu: ~
1  [ 0.0%]
2  [ 0.0%]
3  [ 1.3%]
4  [ 0.0%]
Mem [|||||] 278M/1.63G
Swp [ 0K/0K]
Tasks: 39, 21 thr; 1 running
Load average: 0.58 0.44 0.19
Uptime: 00:03:34

  PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
1921 ubuntu    20   0 9868 3464 2556 R  2.0 0.2   0:00.92 htop
1555 root       20   0 233M 7012 6080 S  0.0 0.4   0:00.02 /usr/lib/accountsservice/accounts-daemon
1811 ubuntu    20   0 143M 73792 54516 S  0.0 4.3   0:01.99 weston
1839 ubuntu    20   0 15368 7068 5240 S  0.0 0.4   0:00.59 /usr/bin/ssh -R /tmp/waypipe-server-dMVG
1840 ubuntu    20   0 441M 55372 37180 S  0.0 3.2   0:00.75 waypipe --threads 4 --compress zstd=2 ss
1551 root       20   0 81828 1744 1520 S  0.0 0.1   0:00.06 /usr/sbin/irqbalance --foreground
1913 ubuntu    20   0 16556 4296 3048 S  0.0 0.3   0:00.13 sshd: ubuntu@pts/1
  1  root       20   0 164M 10868 7504 S  0.0 0.6   0:03.24 /sbin/init fixrtc splash
883  root       19  -1 52208 14596 13504 S  0.0 0.9   0:00.49 /lib/systemd/systemd-journald
904  root       20   0 21312 5140 3588 S  0.0 0.3   0:01.11 /lib/systemd/systemd-udev
919  systemd-n 20   0 27592 6500 5556 S  0.0 0.4   0:00.15 /lib/systemd/systemd-networkd
1514 root       RT   0 210M 17524 7072 S  0.0 1.0   0:00.00 /sbin/multipathd -d -s
1515 root       RT   0 210M 17524 7072 S  0.0 1.0   0:00.00 /sbin/multipathd -d -s
1516 root       RT   0 210M 17524 7072 S  0.0 1.0   0:00.00 /sbin/multipathd -d -s
1517 root       RT   0 210M 17524 7072 S  0.0 1.0   0:00.01 /sbin/multipathd -d -s
1518 root       RT   0 210M 17524 7072 S  0.0 1.0   0:00.00 /sbin/multipathd -d -s
1519 root       RT   0 210M 17524 7072 S  0.0 1.0   0:00.00 /sbin/multipathd -d -s
1513 root       RT   0 210M 17524 7072 S  0.0 1.0   0:00.32 /sbin/multipathd -d -s
1542 systemd-r 20   0 25104 11364 7360 S  0.0 0.7   0:00.20 /lib/systemd/systemd-resolved
1676 systemd-t 20   0 91140 6668 5760 S  0.0 0.4   0:00.00 /lib/systemd/systemd-timesyncd
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice - F8Nice + F9Kill F10Quit

```

Kuvio 19. Asiakaskoneen resurssien käyttö, kun vieraskoneella ei tehdä mitään

```

super@zephyrus: ~
Device enx002427880386 [10.0.0.34] (1/6):
=====
Incoming:

                                     Curr: 2.05 kBit/s
                                     Avg: 12.11 kBit/s
                                     Min: 0.00 Bit/s
                                     Max: 123.41 kBit/s
                                     Ttl: 467.94 kByte

Outgoing:

                                     Curr: 1.02 kBit/s
                                     Avg: 40.03 kBit/s
                                     Min: 0.00 Bit/s
                                     Max: 864.28 kBit/s
                                     Ttl: 13.13 MByte
|#|

```

Kuvio 20. Verkonkäyttö, kun vieraskoneella ei tehdä mitään

YouTube-videon toistaminen

Videon kuvanlaatu on hyvä mutta siinä tuntuu olevan ihan pieniä ongelmia ruudunpäivitysnopeudessa. Jos videosoitinta käytetään koko ruudun kokoisena (1920x1080 resoluutio) kasvaa huomattavasti useaan sataan megabittiin sekunnissa (ks. Kuvio 23). Suoritinaikaa kuluu isäntäkoneella normaalitasoa enemmän (ks. Kuvio 21), mutta se selittyy Waypipen tekemällä pakkauksella, ja sillä että isäntäkoneen prosessori de-koodaa YouTube -palvelusta katsotun videon. Kuvio 22. näkee kuinka asiakaskoneena käytetyn Raspberry Pi 4:n suoritinkäyttö on jo suhteellisen korkealla tasolla.

```

super@zephyrus: ~
1  [|||||] 27.6%] 5  [|||||] 24.8%] 9  [|||||] 20.7%] 13  [|||||] 13.7%]
2  [|||||] 16.0%] 6  [|||||] 20.5%] 10 [|||||] 15.4%] 14  [|||||] 17.2%]
3  [|||||] 23.3%] 7  [|||||] 27.9%] 11 [|||||] 17.2%] 15  [|||||] 25.0%]
4  [|||||] 24.2%] 8  [|||||] 18.8%] 12 [|||||] 16.2%] 16  [|||||] 18.0%]
Mem[|||||] 8.95G/15.1G] Tasks: 54, 166 thr; 5 running]
Swp[|||||] 0K/4.00G] Load average: 1.30 0.73 0.32]
Uptime: 00:09:48

  PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
 1408 libvirt-q 20   0 12.2G 8387M 51148 R 229. 54.4 2:17.36 /usr/bin/qemu-system-x86_64 -name guest
1582 super      20   0 2072M 64928 37856 S 72.0  0.4 0:06.72 waypipe -c zstd=2 --threads 4 --unlink-
1448 libvirt-q 20   0 12.2G 8387M 51148 S 29.6 54.4 0:14.59 /usr/bin/qemu-system-x86_64 -name guest
1445 libvirt-q 20   0 12.2G 8387M 51148 S 28.9 54.4 0:21.12 /usr/bin/qemu-system-x86_64 -name guest
1452 libvirt-q 20   0 12.2G 8387M 51148 S 27.6 54.4 0:15.23 /usr/bin/qemu-system-x86_64 -name guest
1451 libvirt-q 20   0 12.2G 8387M 51148 S 26.9 54.4 0:13.82 /usr/bin/qemu-system-x86_64 -name guest
1447 libvirt-q 20   0 12.2G 8387M 51148 S 26.3 54.4 0:16.35 /usr/bin/qemu-system-x86_64 -name guest
1450 libvirt-q 20   0 12.2G 8387M 51148 R 26.3 54.4 0:14.07 /usr/bin/qemu-system-x86_64 -name guest
1449 libvirt-q 20   0 12.2G 8387M 51148 S 25.6 54.4 0:13.93 /usr/bin/qemu-system-x86_64 -name guest
1446 libvirt-q 20   0 12.2G 8387M 51148 S 25.6 54.4 0:15.46 /usr/bin/qemu-system-x86_64 -name guest
1584 super      20   0 2072M 64928 37856 S 18.2  0.4 0:01.44 waypipe -c zstd=2 --threads 4 --unlink-
1583 super      20   0 2072M 64928 37856 S 17.5  0.4 0:01.43 waypipe -c zstd=2 --threads 4 --unlink-
1585 super      20   0 2072M 64928 37856 S 16.8  0.4 0:01.42 waypipe -c zstd=2 --threads 4 --unlink-
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice - F8Nice + F9Kill F10Quit

```

Kuvio 21. Isäntäkoneen resurssien käyttö videokatselun aikana

toimiville yrityksille. Lähinnä kuitenkin työpaikoille tai käyttötarkoituksiin, jossa ei käsitellä paljon liikkuvaa kuvaa tai videoita. Tämäntyyppiset käyttötarkoitukset käyttävät useita kymmeniä megabittejä kaistaa sekunnissa. Muutamissa koneissa ongelmia ei toki tulisi, vaan kaistan käyttö olisi ongelma, kun puhuttaisiin kymmenistä työasemista.

Kehitystyöstä syntynyttä prototyyppiä voi jatkokehittää useampaankin suuntaan. Jos tarkoituksena olisi kehittää virtuaalityöasemainfrastruktuuria tukeva järjestelmä, pitäisi lähteä suunnittelemaan virtuaalikoneiden käynnistyksen automatisointia, etätyöasemaan yhdistämisen helpottamista ja muita käyttäjää helpottavia toimintoja.

Johtuen useista kaupallisista kilpailijoista, joilla on jo valmis VDI järjestelmä, voi prototyypin kaupallistaminen olla haastavaa, ellei jopa mahdotonta. Pienenä toimijana markkinoille penetroituminen vaatii huomattavaa määrää ruohonjuuritason toimintaa ja tuotekehitystyötä, johon tässä vaiheessa ei opinnäytetyön kirjoittajalla riitä aika ja resurssit.

Validointiosassa selvisi, että prototyyppijärjestelmä käyttää joissain tilanteissa verrattain vähän kaistaa ottaen huomioon, että pakkaus on häviötön ja yleisesti ottaen todella vähän prosessoriaikaa ollen näin resurssienkäytöltään tehokas. Valitettavasti verrokkijärjestelmiä ei päästy testaamaan tällä tasolla, koska ne ovat maksullisia tai haasteellisia testata.

5.2 Luotettavuus

Tutkimus- ja kehitystyötä on koitettu tehdä mahdollisimman systemaattisesti ja analyyttisesti, mutta esille nousi kolme suurehkoa luotettavuuteen vaikuttavaa seikkaa.

Koska työn tekijällä ei ollut resursseja testata järjestelmää useammalla asiakaskoneella tai isäntäkoneella, jossa on Intelin prosessori, ovat tutkimustulokset epäluotettavia. Myöskään Windows -virtuaalikonetta ei testattu, koska Microsoftin

lisenssiehdot ovat hieman epäselviä sen laillisuuden suhteen. Tämä johtuu siitä, että samat asetukset ja ohjelmistot eivät välttämättä toimi jokaisella tietokoneella samalla tavalla ja suorituskyky voi heitellä paljonkin.

Luotettavuutta olisi lisännyt, jos prototyypissä olisi mitattu niin sanottua ”klikkauksesta fotoniiin” menevää aikaa. Termi tarkoittaa aikaa, joka menee hiiren klikkauksesta siihen, että näyttölaitteelle ilmestyy kuva. Tekijältä ei kuitenkaan löytynyt laitteistoa tai resursseja hankkia niitä, joten testi jätettiin tekemättä.

Luotettavuutta heikentää myös käyttäjätestauksen puute. Järjestelmää olisi voinut testata verrokkijärjestelmiä vastaan tietokonetta töissään käyttävillä henkilöillä. Verrokkijärjestelminä olisi voinut toimia kaupalliset etätyöasemavirtualisointiratkaisut ja normaali virtualisoimaton työasema.

Lähteet

Acharya, S. 2019. What is VDI And How Does It Work? Artikkele verkkosivustolla. Viitattu 10.11.2020 <https://www.acecloudhosting.com/blog/what-is-vdi-how-it-work/>

Adams, J. 2016, Top 5 Benefits of Desktop Virtualization. Blogikirjoitus verkkosivustolla. Viitattu 7.11.2020 <https://blog.whitehatvirtual.com/top-5-benefits-of-desktop-virtualization>

Amazon Workspaces. N.d. Artikkele verkkosivustolla. Viitattu 4.11.2020. <https://aws.amazon.com/workspaces/>

Beal, V. N.d. Desktop virtualization. Artikkele verkkosivustolla. Viitattu 13.10.2020, https://www.webopedia.com/TERM/D/desktop_virtualization.html

Bister, T. 2019. Tietojenkäsittelyn opinnäytetyö. Jyväskylä: Jyväskylän ammattikorkeakoulu.

Chiramal, H. D., Mukhedkar, P & Vettathu, A. 2016, Mastering KVM Virtualization Birmingham: Packt PublishingLinux Bible, 9th Edition

config.txt, N.d. Dokumentaatio verkkosivustolla. Viitattu 8.11.2020 <https://www.raspberrypi.org/documentation/configuration/config-txt/>

Coutinho, N., Schaapman, P., Temin, T., 2010, Virtualization and Infrastructure Optimizationreference guide. CDW Government LLC

Desktop and App Virtualization. N.d. Tuotesivusto. Viitattu 8.11.2020. <https://www.vmware.com/fi/products/desktop-virtualization.html>

Gaille, L. 2017, 14 Advantages and Disadvantages of Virtualization. Artikkele verkkosivustolla. Viitattu 2.11.2020 <https://vittana.org/14-advantages-and-disadvantages-of-virtualization>

Golden, B. 2009. Virtualization for Dummies, 2nd edition. Indianapolis: Wiley Publishing, Inc.

Ivanov, K. 2017, KVM Virtualization Cookbook, Birmingham: Packt Publishing

Kananen, J. 2012. Kehittämistutkimus opinnäytetyönä. Jyväskylä: Jyväskylän ammattikorkeakoulu

Karl, A. 2016, Pros and cons of server virtualization. Artikkeliverkkosivustolla. Viitattu 2.11.2020 <http://techgenix.com/pros-cons-server-virtualization/>

Kelly, W. 2017, 4 Ways Desktop Virtualization Improves Employee Efficiency. Artikkeliverkkosivustolla. Viitattu 7.11.2020. <https://insights.samsung.com/2017/10/24/4-ways-desktop-virtualization-improves-employee-efficiency/>

Madden, B. 2018, Is VDI dead? Blogikirjoitus verkkosivustolla. Viitattu 7.11.2020 <https://techzone.vmware.com/blog/vdi-dead>

Memory options in config.txt, N.d. Dokumentaatio verkkosivustolla. Viitattu 8.11.2020 <https://www.raspberrypi.org/documentation/configuration/config-txt/memory.md>

Microsoft 365 Business Premium, N.d. Tuotesivu verkkokaupassa. <https://www.microsoft.com/en-us/microsoft-365/business/microsoft-365-business-premium?activetab=pivot%3aoverviewtab>

Piltch, A. 2020, Raspberry Pi 4: Review, Buying Guide and How to Use. Arvostelu tuotteesta verkkosivustolla. Viitattu 4.11.2020 <https://www.tomshardware.com/reviews/raspberry-pi-4>

Portier, J. 2020, Vagrant with the libvirt plugin. Blogikirjoitus verkkosivustolla. Viitattu 9.11.2020 <https://www.moonstreet.nl/posts/libvirt/>

Portnoy, M. 2016, Practical LXC and LXD: Linux Containers for Virtualization and Orchestration

Portnoy, M. 2016, Virtualization Essentials, p.2, Indianapolis: John Wiley & Sons

README.md, 2020, "Lue minut" -tiedosto Wayland projektin git repositoryssa. Viitattu 7.11.2020. <https://github.com/wayland-project/weston/blob/master/README.md>

README.md, N.d., "Lue minut" – tiedosto Waypipe projektin git repositoryssa. Viitattu 8.11.2020. <https://gitlab.freedesktop.org/mstoeckl/waypipe/-/blob/master/README.md>

Red Hat Enterprise Virtualization. N.d. "Usein kysytyt kysymykset" -dokumentti verkkosivustolla. Viitattu 10.11.2020 <https://www.redhat.com/cms/managed-files/rh-enterprise-virtualization-faq-inc0357243mm-201604-en.pdf>

Red Hat Virtualization. N.d. Tuotesivu verkkosivustolla. Viitattu 10.11.2020 <https://www.redhat.com/en/technologies/virtualization/enterprise-virtualization>

Rufus. N.d., verkkosivusto. Viitattu 7.11.2020. <https://rufus.ie/>

Schimscheimer, F. N.d. Server and Storage Sizing Guide for Windows 7 Desktops in a Virtual Desktop infrastructure. Tekninen dokumentaatio verkkosivustolla. <https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/technicalnote/view/server-storage-sizing-guide-windows-7-technical-note.pdf>

Stoeckl, M. 2019, Network transparency with Wayland. Blogikirjoitus-sarja verkkosivustolla. Viitattu 8.11.2020 <https://mstoeckl.com/notes/gsoc/blog.html>

Strickland, J. 2008. Artikkelit verkkosivustolla. Viitattu 17.10.2020, <https://computer.howstuffworks.com/server-virtualization.htm>

Sutela, H. 2020, Kun mahdoton kävi mahdolliseksi – tietotyön yleisyys mahdollisti etätöiden läpimurron Suomessa. Blogikirjoitus verkkosivustolla. Viitattu 4.11.2020. <http://www.stat.fi/tietotrendit/blogit/2020/kun-mahdoton-kavi-mahdolliseksi-tietotyon-yleisyys-mahdollisti-etatyon-lapimurron-suomessa/>

Taylor, K. N.d. Pros and Cons of Virtualization. Artikkelit verkkosivustolla. Viitattu 2.11.2020 <https://www.hitechnectar.com/blogs/pros-cons-virtualization/>

Top 10 key benefits of Desktop Virtualization. N.d. Artikkelit verkkosivustolla. Viitattu 7.11.2020 <https://www.virtualizewithus.com/top-10-key-benefits-of-desktop-virtualization/>

Ubuntu 20.10 on Raspberry Pi delivers the full Linux desktop and micro clouds. 2020. Artikkelit verkkosivustolla. Viitattu 7.11.2020 <https://ubuntu.com/blog/ubuntu-20-10-on-raspberry-pi-delivers-the-full-linux-desktop-and-micro-clouds>

What is a hypervisor? N.d. Dokumentaatio verkkosivustolla. Viitattu 10.11.2020 <https://www.redhat.com/en/topics/virtualization/what-is-a-hypervisor>

What is a Thin Client? N.d. Artikkelit verkkosivustolla. Viitattu 10.11.2020 <https://www.stratodesk.com/what-is-a-thin-client/>

What is Citrix virtualization? N.d. Artikkele verkkosivustolla. Viitattu 13.10.2020, <https://www.citrix.com/solutions/app-and-desktop-virtualization/>

What is Desktop as a Service (DaaS)? N.d. Artikkele verkkosivustolla. Viitattu 6.11.2020 <https://www.citrix.com/glossary/what-is-desktop-as-a-service-daas.html>

What is Desktop Virtualization? N.d. Artikkele verkkosivustolla. Viitattu 13.10.2020, <https://www.anuntatech.com/about-us/learning-center/what-is-desktop-virtualization/>

What is free software? 2019. Artikkele verkkosivustolla. Viitattu 4.11.2020. <http://www.gnu.org/philosophy/free-sw.html>

What is Windows Virtual Desktop? 2020. Dokumentaatiota verkkosivustolla. Viitattu 10.11.2020 <https://docs.microsoft.com/en-us/azure/virtual-desktop/overview>

Whitney, L. 2020. How to choose the best desktop as a service (DaaS) vendor for your business. Artikkele verkkosivustolla. Viitattu 2.11.2020. <https://www.techrepublic.com/article/how-to-choose-the-best-desktop-as-a-service-daas-vendor-for-your-business/>

Whitney, L. 2020. Top reasons why businesses should deploy Desktop as a Service (DaaS). Artikkele verkkosivustolla. Viitattu 2.11.2020. <https://www.techrepublic.com/article/top-reasons-why-businesses-should-deploy-desktop-as-a-service-daas/>

Vigliarolo, B. 2020, VDI vs. DaaS: What is the difference, and which is best for your virtualization needs? Artikkele verkkosivustolla. Viitattu 2.11.2020 <https://www.techrepublic.com/article/vdi-vs-daas-what-is-the-difference-and-which-is-best-for-your-business-virtualization-needs/>

VMWare Horizon Standard. N.d. Tuotesivu verkkokaupassa. Viitattu 10.11.2020. <https://store-us.vmware.com/vmware-horizon-standard-299528700.html>