

Joel Haapalainen

PIENTEN JA KESKISUURTEN YRITYSTEN

TEHOKAS PALVELINRATKAISU

PIENTEN JA KESKISUURTEN YRITYSTEN

TEHOKAS PALVELINRATKAISU

Joel Haapalainen
Opinnäytetyö
Syksy 2015
Hyvinvointitekniikan koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Hyvinvointitekniikan koulutusohjelma

Tekijä: Joel Haapalainen

Opinnäytetyön nimi: Pienten ja keskisuurten yritysten tehokas palvelinratkaisu

Työn ohjaaja(t): Jukka Jauhiainen

Työn valmistumislukukausi ja -vuosi: syksy 2015

Sivumäärä: 47 + 1 liite

Opinnäytetyön aiheena oli tehdä tutkimusta erilaisten palvelinratkaisujen välillä. Lisäksi työssä arvioitiin palvelinratkaisujen sopivuutta toimeksiantajan käyttöön, mutta myös laajemmin pienten ja keskisuurten yritysten käyttöön.

Tutkimuksessa kerättiin tietoa olemassa olevista palvelinratkaisuista ja palvelinjärjestelmien testaamisesta. Kokeellisia palvelinjärjestelmiä testattiin aikarajojen puitteissa mahdollisimman monta, jotta löydettäisiin paras mahdollinen palvelinratkaisu toimeksiantajan tarpeisiin. Tutkimukseen kuului pohtia pienten ja keskisuurten yritysten tarpeita palvelimen toteuttamiseksi ja menetelmiä palvelinratkaisun tehokkuuden määrittämiseksi.

Erilaisia palvelinratkaisuja vertailtiin virtuaalisella ja fyysisellä tasolla. Toimeksiantajalle osoittautui parhaimmaksi ratkaisuksi Microsoft Azure. Pienten ja keskisuurten yritysten palvelin vaihtoehtoratkaisuja valitessa ei voida päätyä mihinkään erityiseen ratkaisuun, sillä yrityksissä on erilaiset tarve- ja resurssivalmiuserot. Työn lopuksi pohditaan palvelintekniikan kehittymistä ja mahdollisia tulevaisuuden näkymiä.

Asiasanat: palvelimet, pilvipalvelimet, verkkopalvelut, virtuaalipalvelimet, palvelinratkaisut

ABSTRACT

Oulu University of Applied Sciences
Medical engineering

Author: Joel Haapalainen

Title of thesis: Efficient server solution for small and medium sized corporations

Supervisor(s): Jukka Jauhianen

Term and year when the thesis was submitted: Autumn 2015

Pages: 47 + 1 appendices

Group Builder is a fast growing startup ICT company in computer aided designing for apartments and households. Their old server plan wasn't just quite what their demands were and there was need for new server solution. Thesis assignment was to research in different kind of server solutions and find the most equivalent one for company's use. Company's old server was Oulu DC's dedicated virtual server solution with two running virtual machines on VMware. These virtual machines had Windows server and Debian operating systems and Glassfish application server using Java EE.

To find this best suited server for company's use it was to first find out all company's services run by old server and possible future services. Then it was to acquiring knowledge of different systems and their support on software frameworks and applications. If there wasn't straight solution, it was needed to search optional ways to make server solution work. After there was working solution ideas it was to set up test environments and running forensic tests on them. Evaluating needs of company and analyzing operating cost efficiency.

During thesis work was tested Cloud computing systems and different kind server solution from traditional servers to virtual machine servers. Equipment on these test were old laptops on company's network and dedicated data centers from wide variety service providers. Thesis worker was quite new to using servers, but had good knowledge of computer environments and information transfer technologies. Thesis result was to build Microsoft Azure cloud service for company's use with Bizpark account deal. So the company would have three years contract with 750\$ worth of monthly usage in Azure environment.

It is hoped that this research made for this thesis work is exploitable also for other companies to find the most efficient server solution for their needs. Also for single individuals to find keen interest from server side and help them understand how back-end of the web works.

Keywords: server, dedicated server, virtual server, cloud computing, web services, server solution

ALKULAUSE

Kiitän työni toimeksiantajaa Group Builderia mielenkiintoisen aiheen antamisesta ja opinnäytetyöni toteuttamisen mahdollistamisesta. Koko työskentelyn ajan työilmapiiri yrityksessä oli hyvä ja edellytykset vapaaseen tutkimiseen olivat otolliset. Sain sopivasti vastuuta ja odotuksia työni tekemiselle. Työskentelyvapaus ja joustavuus tukivat luovuutta ja työni hyvää edistymistä. Palvelimet olivat minulle aiheena uusi aihe, mikä toi tekemiseeni haastetta, mutta myös mielenkiintoa.

Jukka Jauhiaista kiitän työni ohjauksesta ja tukemisesta työskentelyn eri vaiheissa. Haluan kiittää myös kaikkia niitä ihmisiä, joita olen voinut konsultoida työtäni koskevissa asioissa.

Haluan erityisesti kiittää perhettäni ja ystäviäni kannustavasta asennoitumisesta opinnäytetyöhöni. Tukenne loi minuun uskoa, että selviän hyvin tästä työstä.

Oulussa 17.12.2015

Joel Haapalainen

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
ALKULAUSE	5
SISÄLLYS	6
SANASTO	8
1 JOHDANTO	9
1.1 Työn tarkoitus	9
1.2 Työn tilaaja	10
2 PALVELIN	11
2.1 Standardit ja protokollat palvelinten ympärillä	11
2.2 Palvelimen fyysinen rakenne	13
2.3 Palvelintietokoneen osat	14
2.3.1 CPU	14
2.3.2 RAM ja ROM	15
2.3.3 Massamuisti	17
2.3.4 Oheislaitteet	18
2.4 Virtuaalinen taso	19
2.5 Ekologinen ajattelumalli	20
3 YRITYKSIEN TARVE TOTEUTTAA PALVELUITA	22
3.1 Yleiset tarpeet palvelinympäristön toteutukseen	22
3.2 Pienten ja keskisuurten yritysten tarve erot	23
3.3 Vaihtoehtoiset ratkaisut palvelinympäristölle	24
4 TURVALLISUUS JA TEHOKKUUS	26
4.1 Turvallisuuden ongelmat ja niiden mahdolliset ratkaisut	27
4.2 Kustannustehokkuus	28
5 PALVELINJÄRJESTELMÄT JA OHJELMISTOT	31
5.1 Palvelinjärjestelmät	31
5.2 Virtuaalipalvelin järjestelmät	32
6 TYÖN TOTEUTUS	36
6.1 Lähtötilanne	36
6.2 Kokeilut eri palvelinvaihtoehtoista	37

6.3 Palvelintoteutus Azureen	39
7 PALVELINKEHITYKSEN TULEVAISUUS	40
8 POHDINTA	42
LÄHTEET	44

SANASTO

IO	Input Output. Sisään tuleva ja lähtevä data
Data	Signaalia, joka sisältää digitaalisessa muodossa kutsutaan tietotekniikassa dataksi
CPU	Central Processing Unit on suoritin palvelimissa ja tietokoneissa
RAM	Konejärjestelmän käyttömuisti "Random Access Memory
RAID	"Redundant Array of Independent Disks" on tekniikka, jolla yhdistetään kaksi tai useampi kiintolevyä yhdeksi loogiseksi levyksi
BIOS	"Basic Input-Output System" on tietokoneissa käynnistysohjelmisto
UEFI	Unified Extensible Firmware Interface on standardi, joka määrittelee rajapinnan laiteohjelmiston ja käyttöjärjestelmän välillä.
DNS ja DDNS	Dynamic- ja Domain Name System lyhenne nimipalvelulle, joka hallinnoi verkkotunnuksia ja niiden IP-osoitevas-tineita
TCP/IP	Transmission Control Protocol / Internet Protocol on Internet-liikennöinnissä käytettävän tietoverkkoprotokolla
Kernel	Käyttöjärjestelmän ydin ja rajapinta fyysisen kokoonpanon sekä ohjelmistojen välillä
UPS	Uninterruptible Power Supply varavirtalaitteisto, jonka tehtävänä on sähköverkon yli- ja alijännitetilanteissa taata tasaista syöttöjännitettä

1 JOHDANTO

1.1 Työn tarkoitus

Työn tarkoitus oli toteuttaa Group Builder Oy:n tarpeisiin sopiva palvelinratkaisu. Palvelinratkaisu on toteutuskokonaisuus, jossa on valittu tarpeisiin sopiva konejärjestelmä ja palvelinohjelmisto. Koska esitettyyn työhön liittyi paljon palvelinratkaisujen vertailua ja tutkimusta, työssä päätettiin toteuttaa Group Builderille toimiva palvelinratkaisu. Lisäksi vertailtiin yleisesti palvelimia pienten ja keskisuurten yritysten tarpeisiin. Pienten ja keskisuurten yritysten tarpeet vaihtelevat hyvin laajasti ja siksi ei ole olemassa yhtä oikeata ratkaisua. Työssä esitellään ratkaisu, joka toteutettiin Group Builderille, mutta myös käsitellään yleisiä hyviä kokonaisvaihtoehtoja yritysten käyttöön. Lisäksi esitellään myös vertailevia arvioita toiminnollisuuksista ja analysoidaan tehokkuuksia eri tarve- ja tavoitekriteereillä.

Eri toimi- ja palvelualojen yritykset voivat riippumatta niiden palveluista tai tuotevalikoimasta käyttää samaa palvelinratkaisua. Palvelinratkaisun toteutukseen vaikuttaa palvelulle asetetut vaatimukset ja palvelujen toteutus. Palvelin ja palvelujen toteuttaminen ovat yhdenmukaistumassa ja nykyään yhä useampi ohjelmisto toimii ristiin erilaisissa järjestelmissä toimivien ohjelmistojen kanssa.

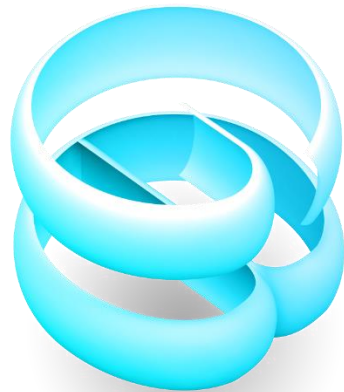
Palvelimen toteutusstrategia tulee toteuttaa lähtemällä liikkeelle palvelujen toteuttamiseen tehokkaimmaksi todetuilla metodeilla tai olemassa olevien palvelujen ehdoilla.

Työn toteuttamiseen käytettiin lähteinä palvelinkomponenttien ohjeita ja standardimääritelmiä. Jotta tutkimus osoittaisi, että myös muutkin pystyisivät tämän työn ja internetistä löytyvien ohjeiden perusteella toteuttamaan oman palvelinympäristönsä, on työ ja ohjeet tehty ilman ulkopuolista neuvomista. Työn haluttiin osoittavan, että nykyään riittävillä tiedoilla kuka vain voisi tehdä toimivan palvelinratkaisun omalle yritykselle tai jopa yksityiskäyttöön. Työn tarkoitus ei ole olla kuitenkaan suora ohje palvelinratkaisun toteuttamiseen, vaan vertailla ja arvioida erilaisten komponenttien toimivuutta sekä pohtia toimivia kokonaisuuksia yrityksen hypoteettisiin tarpeisiin.

1.2 Työn tilaaja

Group Builder on yritys, joka toteuttaa huoneistojen pohjakuvista 3D- mallin. Yritysasiakkaat käyttävät Group Builderin palveluja omien asiakkaidensa huoneistojen esittelyyn, hallinnoimaan viestittämistä asiakkaiden ja yrityksen välillä, ylläpitämään asiakasrekisteriä ja toteuttamaan huoneistojen rakentamismallit asiakkaiden toiveiden pohjalta. Group Builder on vielä tällä hetkellä pienyritys, mutta nopeasti kasvava yritys.

Asiakkaiden suuri tietoliikennekäyttö ja yrityksen toimintamalli tuottaa palvelut internetissä loivat hienon tilaisuuden järjestää tehokkaan palvelujärjestelmän yrityksen käyttöön. Kuvassa yksi on yrityksen logo, joka tehtiin työn ohessa vektorigrafiikka pohjaiseen SVG (Scalable Vector Graphics)-formaatin suunnittelijan mallia mukaillen.



GBuilder

KUVA 1. Group Builderille tehty logo.

2 PALVELIN

Palvelin on palveluja tuottava konejärjestelmä. Palvelinjärjestelmiä on usean kokoisia ja palvelurooiltaan vaihtelevia kokoonpanoja. Varsinaisesti konejärjestelmän tekee palvelimeksi sen palvelinohjelmisto ja sen palvelumalli. Palvelimia toteutetaan useisiin erilaisiin tehtäviin, joita voi hoitaa yksi tai useampi palvelin tai palvelinohjelmisto. Yleisimpiä palvelimien tehtäviä ovat sovellus-, nimi-, WWW (World Wide Web)-, sähköposti-, tiedosto-, tietokanta-, tulostus-, media- ja pelipalvelin. Palvelin on harvoin toteutettu vain yhtä tehtävää varten, vaan yleensä yksi palvelin perustetaan hoitamaan mahdollisimman montaa tehtävää. Palvelinohjelmistot pystyvät toteuttamaan samassa järjestelmässä erilaisia palvelurooleja, jos palvelin laitteiston puolesta siihen kykenee. Joskus kuitenkin joudutaan toteuttamaan välityspalvelun kautta päällekkäin menevät palvelut, kun ristiriitojen kanssa tulee ongelmia. Tällöin välityspalvelin tunnistaa pyynnön mukaan, mitä palveluohjelmaa tulee käyttää. Tämän jälkeen välityspalvelin ohjaa pyynnön oikealle ohjelmalle. Välityksen voi toteuttaa käyttämällä eri portteja tai virtualisoidulla sijainteja järjestelmän sisällä, jolloin ohjelmat voivat käyttää samoja portteja. Portti on internet-protokollan mukaan kohdennettu sijainti IP-osoitteen alla. (Dostálek – Kabelová 2006, 316–317.)

2.1 Standardit ja protokollat palvelinten ympärillä

The Internet Engineering Task Force (IETF) hallinnoi internettiin liittyviä standardeja. Internet-protokolla on IETF:n määrittämä tietoliikenneyhteys pakettikytkentäisten internet-verkkoon liitettyjen koneiden välillä. Internet-protokolla on siis koko internet-idean ydin ja kaikki koneet käyttävät sitä. Internet-protokollaa käytetään palvelujen mahdollistamiseksi verkon yli IP-avaruutta hyödyntäen. IP-avaruus tarkoittaa IPv4- ja IPv6-osoitteita, jotka osoitetaan kullekin verkkoon liittyvälle koneelle. IP-osoitetta voisi siis verrata postitusosoitteeseen, jonne paketit toimitetaan. Internet-protokolla koostuu kolmesta kerroksesta, jossa verkkokerros on alin, kuljetuskerros on välissä ja sovelluskerros ylin. Verkkokerros on siis internetissä IP-osoiteavaruus ja kuljetinkerroksen avulla tuodaan paketit asiakkaan sekä palvelimen välillä. Paketti on protokollan mukaan tietolähete, joka si-

sältää käytettävän kuljetusprotokollan mukaisesti aina järjestyksessä lähetysosoitteen, kohdeosoitteen, paketin koon ja paketin datan. (Dostálek ym. 2006, 14–20.)

Kuljetinkerroksessa pakettien siirtämiseen käytetään TCP (Transmission Control Protocol)-protokollaa ja UDP (User Datagram Protocol)-protokollaa. Protokollien ero on yhteyden luominen ja tiedon perille pääsyn varmistaminen. TCP:ssä yhteys toteutetaan kolmivaiheisesti. Kättelyn avulla luodaan yhteys asiakkaan sekä palvelimen välille, tiedon siirrossa varmistetaan tiedon perille pääsy järjestyksessä ja päätetään lopetus sekä kuittaus paketeilla kumpienkin osapuolten toimesta. UDP:n mukaan yhteyden luontia ei tarvita, vaan paketit lähetetään varmistamatta, saako vastaanottaja niitä. Tämä menetelmä on tehokkaampi, jos tiedon perille pääsyä ei tarvitse varmistaa ja yhteyden pitoa ei tarvita. Sovelluskerroksen protokollat määräävät pakettien datan muodon. Digitaalinen binääridata muunnetaan protokollien käyttämään muotoon, joka voidaan jatkokäsittellä protokollaa hyödyntävillä ohjelmilla. Sovelluskerroksen protokollia on paljon ja ne ovat tärkeimpiä internet-protokollia palvelinten palveluja suunniteltaessa. (Dostálek ym. 2006, 14–20.)

Palvelintoteutus on monitasoinen hierarkia, missä käyttöjärjestelmän ydin (kernel) toteuttaa rajapinnan laitteiston ja ylemmän tason ohjelmien välille. Kernelissä laitteistoja ohjataan ajureilla eli laitteisto-ohjelmilla. Laitteisto-ohjelmat antavat käyttökomennot käyttöjärjestelmille, joilla voidaan helposti toteuttaa laitteistolla tehtäviä. Kun laitteistoja halutaan käyttää ylemmän tason ohjelmilla, niin ohjelmien tarvitsee kutsua vain laitteistotyyppiä ja antaa tehtävä. Tämä helpottaa valtavasti ohjelmien kirjoittamista, kun laitteistoa ei varsinaisesti tarvitse tuntea. Sovellukset ovat tässä hierarkiassa ylimpänä ja palvelinpalveluohjelmistot sen alapuolella lähellä Kernel-tasoa. (Chevance 2004, 1–12.)

2.2 Palvelimen fyysinen rakenne

Kaikki palvelimet ovat aina, kun tarpeeksi pitkälle katsotaan, jonkinlaisia fyysisiä kokoonpanoja. Palvelimen fyysinen koko voi kuitenkin vaihdella suuresti. Suurimmillaan palvelimet ovat massiivisia laitoksia, jotka tarvitsevat hallien kaltaisia rakennuksia ympärilleen. Pienimmillään palvelin voi olla jopa USB-tikun kokoinen laite. Maailmassa on eniten palvelimia, jotka ovat kooltaan normaalin pöytäkoneen kokoluokkaa tai alle. Palvelinkeskukset ja suurtietokonejärjestelmät ovat useiden pienten palvelimien laskentakyvyn vertaisia ja usein tehokkaampia kokonaisuuksia. Tehokkuus käsitetään koneissa suorituskyvyn ja virrankulutuksen suhteesta. Palvelimien tehokkuus voidaan jakaa kahteen kategoriaan: operatiiviseen tehokkuuteen ja komponenttien tehokkuuteen. Operatiivinen tehokkuus tarkoittaa järjestelmässä toimivien ohjelmien ajon hyötysuhdetta ja palvelimen käytöstä. Palvelimella, jonka hyötykäyttösuhde on korkeampi, on alhaisempi sähkön hyötykulutus. Tämä johtuu siitä, että kone ollessaan päällä kuluttaa aina sähkövirtaa ja palvelimia ei sammuteta ja käynnistetä käytön mukaan, vaan ne ovat yleensä aina päällä. Komponenttien tehokkuudella tarkoitetaan niiden hyötysuhdetta käyttää sähkövirtaa mahdollisimman paljon laskentaan tai niiden tehtävään ja tuottaen mahdollisimman vähän jätelämpöä. Toisaalta joskus puhutaan komponenttien tehokkuudesta merkityksessä tuottaa mahdollisimman suurta suorituskykyä virran kulutukseen nähden ja jättää huomiotta, kuinka paljon komponentti tuottaa lämpöä. (Själänder – Martonosi – Kaxiras 2014. 18–59.)

Palvelinkeskukset tai datakeskukset koostuvat useista fyysisistä palvelimista ja laitteistoista. Niitä kutsutaan solmukohdiksi, joilla jokaisella on roolinsa järjestelmässä. Roolit voivat olla samoja, mutta perusroolit tulee kuitenkin toteuttaa. Kytkimen rooli on ohjata tehtäviä oikeille solmukohdille, missä kytkin voi olla esimerkiksi perinteinen pakettien ohjauskytkin tai tehtävien esikäsittelevä ja jakava laitteisto. Yksinkertainen kytkin vain vastaanottaa tietoliikennepaketit ja lähettää sen eteenpäin järjestelmän sisällä. Suuremmalla suorituskyvyllä oleva laitteisto saattaa toimia yhteyspalvelimena ja käyttää muita solmukohtia vain joidenkin palveluiden toteuttamiseen. (Själänder ym. 2014, 34.)

2.3 Palvelintietokoneen osat

Konejärjestelmä koostuu useista osista, missä osat koostuvat emolevystä, suorittimista, laitteisto-ohjaimista, muisteista ja kiintolevyistä. Emolevy hallitsee kaikkia tietokoneen sisällä tapahtuvaa ja ulos lähtevää. Monet osat saattavat olla integroituneina toisiinsa ja toiset modulaarisia lisäkortteja. Tietokoneeseen voi liittää yleensä monenlaisia lisäosia, joilla on parempi suorituskyky erikoistehtäviin tai lisäämään suorituskykyä ja kapasiteettia palveluiden toteuttamiseen. (Singh 2009, 5-9.)

2.3.1 CPU

Central processing unit (CPU) eli prosessori on suoritin, joka laskee aritmeettisia, loogisia laskuja ja hallinnoi tietokoneen prosesseja käskytaulukon mukaisesti. CPU ei ole yksinkertainen laskentasiiru, vaan se koostuu eri osista, joilla jokaisella on oma merkityksensä prosessorin toiminnassa. Suorittimella hallintayksikkö kertoo laskenta- ja IO-yksiköille käytetyn arkkitehtuurin mukaisesti ohjelmakäskyjen toteutuksen. Hallintayksikkö ohjaa väylillä dataliikennettä oheislaitteiden ja logiikkalaskimen välillä. 64-bittisyys yleisin väyläleveys suorittimissa ja se kertoo laskettavan tiedon enimmäiskoon. Suorittimen logiikkayksikkö kykenee vain yhteen prosessiin kerrallaan, mutta nopean prosessin vaihtelulla saadaan käyttäjälle luotua vaikutus rinnakkaisprosessoinnista eli moniajosta. Suoritin toisaalta saattaa koostua useammasta ytimeistä, mutta harvemmin erillisistä muistiväylistä, jolloin ytimet joutuvat jakamaan ajettavan ohjelman suorituksen. (Young 2009, 87–115.)

Suorittimia on kehitetty useita erilaisia, joista jokaisella on oma vahvuusalueensa tiedonkäsittelyssä tai ympäristössä. Toiset sirut on suunniteltu sulautettuihin järjestelmiin toteuttamaan mahdollisimman paljon toiminnollisuuksia pienessä koossa ja vähäisellä virran käytöllä. Siruja luodaan myös raskaaseen ja kevyeen laskentaan. Niiden käyttöympäristöön vaikuttaa käytetyt käskytaulukot ja komponenttien mikroarkkitehtuuri. (Young 2009, 87–115.)

Palvelimissa halutaan yleensä mahdollisimman suurta kykyä selviytyä useista prosesseista yhtä aikaa, mutta silloin tällöin rakennetaan palvelimia tai suurtietokoneita, jotka kykenevät mahdollisimman kompleksisiin laskuoperaatioihin.

Nämä suurtietokoneet ovat pääsääntöisesti tiedeyhteisöjen ja suurten yritysten käytössä tutkimustöissä. Palvelimissa löytyy usein useita laskentaytimiä ja valtavia muistikapasiteetteja, jotta voitaisiin tarjota mahdollisimman suurelle käyttäjäkannalle vakaita palveluja. Siksi palvelinympäristöissä saattaa olla käytössä useampi suoritin, joissa jokaisessa on tavallista prosessoria enemmän laskentaytimiä. (Young 2009, 87–115.)

Laskentasuoritukset tapahtuvat kellosykleittäin ja suorittimen käyttämä taajuus periaatteessa kertoo suorittimen laskentanopeuden. Tämä ei kuitenkaan suoraan kerro prosessorin laskentakyvystä, varsinkin kun erilaiset käyttöympäristöt antavat prosessoreille hyvin erilaisia laskentatehtäviä. Siksi prosessori tai palvelin tulisikin valita aina käyttöympäristönsä mukaan. (Young 2009, 87–115.)

Intel ja AMD ovat tunnetuimpia palvelinprosessorivalmistajia. Intel valmistaa Xeon-, Itanium- ja Atom-tuoteperheen prosessoreja palvelinympäristöjä silmällä pitäen. AMD:lla on vastaavasti Opteron- sekä Opteron -A- prosessorituoteperheet. Näissä vaihtelee suunniteltu laskentakuorma ja toteutettavien tehtävien monimutkaisuudet. Atom ja Opteron -A ovat suunniteltuja kevyisiin ja sulautettuihin ratkaisuihin. (AMD Server Processors. 2015; Processors.)

2.3.2 RAM ja ROM

RAM (Random-Access Memory) eli tietokonejärjestelmän keskusmuisti hoitaa väliaikaista tietoa järjestelmien osien välillä. ROM (Read Only Memory) on luku-muistia, josta tietokone käynnistettäessä hakee alustustietonsa järjestelmän ylösajoon, kuten BIOS-(Basic Input-Output System) ohjelman. Nykyään UEFI-(Unified Extensible Firmware Interface) standardi on korvannut BIOSin käytön emolevyissä. (Young 2009, 37–43.)

Samankaltaisista lyhenteistä huolimatta nämä kaksi muistitekniikkaa ovat hyvin erilaisia tehtäviltään ja toiminnoiltaan. Vielä nykyään näiden komponenttien fyysinen koostumus poikkeaa täysin toisistaan, minkä takia niille on laadittu juuri nimenomaiset tehtävät. Muistit jaetaan haihtuviin ja haihtumattomiin muisteihin sen mukaan, miten ne reagoivat jännitteen puutteeseen. Muistipiirien suurin ero

on tiedon säilyminen jännitteen laskettua alle kynnysjännitteen. Tällöin tyypillisissä RAM-piireissä ollut tieto katoaa, kun taas ROM-piireissä tieto säilyy muuttumattomana. (Young 2009, 37–43.)

Aito ROM on kerran tehtaalla kirjoitettu valmiiksi, eikä sitä voi muuttaa jälkikäteen. Tästä on paljon poikkeuksia, missä ROM-muistin sijaan käytetään haihtumattomia muistipiirejä, joille voi kirjoittaa uudestaan tai ROM-piirin muistijatkeita, joihin kirjoitetaan yliajavat muutokset. Näitä muutoksia voi olla esimerkiksi väylänopeuksien muutokset tai käytettävät jännitteet ja virrat. ROM -muistipiireissä käytetään perinteisesti flash- tai EEPROM-tekniikkaa niiden pitkäikäisyyden ja tiedon säilyvyyden takia. Kumpikaan tekniikoista ei ole kovinkaan nopeita luku- tai kirjoitusoperaatioissa, mutta tämän hetken käytettävissä olevat tekniikat eivät tarjoa parempaa ja toisaalta niiden roolin takia lisänopeudesta ei olisikaan suurta hyötyä. (Young 2009, 44–49.)

RAM on nykyisissä konejärjestelmissä aina haihtuvaa muistia, koska mitkään muut tekniikat eivät kykene samanlaisiin suorituskykyihin. Käytetyin ja nykyään lähes ainoa keskusmuistina käytetty tekniikka on DRAM (Dynamic Random-Access Memory) ja niistä DDR (Double Data Rate)-tyypin evoluutiot. DDR3-tyypin muistia käytetään vielä paljon, mutta DDR4 on uusin tyyppi ja se yleistyy kovaa vauhtia. Silloin tällöin tulee vastaan sulautettuja ratkaisuja, jotka käyttävät DDR:n sijaan GDDR (Graphical Double Data Rate)-tyypin 5 muistia. Tämä johtuu siitä, että tilaa sulautetuissa järjestelmissä on vähän ja CPU sekä GPU (Graphics Processing Unit) jakavat saman muistipiirin välimuistinaan. Tiedon haihtuminen jännitteen laskiessa ei ole haluttu ominaisuus muistipiireissä, vaan se on suurempien luku- ja kirjoitusnopeuksien kehittelystä syntynyt ominaisuus. DDR-muistit käyttävät kondensaattoreita bittivarastoina, ja kun jännite laskee, niin bitin HIGH-tila alkaa jännitteen myötä pikkuhiljaa laskemaan, kunnes muistipiirissä ei ole kuin LOW-tilan bittejä. (Young 2009, 50–72.)

Muisteihin on olemassa jo muutamia uusia kehitteillä olevia ratkaisuja, jotka saattavat tuoda nykyisiä muisteja nopeampia haihtumattomia tekniikoita. Palvelimissa muistien kasvavista nopeuksista on suurta hyötyä ja virrankulutuksellisesti

muistit, joiden ylläpitoon ei tarvitse paljoa virtaa, säästävät jo suurissa kokonaisuuksissa valtavasti sähköä. Kuluttajien tietokoneet lähinnä saavat uuden virransäästötilan, missä tietokone ei käytä virtaa valmiustilaa enempää, mutta siirtyy käyttö- ja virransäästötilojen välillä lähes välittömästi. Kenties joskus tiedon haihtumattomuus on niin vakaata, että tulevat muistit voivat korvata nykyään massamuistina käytössä olevia kiintolevyjä. (Young 2009, 50–72.)

2.3.3 Massamuisti

Tietokoneen tallennustilaa kutsutaan massamuistilaitteeksi. Sinne kirjoitetaan kaikki, mitä tarvitaan tallentaa pitkäaikaiseen säilytykseen. Historiassa massamuistit ovat muuttuneet materiaaleiltaan ja kapasiteeteiltaan, mutta niiden funktio on säilynyt samankaltaisena. Suurin muutos on ollut mekaanisuuden vähentyminen ja tallennettavan tiedon fyysinen pienentyminen. Tämä on johtanut lukuisiin uudelleenkirjoitusmahdollisuuksiin ja suurempiin tiedontallennuskapasiteetteihin. Massamuistit ovat palvelimissa lähes aina erillisiä tietoliikenneväylillä konejärjestelmään kiinnitettäviä laitteita. Konejärjestelmät eivät ole riippuvaisia toimimaan alkuperäisillä massamuistilaitteilla, vaan niitä voidaan vaihtaa ja lisätä. (Singh 2009, 57–64.)

Mekaanisen vioittumisherkkyyden takia massamuisteihin kehiteltiin RAID (Redundant Array of Independent Disks)-tekniikat, mitkä mahdollistivat useampien massamuistilaitteiden synkronisoidun käytön, nopeuttaen käyttöä ja kasvat-
taen vikasietoisuutta. Nämä tekniikat mahdollistavat kiintolevyjen lennosta vaihtamisen palvelimille, jopa käytössä olevilla levyillä. Koska useampi levy pystyi pitämään hallussa samoja tietoja ja tarkistussummilla synkronisoitiin levyjen välistä tietoa, niin vaihdettavat kiintolevyt voitiin poistaa ja asentaa uudet tilalle. Tällöin järjestelmä synkronisoi uudelle levyille poistettujen levyjen tiedot käyttäen järjestelmään jääneiden levyjen tietoja. Tämän aikana käyttö pystyi jatkumaan ilman häiriötä. (Erkens–Haustein – Mueller-Friedt – Troppens – Wolafka 2009, 13–51.)

Osiointi on perinteinen tapa määritellä levyn käytettävä tila käyttöjärjestelmälle tallennusresurssiksi. Alustamalla osioon tehdään osion ensimmäiseen sektoriin käytettävä tiedostojärjestelmätunniste, joka on käyttöjärjestelmälle ohje käytettä-

västä tiedostojärjestelmästä. Tiedostojärjestelmä on rajapinta, joka mahdollistetaan tallennuslaitteiden käytön tietojen tallentamiseen ja lukemiseen sekä hakemistojen että tiedostojen muodossa. Tiedostojärjestelmien takia massamuistilaitteiden tekninen rakenne voidaan jättää huomiotta ja keskittyä olennaisempaan. Osiointi on toteutettu yleensä yhdelle massamuistilaitteelle käyttäen osaa sen tilasta tai kokonaan. Osioinnissa yleensä määrätään tiedostojen tallennukseen varausyksikön koko tiedostojärjestelmän mukaan. Tämä varausyksikkö on pieni lohko, jonka koko on 512 -tavua tai sen monikerta. Lohkoissa on tallennettu tieto ja lohkoja hallinnoidaan tietojärjestelmän kirjanpidossa. (Brouwer 2015.)

2.3.4 Oheislaitteet

Oheislaitteet ovat konejärjestelmään liitettäviä laitteita, joiden puuttuminen ei estä konejärjestelmän perustoimintoja. Niillä pikemminkin pyritään tehostamaan tai mahdollistamaan erikoistehtäviä, mihin peruslaitteisto ei välttämättä ole kykeneväinen. Näppäimistö, hiiri ja näyttö ovat oheislaitteita, joita palvelin ei ylösajon jälkeen tarvitse, mutta ovat tarpeellisia palvelinta asennettaessa ja paikallisesti diagnosoidessa vikoja. Yleensä palvelimia ei paikallisesti ylläpidetä, vaan ylläpito tapahtuu etäyhteyden avulla, jolloin paikallisesta käyttöliittymästä ei ole mitään etua. (Singh 2009, 11-36.)

Näytönohjain ja fysiikkakortit ovat oheislaitteita, jotka pyrkivät tuomaan lisää suorituskykyä erikoistehtäviin. Ne ovat räätälöityjä järjestelmiä omiin rooleihinsa ja siksi ne suoriutuvat huomattavasti tehokkaammin kuin tavalliset prosessorit erikoisosa-alueillaan. Näytönohjain tai grafiikkakortti löytää palvelimissa harvoin omaa erikoisrooliansa eli näytölle piirtyvien tapahtumien prosessointia, mutta sen sijaan palvelinympäristöissä on opittu hyötykäyttämään grafiikkakorttien valtavaa yksinkertaisten laskuoperaatioiden suorituskykyä. (Själänder ym. 2014. 39)

Grafiikkakorteissa on nykyään satoja tai jopa tuhansia laskentaytimiä, kun taas parhaimmissakin palvelinprosessoreissa on ytimien määrä 16 ytimen luokkaa vain kaksinkertaisella kellotaajuudella. Teoriassa näytönohjain pystyy siis toteuttamaan sata kertaa enemmän laskusuorituksia kuin prosessori, jolloin palvelinympäristössä siitä voi olla suurta etua, mikäli sen osaa hyödyntää oikein. (Själänder ym. 2014, 39.)

Näiden lisäksi oheislaitteina löytyy muitakin mahdollisesti tehokkuutta nostavia kortteja kuten äänikortti. Palvelimille hyödyllisiä lisäkortteja ovat erilaiset tietoliikennekortit. Näitä tietoliikennekortteja on langallisia kuitu- ja Ethernet-kortteja sekä langattoman tietoliikenteen WLAN-, Bluetooth- ja Zigbee-kortteja. Näiden korttien tehtävänä on yhdistää tai toteuttaa tietoliikenneverkkoja palvelimen käyttöön, jotta palvelin voisi toteuttaa palveluja yhdistyville laitteille. (Rackley 2007, 45–62.)

2.4 Virtuaalinen taso

Virtuaalinen taso toteutetaan palvelinympäristöissä, kun halutaan erottaa pohja-järjestelmä käytettävästä palvelinympäristöstä. Ymmärrettävin esimerkki palvelinjärjestelmissä on palvelinympäristön toteuttajan ja hallinnoijan ollessa eri organisaatio. Palvelinympäristöntoteuttaja luo palvelinkeskuksen ja tarjoaa asiakkailleen resursseja eli palvelimen suorituskykyä ja tallennustilaa. Tällöin on helppointa, kun toteuttajalla on yksi hallinnoitava järjestelmä ja jokaiselle asiakkaalle oma virtuaalinen palvelinympäristö. (Portnoy 2012, 9–11.)

Virtuaalisuuden tasoja on useita. Osassa tasoista palvelun tarjoaja antaa virtuaalisen palvelinympäristön suoraan käyttäjäjärjestelmän fyysisiä osia kuten prosessoreita tai ytimiä, jolloin fyysiset komponentit lukitaan kyseisiin virtuaalisiin järjestelmiin sopimuksen mukaisesti. (Portnoy 2012, 15–17.)

Resurssit voidaan myös jakaa käytettävän suorituskyvyn mukaan virtuaalisella tasolla. Tällöin voidaan varsinainen suorituskyky jakaa käytettävien prosessorien suoritusajan mukaan, jolloin virtuaalinen taso saa käyttää määritettyinä ajanjaksoina prosessoria tai prosessorin käytöstä lasketaan aikaa. Kellotaajuudella määritellään prosessorin käyttämä ajanjakson sykli. Prosessorin käyttämä aika on laskutuksen perusteena. Laskutuksen perusteena voidaan käyttää myös jakamalla virtuaalitasot prioriteetin mukaan, missä jonotusaika voi kasvaa hyvin pitkäksi matalan prioriteetin tasoilla, kun korkeamman prioriteetin tasot menevät edelle. (Portnoy 2012, 125–129.)

Virtualisoinnin hyvä puoli on, että järjestelmät ovat pääsääntöisesti fyysisestä tasosta riippumattomia ja niitä voidaan siirtää järjestelmästä toiseen ilman, että se vaikuttaa palvelinympäristöön. Virtualisointi tuo myös vakauden palvelimen fyysiseen tasoon, kun virtuaalisen tason kaatuminen ei voi kaataa koko palvelinjärjestelmää. Jos virtuaalinen taso on täysin kyvytön vastaamaan mihinkään käskyihin tai pyyntöihin, se voidaan uudelleen käynnistää alemmalta tasolta, ilman koko järjestelmän uudelleen käynnistystä. (Portnoy 2012, 198–201.)

Looginen levy on yhden tai useamman levyn virtuaalisen tilan taltioryhmä. Yhdessä massamuistilaitteessa voi olla myös useita virtuaalisia levyjä, jos niin halutaan. Loogisten levyjen ei tarvitse olla samassa palvelinjärjestelmässä kuin muut palvelinjärjestelmän osat. Ne voidaan kytkeä tietoliikenne väyliä pitkin tai suoraan laiteväylillä. Järjestelmään kytkemistavasta riippuu loogisen aseman käyttönopeus ja väylä koko. Käyttönopeus eli vasteaika tarkoittaa kulunutta aikaa pyynnön lähettämisestä vastauksen saamiseen ja väylän koko kuinka paljon tieto voi liikkua väylällä. SAN- (Storage Area Network) arkkitehtuuria käytetään tiedostopalvelimien yhdistämiseen palvelimien kanssa.

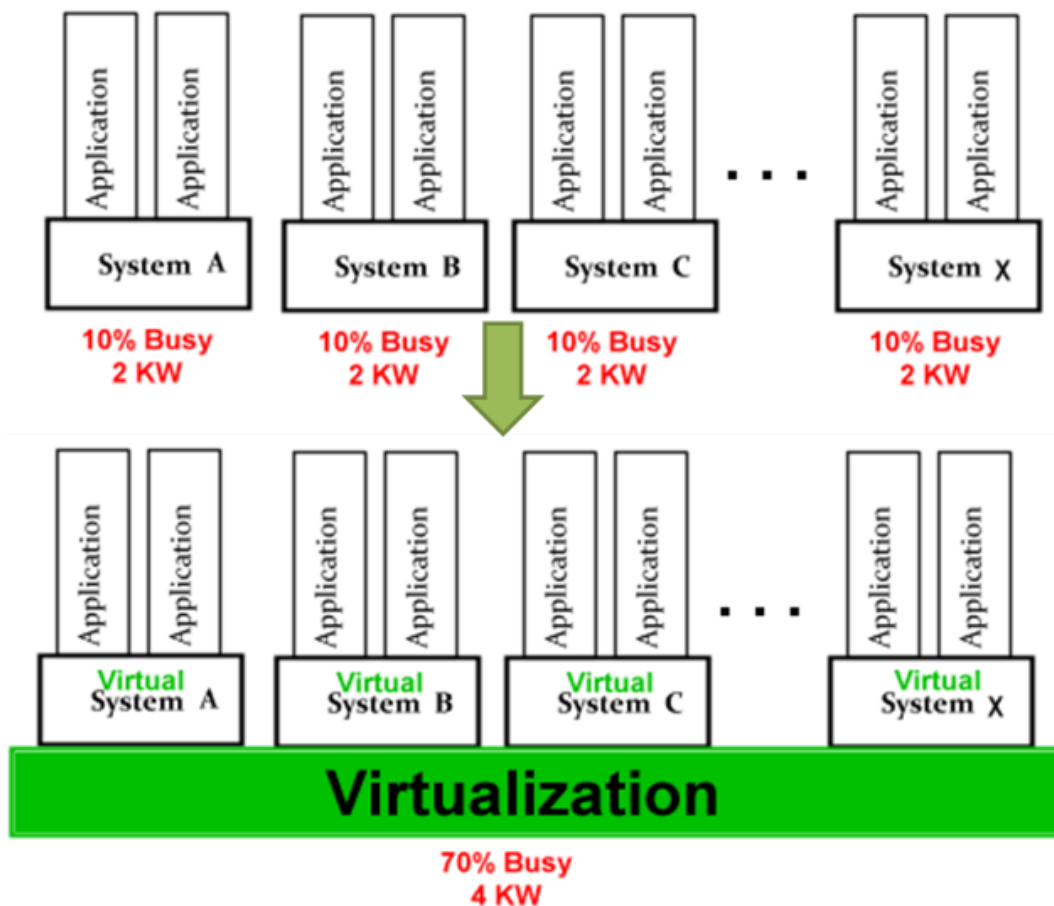
Virtuaalisia asemia tehdään, jotta ohjelmistojen olisi helpompi käsitellä yhtä käytettävää taltiointi tilaa useiden sijaan. Rajapinnat huolehtivat ohjelmille tiedon kulkeutumisesta oikeaan paikkaan. (Erkens ym. 2009, 172–173.)

2.5 Ekologinen ajattelumalli

Ekologinen ajattelumalli on nykyajan trendi ja havahtuminen ympäristövaikutuksiin on muokannut ajattelutapaamme toteuttaa uusia asioita ympäristöystävällisin menetelmin. Uusien palvelimien toteutukset eivät tee tässä poikkeusta. Komponenttien valmistuksessa pyritään käyttämään mahdollisimman vähän ympäristöä kuormittavia materiaaleja ja valmistusmenetelmiä sekä kehittämään niistä mahdollisimman tehokkaita. Infrastruktuurit palvelinjärjestelmäkehityksessä suuntaa keskitettyihin palvelinkokonaisuuksiin, jolloin niistä saadaan vähennettyä solmu kohtia ja monimutkaisuutta pois. (Spafford 2009, 13–14.)

Kuvassa kaksi havainnollistetaan tavanomaisten palvelimien yhdistämistä yhdelle virtuaalipalvelin järjestelmälle, missä palvelimen suorituskuorman ollessa

sama ja sähkönkulutus vain osa aikaisemmasta. Palvelinkeskuksista voidaan hyödyntää myös niistä aiheutuva hukkalämpö. Hukkalämpö voidaan johtaa yrityksen toimitilojen käyttöön. Suurimmat palvelinkeskukset pystyvät lämmittämään jopa asuntoja ja muita kuntainfrastruktuurin kuuluvia rakennuksia. Kun palvelimet keskittyvät läheisiin sijanteihin toistensa kanssa, on palvelimien välinen liikennöintikin huomattavasti nopeampaa ja tehokkaampaa. (Spafford 2009, 13–14.)



KUVA 2. Virtualisointi keventää palvelinten kokonaissähkökuormaa (Kamula 2010.)

3 YRITYKSIEN TARVE TOTEUTTAA PALVELUITA

3.1 Yleiset tarpeet palvelinympäristön toteutukseen

Palvelin halutaan toteuttamaan palveluita yritysten asiakkaille tai palvelutyökaluina yrityksen käyttöön. Palvelut voivat olla internetvälitteisiä tai sisäisiä, asiakkaan järjestelmästä riippuvia tai riippumattomia. Palvelut toteutetaan palvelinjärjestelmään, jossa palveluita myös ajetaan. Palveluohjelma ajetaan eli siis suoritetaan järjestelmässä, kun sen käynnistämiseen määritetyt ehdot täyttyvät. Ehtoja palveluiden ajoon voi toteuttaa monin tavoin, mutta pääsääntöisesti ehtoina ovat määritetyin ajoin ajettavat palvelut palvelinjärjestelmän suorittamana ja käsky suorittaa järjestelmän palvelu. (Brown.)

Palvelu on hyödykkeen tuottamista tilaajalle tai asiakkaalle, missä hyödyke on pääsääntöisesti aineeton. Tosin aineettomuus on venyvä käsite, kun palvelusta toteutuva tieto voi olla haihtuvaa tai talteen jäävää ja arkistoituna tieto tallennetaan tapauskohtaiseen taltiointilaitteeseen, kuten kiintolevyille. Tämä siis on toisaalta myös tuottamista. Palvelin kuitenkin toteuttaa palveluita ja sitä voidaan verrata asiakaspalveluun käytännöllisessä merkityksessä. Kun asiakas saapuu liikkeeseen, hän hakee itsellensä vuoroa asiakaspalvelusta. Tämä tapahtuma on sama kun asiakas hakee verkosta esimerkiksi yrityksen internetsivua tai verkkosovellusta. Erona on, että asiakaspalvelun työntekijä palvelee yhtä asiakasta kerrallaan, kun serverillä voi olla helposti tuhansia asiakkaita, niin ihmisiä kuin laitteita, eikä asiakas sitä edes huomaa. Kun asiakas ja asiakaspalvelija keskustelvat, niin sitä voi verrata internet sivulla selailuun tai palvelimen sovelluksien käyttämiseen. Samalla lailla yksinkertaistetulla ajattelulla asiakaspääte keskustelelee palvelinkoneen kanssa. Ne vaihtavat vuoropuhelun tavoin kysymyksiä ja vastauksia. (Brown.)

Nykyään ihmiset eivät enää mene liikkeisiin niin usein kuin ennen hakemaan tietoa ja ostamaan palvelua tai edes tuotetta. Monet yritykset ovat saaneet kokea, että verkossa palvelujen ja tuotteiden myyminen on tuottoisaa bisnestä. Kun asiakas lähtee asioimaan liikkeeseen, hänellä kuluu aikaa siirtyä liiketiloihin. Asiak-

kaalla kuluu aikaa myös liiketilassa tuotteiden etsimiseen ja liikkumiseen eri osastojen välillä. Asiakas todennäköisesti joutuu myös odottamaan omaa vuoroaan saadakseen palvelua.

Yritykset pyrkivät pitämään asiakaspalvelutyöntekijöiden määrän minimissä. Asiakaspalvelija, joka ei palvele jatkuvasti asiakasta, aiheuttaa ylimääräistä kuluja yrityksen tuloksessa. Yksi asiakaspalvelija ansaitsee kuukaudessa 2105 €, kun hän tekee töitä minipalkalla tietualan työehtosopimuksen (Tietotekniikan palvelualan työehtosopimus. 2013) mukaan. Hänen työpanoksensa aiheuttaa kuluja yritykselle yrittäjien palkkalaskurin (Palkkalaskuri. 2015) laskentakaavan mukaan vähimmillään noin 2600 € kuukaudessa. Laskentakaavassa käytettiin seuraavia arvoja, TyEL 17,5 %, sosiaaliturvan maksuluokka I 2,08 %, pakolliset vakuutukset 0,7 % ja työttömyys 0,8 %. Tällä summalla yritys pystyisi hankkimaan käyttöönsä melko kattavan palvelinratkaisun, jopa kohtalaisen suurenkin yrityksen käyttöön. Esimerkiksi jopa suuren yrityksen käyttöön sopiva pilvipalvelu OVH:lta XL-tuotepaketti Intel valmistamilla suorittimilla kustantaisi 2158,78 €/kk (Tuotepaketit. 2015).

Palvelinta ei toteuteta vain asiakkaiden palvelemiseen. Myös yrityksen sisäisille toimille löytyy tehostavia ja turvaa luovia ratkaisuja. Voi olla, että yritys haluaa arkistoida omia asiakirjojaan palvelimelle ja saada varmuuskopioitua niitä turvaan. Yritys voi tehdä ryhmätyöskentelyä, jossa palvelin hallinnoi työstettävää työtä ja työntekijät toteuttavat tuotetta omilta päätteiltään, esimerkiksi Googlen Docs -sovellus. Tällaisilla sovelluksilla saadaan työt hallitusti yhteen paikkaan ja työntekijät voivat nähdä paremmin omat työtehtävänsä. Organisaatio voi toteuttaa palvelimellaan yritykselle organisaation strategisen hallintaympäristön. Tämä tarkoittaa, että yritys voi laatia tehtävälisteriä, aikatauluja, työtehtävä- ja projekti-suunnitelmia työntekijöiden noudatettavaksi. (Brown.)

3.2 Pienten ja keskisuurten yritysten tarve erot

Pienet yritykset tarvitsevat nettisivustoja markkinoimaan omia tuotteitaan ja palveluita Internetin yli mahdollisimman laajalle asiakaskunnalle. Yritys tarvitsee järjestelmän, jolla voi hallita erilaisia tietorekistereitä esimerkiksi asiakasrekisteriä, sekä taltioimaan yritystä koskevia tietoja kuten kirjanpitoa tai yhtiöpöytäkirjoja,

että toteuttamaan asiakas läheisiä sovelluspalveluita. Keskisuurten yritysten tarpeet eroavat pienestä kasvaneen yrityksen koossa. Keskisuurilla yrityksillä on enemmän henkilöstöä tai liikevaihtoa kuin pienillä yrityksillä. Tällöin projektien hallinta on järkevää toteuttaa yhteen sijaintiin, josta esimiehet ja projektin osapuolet voivat seurata projektin etenemistä.

3.3 Vaihtoehtoiset ratkaisut palvelinympäristölle

Palvelimilla toteutetaan yleensä yritykselle nettisivut, hoidetaan kirjanpitoa, asiakasrekisteriä ja markkinointia tai tarjotaan asiakkaille websovellusta. Palvelimen toteuttaminen ei ole todellakaan välttämättömyys. Yritykselle löytyy kyllä halpoja tai ilmaisia ratkaisuja, joita palvelimen ajateltiin toteuttavan. Tällöin pitää miettiä toteuttaako palvelinratkaisu paremmin yrityksen tarpeet, kuin vaihtoehtoiset menetelmät. (Brown.)

Yrityksen nettisivut voidaan toteuttaa ostamalla yksityiseltä henkilöltä tai yritykseltä nettisivujen ylläpito ja luomispalvelu. Se voidaan korvata toteuttamalla sosiaalisiin medioihin yrityksen sivusto ja hallinnoida päivityksiä yritykselle saatavan tunnuksen avulla. Sosiaalisiin medioihin liittyminen on yrityksille joka tapauksessa kannattavaa, koska se lisää yrityksen näkyvyyttä ja markkina-arvoa. (Brown.)

Kirjanpitoa voi hoitaa kotikoneella ja käsin tai ostamalla kirjanpitopalvelun tilitoimistolta. On kuitenkin huomioitava, että varmuuskopiointia ei kannata laiminlyödä ja yritysten kirjanpidolle on säädetty laki (L 30.12.1997/1336. 1997). Vaikka yritys itse tekisikin kirjanpidon, niin yritys yleensä tarvitsee tilintarkastajan (L 18.9.2015/1141. 2015). Markkinointi on hoidettu tyypillisesti markkinointifirmojen kautta ja useat firmat tukeutuvat markkinointi firmoihin, vaikka heillä olisi palvelin. Palvelimella kun ei saa näkyvyyttä muuta kuin verkkomedioihin.

Websovelluksien ja rekisterien pitäminen voi toteuttaa ilman palvelinta, jos vaatimukset rekisterille ja sovellukselle ei ole liian erikoisia. Sovelluksen kanssa voi miettiä onko se välttämätön toteuttaa selaimessa toimivaksi. Jos sen voi toteuttaa mobiili- ja työpöytäsovelluksena, niin jakelukanavia löytyy siinä tapauksessa, eikä

palvelinta tarvita. Jos sovellus tarvitsee tallentaa jotain pelaajarekisteriä ja tilastointia, niin vaihtoehtona on neuvotella tilastopalvelin tarjoajien kanssa sopimuksesta sovelluksen rekistereiden ylläpidosta.

4 TURVALLISUUS JA TEHOKKUUS

Turvallisuus ja tehokkuus ovat palvelinratkaisuille erittäin tärkeitä piirteitä. Kukaan ei haluaisi toteuttaa palvelinta, jonka tietojen turvallisuuteen ei voisi luottaa, tai sen toimintakyky ei voisi taata edes peruspalvelutarpeita. Palvelimia vastaan on paljon erimuotoisia uhkakuvia, mutta onneksi niitä vastaan suojautuminen ei ole niin monimutkaista. Tehokkuus ja kustannukset voivat olla hankalia asioita käsitellä, mutta niihinkin löytyy hyviä peruspäteviä ratkaisuja.

Palvelimien turvallisuus on myös muuta kuin laitteiston tietoturvallisuus. Turvallisuus pitää hahmottaa laajemmin ja avoimella mielellä. Kun palvelinratkaisua lähtee suunnittelemaan, on hyvä miettiä yksi asia kerrallaan ja ottaa huomioon mahdolliset uhkakuvat. Jokaiselle asiakohdalle voi löytää ongelmia, mutta aina ongelmat eivät ole uhkaavan suuria. Kun ongelma havaitaan, niin sille voi hahmottaa seuraukset ja sen vaikutukset (Bartolini & Gaspari 2009. 164–165). Palvelinratkaisussa pitää toteuttaa sekä riskianalyysi, että arvioida riskitaso ja vaikutukset. Tällöin tiedetään onko ongelmalle etsittävä ratkaisu. Riskianalyysin ei tarvitse olla raskas prosessi, vaan sen pystyy toteuttamaan pienellä miettimisellä, muistiinpanoilla ja tietojenhakemisella. (Shostack 2014, 59–60.)

Kustannuksien ja operatiivisuuden tehokkuuden arvioimiseen löytyy useampiakin mittareita, ja arvioinnin voi tehdä tarkalla laskennalla tai kevyemmällä tutkimuksella. Kokonaiskustannuksien kanssa tulee huomioida, että tehokkuuden tutkiminen itsessään vie aikaa ja rahaa. Tulee miettiä onko järkevää kuluttaa yrityksen resursseja liian täsmälliseen kustannus-hyötyanalyysiin. Tärkeää on kuitenkin toteuttaa edes pienimuotoinen analyysi, jotta ei kansanomaisesti sanottuna ”osteta sikaa säkissä”. (Chevance 2004. 656.)

Yksityishenkilö voi toteuttaa palvelimen halvimmillaan valjastamalla vanhan tietokoneensa tai ostamalla halvan käytetyn tietokoneen. Tietokoneeseen hänen tulee asentaa palvelinohjelmisto ja käyttää kotiverkkoa internetväylänä. Maailmalla on paljon ilmaisia DNS-nimipalveluita (Domain Name System) ja jos koti-osoite on dynaamisen IP-osoitteen alla, niin käytettäväksi tulee dynaaminen ni-

mipalvelu. Kotipalvelimen pystytys on sallittua ja sen myötä kotiverkkoa saa käyttää vain yksityiskäytössä. Yrityskin voi päästä halvimmillaan samalla tavoin, kuin yksityishenkilö, mutta silloin kannattaa huolehtia, että operaattorit eivät hyväksy kotiverkon yrityskäyttöä, vaan yrityskäyttöön tulisi sopia erillinen yritysverkkosopimus. Lisäksi tulee huomioida asiakkaiden tarpeet ja varmistaa, että palvelinresurssit ovat riittävät. Palveluiden toteuttamisessa on riskinsä ja yrityksen voi olla kannattavaa maksaa palvelimien suorituskyvystä enemmän kuin yksityishenkilön, jotta palveluiden tarjoaminen saadaan luotettavammalle pohjalle. (Chevance 2004. 688)

4.1 Turvallisuuden ongelmat ja niiden mahdolliset ratkaisut

Mahdolliset uhkakuvat, joita palvelinympäristö voi kohdata, liittyvät aineellisiin ja aineettomiin vahinkoihin tai molempiin. Aineellisiin vahinkoihin kuuluu ympäristön aiheuttamat uhat, jolloin fyysiselle palvelimelle voi syntyä vahinkoa. Tällaisia tapauksia voivat olla tulipalot, ilkivalta tai palvelinhuoneen lämmönvaihdon laiminlyöminen. Tällaisilta ongelmilta on helppo välttyä, kun varmistaa palvelimen sijoittamisen ja säilytystilan asianmukaisuudet. Fyysistä vahinkoa saattaa aiheutua myös odottamattomista luonnonkatastrofeista kuten tulvasta. Tällöin keskisuurin yrityksen kannattaa tukeutua lähinnä tiedostojen varmuuskopiontien turvalliseen säilytykseen, koska varotoimet näin harvinaisten tilanteiden varalle ovat kalliit. Kohtuullisen usein palvelimille aiheutuu fyysistä vahinkoa sähköjakeluverkkojen häiriöistä. Jännitevaihtelulta voi suojautua kytkemällä palvelinjärjestelmän eteen UPS (Uninterruptible Power Supply)-laitteen. Sillä voidaan taata tasainen jännite ja sähkökatkoksien varalle UPS -laitteessa on tarpeeksi virtaa sammuttamaan palvelin turvallisesti. (Shostack 2014, 108)

Aineettomat uhkakuvat ovat hyökkääjät ja tietojen häviäminen. Hyökkääjillä on lukuisia keinoja ja tavoitteita palvelimia vastaan. Hyökkäyksissä yleensä tavoitteena on varastaa tietoa, luoda uusia hyökkäysverkkoja tai vahingoittaa yrityksen palvelujen tarjoamista. Tunnetuimpia ohjelmistosodankäynnin (Seitz 2014, 61–69) menetelmiä ovat virukset, palvelunestohyökkäykset ja salasanan haistelu. Kaikkia hyökkäyksiä vastaan toimivat parhaiten hyvin toteutetut palomuurit, vä-

häinen ohjelmistojen määrä, tietoliikenteen salaaminen ja epätyypilliset palvelinjärjestelmät. Hyökkääjä pyrkii etsimään järjestelmästä haavoittuvuuksia seuraamalla tietoliikennettä ja pommittamalla tietoliikenneväyliä. Haistelemalla hyökkääjä etsii salasanoja tai tuttuja ohjelmia, joihin hän tietää haavoittuvuudet. Mikäli hyökkääjä pääsee järjestelmään läpi, hän pyrkii todennäköisesti viemään järjestelmästä häntä kiinnostavaa tietoa tai vahingoittamaan palvelinta. Joskus hyökkääjä saa vahingoitettua laitteita ajamalla palvelinta yli sen lämpökuormituskyvyn, jolloin komponentit voivat vioittua. Jos hyökkääjä ei pääse läpi järjestelmään, niin todennäköisesti on odotettavissa palvelunestohyökkäys. Tällöin tavoitteena on ylikuormittaa tietoliikenneväylä palvelimelle, jolloin palvelin ei pysty vastaamaan palvelupyyntöihin. (Seitz 2014, 61–69)

Hyökkäyksiä voi torjua säätämällä palomuurin asetuksia niin, että palomuri hyväksyy vain käytössä olevat väylät, sekä reagoi vain osoitettuihin palvelupyntöihin ennalta määritetyn kertojen verran. Mitä vähemmän on ohjelmia, jotka keskustelevat internetin ylitse, sitä vähemmän on mahdollisia tietoturva-aukkoja. Turhien ohjelmien pitämistä järjestelmässä tulisi välttää. Yksi turvatoimi on käyttää järjestelmää, joka ei ole suuressa käytössä. Tällöin se ei ole todennäköisesti tuttu hyökkääjälle, eikä hyökkääjä silloin tunne järjestelmän toimintaa. Vastavaikoilu on turvatoimi, jossa seurataan epäilyttävää verkkokäyttäytymistä. Tavoitteena on selvittää epäilyttävien kohteiden osoitteet ja tehdä häirintää estäviä toimia. (Seitz 2014, 129)

4.2 Kustannustehokkuus

Kustannustehokkuus otetaan huomioon, kun yrityksen taloudesta kannetaan erityistä huolta. Toteutettu palvelinratkaisu ei tuo parasta liikevoittoa, jollei yrityksen tarpeiden vastaavuutta arvioida ja tehdä palvelimen toteuttamiseen selvää suunnitelmaa.

Palvelin ei ole aina kustannustehokkain ratkaisu yrityksen toiminnan edistämiseksi. Palvelimen toteuttamiseen ja ylläpitämiseenkin kuluu resursseja ja kaikilla yrityksillä ei niitä ole. Lisäksi ne rokottavat resursseja yrityksen perustoimien toteutukselta. Tällainen voisi olla esimerkiksi tilanne, jossa yritystä on hoidettu pe-

rinteisesti ja sekä internet että sähköinen media tai jopa tietotekniikka ovat yrityksen toimijoille suhteellisen outoja käsitteitä. Yrityksen resurssit kuluvat selvitetessä toteutusmahdollisuuksia ja yleensä, jos asiantuntemusta yrityksessä ei ole, niin päädytään hyväksymään ensimmäinen vastaan tuleva tarjous. Tällöin ei kuitenkaan välttämättä saada parasta mahdollista vastinetta yrityksen tarpeisiin. (Barroso – Clidaras – Hölzle 2013. 33–38.)

Joskus yritykset tekevät itse oman fyysisen palvelimensa, eli heillä saattaa olla hieman tietotaitoa taustalla. Tällöin pitää ottaa huomioon yrityksen palvelimen mahdollisuus kuormituksen kasvamiseen. Kun palvelupyyntöjen määrä alkaa ylikuormittaa palvelimen suorituskykyä, rasitus tuntuu kaikissa palvelimen operatioissa. Palvelin ei tällöin voi vastata palvelupyyntöihin joko ollenkaan tai se alkaa vastaamaan niihin hitaasti, jolloin sen merkitys yrityksen lisäpalveluntoteuttajana romahtaa. Toisaalta yritys saattaa myös yliampua palvelinta hankkiessa palvelimen suorituskyvyn ja ostaa kalleimman mahdollisen palvelimen, mutta yrityksen tarpeet eivät koskaan saavuta palvelimen täyttä kapasiteettia. Tämä ei ole niin paha asia kuin kapasiteetin loppuminen. Kustannustehokkaampi ratkaisu olisi ollut hankkia yrityksen tarpeisiin sopivampi palvelin ja tulevaisuudessa päivittää sitä yrityksen tarpeiden kasvaessa ja tekniikan kehittyessä. Tällöin yrityksen vanha palvelin voidaan myydä, jolloin se muuttuu yrityksen taseessa liikevaihdoksi ja käytettäväksi muuhun. Palvelin voidaan toisaalta säilyttää yrityksen taseessa ja säästää se varajärjestelmäksi tai toteuttamaan muita palveluita, kuten yrityksen sisäisten asiakirjojen arkistointia ja viestintää varten. (Barroso ym. 2013. 33–38)

Ostopalveluihin pätee samat huomioitavat seikat, mutta erilaisina. Palvelinpalveluissa on palvelinten päivittäminen huomattavasti helpompaa ja sen ylläpidosta ei tarvitse vastata fyysisellä tasolla. Palvelinpalveluiden ostaminen on monimutkaisempaa yrityksille ostovaiheessa, kuin toimintavaiheessa. Yrityksen tarpeet voidaan kategorisoida moneen lohkoon pilvipalvelinjärjestelmissä. Näitä lohkoja ovat: joustavuus, luotettavuus ja laaja järjestelmätuki. Ostovaiheessa yrityksen tulee huomioida seuraavat asiat: mihin palvelin sijoittuu, mitä palveluita palvelusopimus sisältää, minkälaisia toimia se vaatii yritykseltä, palvelun hallinnoiminen, pal-

velun käytettävyyssä ja nimipalvelut. Ostopalvelun parhaana puolena on ehdottomasti helppous, koska yrityksen ei itse tarvitse huolehtia palvelimien toiminnasta ja ylläpidosta. (Tuloslaskelma. 2015.)

5 PALVELINJÄRJESTELMÄT JA OHJELMISTOT

5.1 Palvelinjärjestelmät

Palvelinjärjestelmät ovat jakautuneet kahteen ohjelmistokuntaan Microsoftiin ja Linux-jakeluihin. Linux on näistä huomattavasti yleisempi, mutta ei Microsoftkaan ole huonossa asemassa. Järjestelmien ympäristöjen suurimmat erot liittyvät palvelujen ohjelmointiin, ylläpitoon ja laitteistorajapinnan tukeen. Linux-palvelimet ovat yleisempiä, koska ne ovat helpommin muokattavissa erilaisiin laitteistoihin ja ne ovat lisenssimaksuttomia, jolloin palvelinkokeiluja on helpompi toteuttaa pienillä kustannuksilla. Linuxin ja Microsoftin palvelinjaketut pohjautuvat hyvin paljon niiden työpöytäjakeluversioihin. Yleensä palvelinjakeluista on poistettu graafinen käyttöliittymä tai sen voi erikseen valita käyttöön. Sen lisäksi niistä on poistettu perinteisten työpöytäversioiden ohjelmapaketteja, jotka ovat palvelimille turhia ja niihin on lisätty palvelimille hyödyllisiä ohjelmapaketteja. (Portnoy 2012, 35-50.)

Microsoft Windows pohjaiset palvelimet ovat hyvin samankaltaisia käyttöliittymiltään, kuin työpöytä Windowsit. Suurimmat erot tulevatkin niiden sisältämistä palvelinohjelmistoista. IIS (Internet Information Services) on Windows-ympäristöissä palvelinohjelmistokokonaisuus, jolla voidaan toteuttaa pääsääntöisesti kaikki Windowsilla toteutettavat palvelinpalvelut. IIS:n olemassaolo ei estä muiden ohjelmistojen käyttöä, mutta on huolehdittava, etteivät ne aiheuta ristiriitoja järjestelmän sisällä. Nykyään Windows palvelinjakeluissa voidaan virtualisoida sisäisiä palvelimia Hyper-V tekniikalla, jolloin virtualisoivan Windows-käyttöjärjestelmän voi jättää hallinnoivaksi laiterajapinnaksi. Windowsin etuna on useammalle käyttäjälle tutumpi käyttöympäristö ja helpompi lähestyttävyyys. Windowsiin tehdyt ohjelmat toimivat todennäköisemmin ilman räätälöintiä, kuin Linux-jakeluihin tehdyt ohjelmistot. Tämä johtuu Linux-ympäristöjen eroista keskenään, jolloin toiselle tehty ohjelmisto ei välttämättä toimi toisessa ympäristössä olevien pakettien kanssa. Windows palvelinohjelmistot ovat hyvin valmiita paketteja palveluiden toteuttamiseen, jolloin niihin tarvitsee harvoin asentaa ulkopuolisia komponentteja.

Näitä komponentteja ei löydy Microsoft Windowsin ohjelmistojakelulistoilta. Sovelluspalvelinohjelmistojen tekoon on järjestelmässä ASP.NET Framework –tuki ja lisä tukea saadaan asentamalla Java ja PHP kääntäjät. (Portnoy 2012, 71–96.)

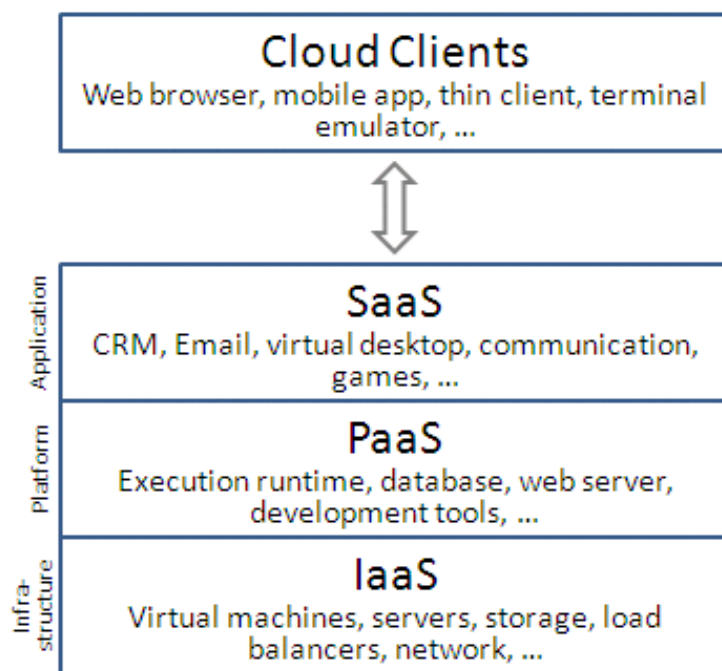
Yleisimpiä Linux-palvelinjakeluita ovat Red Hat, Ubuntu, Suse, Cent OS ja Debian. Linux -ympäristöissä on valtavat määrät erilaisia ohjelmistoja käytettävänä palveluiden toteuttamiseen. Näiden ympäristöjen muokattavuus ja avoimuus ovat niiden vahvuus ja heikkous. Avoimuuden takia niille tehdään paljon ohjelmistoja ja useat ihmiset kehittävät niitä. Koska ohjelmistoja on paljon ja erilaisia, tulee usein ohjelmistoja asennettaessa ristiriitoja ja toimintahäiriötä. Räättälöityvyyden takia tämä on harvoin este, vaan pikemminkin haaste, sillä räätälöinti vie aikaa ja aiheuttaa kustannuksia yritykselle. (Portnoy 2012, 97–124.)

Linux -jakeluilla on toimivat yhteisöt ja niissä paneudutaan ongelmiin tehokkaasti, jolloin voi löytyä ongelmiin useita ratkaisuja. Linux -järjestelmissä käytetään internetpalveluohjelmana pääsääntöisesti Apachea tai Nginx:ää. Sovelluspalvelimien puolelta löytyy lähes kaikki tunnetut ohjelmistokomponenttikirjastot ja nykyään myös ASP.NET, koska Microsoft vapautti ohjelmistokehyksen lähdekoodit avoimiksi syksyllä 2014. (Microsoft takes .NET open source and cross-platform, adds new development capabilities with Visual Studio 2015, .NET 2015 and Visual Studio Online. 2014.)

5.2 Virtuaalipalvelin järjestelmät

Virtuaaliset palvelinohjelmistot erottavat taustalla toimivan järjestelmän toimivasta ohjelmistosta, jolloin voidaan toteuttaa pilvipalvelimien kaltaisiakin palvelin-kokonaisuuksia. Pilvipalvelimissa on yhdistetty lukuisia konejärjestelmiä, jotta saataisiin nopeampaa ja luotettavampaa suorituskykyä palveluiden tuottamiseen. Pilvipalvelinjärjestelmässä konepalvelinten kuorma jaetaan tasaisesti kaikkien kesken, jotta yksittäinen palvelin ei ylikuormittuisi, vaan palvelu pysyisi sulavana. Virtuaalinen palvelin on yhden tai useamman palvelimen kanssa samassa hallintajärjestelmässä. Virtuaalipalvelin voidaan toteuttaa pilvipalvelimelle, mutta pilvipalvelua ei voida toteuttaa yhdelle virtuaaliselle palvelimelle. (Kavis 2014, 1–8.)

Pilvipalvelimella on kolme palvelumallia: IaaS (Infrastructure as a Service), PaaS (Platform as a Service) ja SaaS (Software as a Service), kuten kuvassa kolme esitetään. Jokainen palvelumalli lisää abstraktitason, kun siirrytään seuraavaan korkeampaan palvelumalliin. Mitä korkeampi abstraktitaso on, sitä vähemmän se asettaa vaatimuksia palveluntoteuttajan panostukseen tuottaa palveluita palvelumalliin. Korkeampi abstraktiotaso vaatii vähemmän palvelininfrastruktuurin ymmärtämistä ja antaa enemmän vapautta keskittyä palvelujen toteuttamiseen. (Kavis 2014, 13–22.)



KUVA 3. pilvipalvelimien rakenne (Bikeborg, 2015)

IaaS on pilvipalvelutasoista yksinkertaisin ja lähimpänä pilvilaskennan rajapintaa. Kaksi organisaatiota NIST (The National Institute of Standards and Technology) ja CSA (the Cloud Security Alliance) ovat määritelleet, mitä IaaS:n tulee olla. Molempien organisaatioiden määritelmät ovat samankaltaisia. Pilvipalveluita tilaava asiakas saa tässä palvelumallissa vapauden asentaa määrittämättömiä ohjelmia, hallita verkkoasetuksia ja tallennustilaa ilman, että järjestelmän fyysiseen infrastruktuuriin puututaan. IaaS -tason pilvipalvelutarjoajien, kuten IBM:n softlayerin,

ei tarvitse itse kuluttaa yrityksen resursseja välitaso-ohjelmoijien palkkaamiseen, vaan se voi keskittyä parempien IaaS palvelujen toteuttamiseen. (Kavis 2014, 39.)

PaaS on seuraava pilvipalvelutaso. Tällä tasolla kuuluu pystyä asentamaan ohjelmistoja ja sovelluksia ilman käyttöjärjestelmän tarvetta. NIST määrittelee PaaS:n olevan asiakkaalle tarjottua suorituskykyä pilvi-infrastruktuuriin asennetuille asiakkaan ohjelmoimille ohjelmille tai hankituille sovelluksille. Nämä on toteutettu käyttämällä tuottajan tukemia ohjelmointikieliä, kirjastoja, palveluja ja työkaluja. Asiakas ei hallinnoi alustana olevaa pilvi-infrastruktuuria, mukaan lukien verkkoa, tallennustilaa, käyttöjärjestelmää tai palvelimia. Kuitenkin asiakkaalle tulee jäädä kontrolloitavuus asennettuihin sovelluksiin ja asetusten säätämiseen sovellusten tarjontaympäristössä. (Kavis 2014 , 40.)

CSA:n määritelmä on vapaampi toteutustavaltaan ja palveluiden kattavuudeltaan. CSA:n määrittely käsittää, että pilvipalveluntarjoaja antaa palvelunostajalle laskenta-alustan ja ratkaisupinon palveluna. Tarjonnan kuuluu helpottaa sovellusten käyttöönottoa ilman, että käyttöönottajalla pitää miettiä fyysisten alustojen hallinnan ja oston tuomia kuluja ja monimutkaisuutta. Nykyään pilvipalvelujen tarjoajat ovat alkaneet yhdistää samaan ympäristöön IaaS ja PaaS tasoja. Esimerkiksi Microsoft Azure tarjoaa sekä virtuaalitietokoneiden, että pienten sovellusten toteuttamista. (Kavis 2014, 40.)

SaaS on käyttäjille kaikkein tunnetuin palvelutaso, mikä mielletään yleensä pilvipalveluksi. SaaS tasolla toimivat Dropbox:n, OneDrive:n ja Google apps:n kaltaiset sovellukset. SaaS on valmis sovelluspalvelu, joka toimittaa valmiin sovelluksen palveluna palvelujen käyttäjille. Sovelluksen käyttäjän tulee vain säätää sovelluskohtaisia asetuksia ja hallinnoida käyttäjiä. Tyypillisimpiä SaaS palveluita yrityksille ovat CRM- (Customer Relationship Management), ERP- (Enterprise Resource Planning), palkanmaksu- ja kirjanpito-ohjelmistot. SaaS tyyppiset ratkaisut ovat erinomaisia yrityksille, joilla ei ole minkäänlaista kiinnostusta palvelinten ylläpitoon tai sovellusinfrastruktuuriin. Yritykselle on selkeää käyttää käyttöjärjestelmästä tai laitteesta riippumattomia selainpohjaisia sovelluksia. SaaS pal-

veluiden käytöstä maksetaan käyttösopimusmaksun mukaisesti, eikä se siten aiheuta yllättäviä kustannuseriä yrityksille. Palveluiden käyttäjien ei tarvitse huolehtia suorituskyvystä, sillä sovelluspalvelun tarjoajat huolehtivat siitä. (Kavis 2014, 41-42.)

6 TYÖN TOTEUTUS

6.1 Lähtötilanne

Yrityksellä oli virtuaalipalvelin toteutus Oulu DC:n tarjoamana ja Microsoft Bizpark-tunnukset käytössään. Domain-nimipalvelut olivat Planeetan ja OVH:n tarjoamat. Käytettävät palveluohjelmat olivat toteutettu ASP.NET ja Java pohjaisina. Yrityksen nettisivut haluttiin toteuttaa PHP -pohjaisena Javascript liitännäisinä. Virtuaalipalvelimella suoritettiin kahta eri palvelinta Windows-server 2012 r2 jakelua ja Debian 6.0 Squeeze. Oulun Datakeskus tarjosi virtuaalisten palvelimien hallintaan VMware webhallintaympäristön. Olemassa olevat palvelimet olivat siinänsä oikein hyviä, mutta yritys halusi kaikki palvelinpalvelut yhden järjestelmän alle. Tällä saavutettaisiin parempi hallittavuus ja estettäisiin yhden ohjelmavirheen kaatavan koko verkkopalvelujärjestelmän.

Debian-ympäristön alla pyöri yrityksen varmuuskopiointipalvelut ja Java EE pohjainen sovelluspalvelin GlassFish. Yrityksen asiakkaiden websovellusympäristöt olivat toteutettu GlassFishiin, missä testiympäristöt ja käytössä olevat sovellukset pyörivät omissa virtuaalipalvelin instansseissaan. Käytössä ollut webpalvelinohjelmisto oli Apache ja webpalveluiden tietokantana MySQL.

Windows-ympäristössä suoritettiin yrityksen pääpalvelutarjonta eli GBCAD (Group Builder Computer-aided Design), GBCore (Group Builder Core), GB3D (Group Builder 3D) ja BIM (Builded Industrial Models) -mallien varastot sekä muunnosohjelmat. GBCAD oli asiakasohjelmisto, jolla pystyttiin mallintamaan huoneistoja. Tämä ohjelma toimi asiakaspäätteellä ja sillä voitiin tuoda yrityksen palvelimelle ohjelmalla käsitellyt mallit.

Yritys kehitti oppilasprojektina Unity-engine pohjaista 3D-mallien tarkasteluohjelmaa. Ohjelman tarkoituksena olisi toimia selaimen kautta, ja sen avulla voitaisiin tutustua asiakkaiden malleihin interaktiivisesti.

6.2 Kokeilut eri palvelinvaihtoehtoista

Palvelinratkaisun haluttiin olevan joustava ja edullinen. Henkilöstön palkkaaminen palvelinten ylläpitoon ei ole pienyritykselle kannattavaa, joten järjestelmän täytyy olla mahdollisimman varma ilman jatkuvaa ylläpitoa. Vaihtoehtoina lähdettiin suunnittelemaan pilvipalvelimia, omistuspalvelimia ja näennäispalvelimia, joihin tarpeen mukaan voidaan käyttää palvelinympäristöinä vanhoja vakaita palvelinympäristöjä tai valmistajien vakaiksi lupaamia ympäristöjä. Yrityksen käyttöön rakennettiin vanhoista kannettavista tietokoneista palvelinohjelmistojen testausalustoja. Palvelinohjelmistoihin laitettiin ajoon yrityksen palvelujen testiversioita ja muita kokeellisia palvelinohjelmia. Käyttämällä palvelimia aktiivisesti päivittäin, saatiin todennettua niiden toimivuutta tavallisessa käytössä. Lisäksi palvelimista täytyi testata tietoturvallisuus, minkä takia toteutettiin kokeellisia hyökkäyksiä ja nuuhkimisia.

Kannettaviin tietokonepalvelimiin asennettiin Windows ja Linux -ympäristöjä mahdollisimman suuren otannan kerryttämiseksi. Windows-järjestelmistä kokeilussa oli Windows Server 2012 r2, mikä oli jo yrityksen käytössä. Linux-järjestelmistä kokeiltiin Cent OS, Debian 7 Wheezy, Debian 8 Jessie ja Ubuntu Server 14.04 LTS. Windows-ympäristön etu oli testien perusteella se, että ympäristön kaataminen ylitsepääsemättömään tilaan oli vaikeaa ja vahingossa sitä ei todennäköisesti saisi edes toteutumaan. Ylitsepääsemättömällä tilalla tarkoitetaan sitä, että palvelin ei toteuta tarkoitustaan ja helpoin tie tilanteen korjaamiseksi on uudelleenasennus. Linux-ympäristöissä järjestelmätiedostoja pääsi muokkaamaan vapaammin ja jotkut ohjeet siihen jopa kehottivat. Tästä syystä Linux oli helpompi saattaa ylitsepääsemättömiin ongelmiin kuin Windows, mutta toisaalta tarvittaessa Windowsin järjestelmätiedostoja ei ole aivan yhtä helppo muokata. Näissä ympäristöissä tutustuttiin IIS:n, Nginx:n ja Apachen toimivuuksiin osoittaa ja reitittää palvelupyyntöjä. IIS:n ja Windows-palomuurin sääntöjen määrittäminen oli helppoa mukana olevan graafisenkäyttöliittymän takia. Hyviä hallintaympäristöjä oli saatavilla sekä Apacheen että Nginx:än järjestelmän käsittelyä varten. Työtä varten saatiin hyvä tarjous oululaiselta datakeskusyritykseltä palvelinratkaisun toteuttamiseen. He olisivat olleet valmiita rakentamaan laitteistoratkaisun verkkoliittymällä kohtuulliseen hintaan niin, että ohjelmistollinen puoli jäisi Group

Builderin vastuulle. Tämä tarjous kuitenkin myöhemmin hylättiin, koska haluttiin mahdollisimman helppo hallintaympäristö yrityksen käyttöön.

Suunnitelmissa oli kokeilla kannettavilla tietokoneilla tietokoneiden virtuaalista klusteri-palvelinjärjestelmää. Tätä ei ollut tarpeellista testata, koska jo peruspalvelin kuluttaisi liikaa resursseja ylläpidollisissa toimissa. Peruspalvelimien testaamisesta kerättiin paljon oppia, jota voitiin myöhemmin hyödyntää, kun siirryttiin pilvipalvelimien testaamiseen. Peruspalvelin testaamisesta tiedettiin ohjelmistotarpeet ja ohjelmistotuennan tarpeet, jotta voitiin toteuttaa kaikki yrityksen palvelinpalvelut. Seuraavaksi oli vuorossa pilvipalvelimien testaaminen ja kokeeseen valittiin AWS (Amazon Web Services), IBM softlayer, Jelastic ja Microsoft Azure. Azurea kokeiltiin ensimmäisenä. Azuren pilvipalvelin oli jo luotu yrityksen käyttöön ja siellä ajettiin Wordpress-verkkosovellusta. Wordpress-sovellukseen luotiin kokeellinen yrityksen nettisivu ja sen rajoja koeteltiin. Hallinta oli todella hyvä, mutta suorituskyky jätti valtavasti toivomisen varaa. Azure hylättiin, koska Wordpress-sovellus meni täysin lukkoon ja ei vastannut enää pyyntöihin. Vasta myöhemmin huomattiin, että Wordpress-sovellusta oli ajettu pienimmällä mahdollisella suorituskykysopimuksella.

Työssä kokeiltiin IBM softlayeriä ja AWS:ää pikaisesti, sillä aika alkoi olla lopussa ja palvelinratkaisun julkaiseminen lähestyi. Näitä ennätettiin kokeilla käyttöliittymien ja palveluiden liittämisen arvioimisessa, mutta ei varsinaisesti testaamisessa. Jelastic on dynaaminen palvelinratkaisu, missä resursseja vapautuu palvelimen käyttöön joustavasti. Planeetta niminen yritys tarjoaa Suomessa Jelastic-ympäristöön pohjautuvia ratkaisuja. Jelasticissa oli hyvin valittavissa ohjelmistokomponentit ja se olisi sopinut hyvin yrityksen käyttöön. Tähän ei kuitenkaan päädytty, sillä se olisi voinut tulla arvaamattoman kalliiksi. Alkuperäisessä järjestelmässä tarkasteltiin lokeja ja ohjelmien ajoja. Ohjelmistojen virheet saattoivat silloin tällöin ajaa palvelimen jumiin ja tällöin suoritin saattoi toimia täydellä suorituskyvyllä pitkiä aikoja ennen kuin virhe huomattiin. Dynaamisesti suorituskykyä nostava palvelu laskuttaa resurssien käytöstä ja tällöin ohjelmistovirheet voisivat tulla erittäin kalliiksi pitkällä aikavälillä.

6.3 Palvelintoteutus Azureen

Azuren päätyminen palvelin pohjaksi oli loppujen lopuksi selvä päätös. Group Builderin tiimissä työskenteli vakituisesti henkilö, jolla oli kokemusta Microsoft Azure -ympäristöstä ja siihen palveluiden toteuttamisesta. Tällä henkilöllä oli jo oma kokeiluympäristö olemassa ja hän oli toteuttanut sinne Wordpress-verkkosovelluksen yrityksen käyttöön. Aiemmin työn tutkimusvaiheessa Azure oli tullut hylätyksi, mutta opettajia konsultoidessa havaittiin, että Azurea ei ollut määritetty oikein. Kun Azureen tutustuttiin enemmän, huomattiin sen selkeys ja vaivattomuus palvelinten ja palveluiden lisäämisessä, sekä niiden määrittämisessä. Koska yritys oli startup-vaiheessa, oli sillä mahdollisuus hankkia omat Bizpark-tunnukset Microsoftilta. Tämä vaihtoehto mahdollistaa 750\$ kuukausittaisen käyttövaran kolmeksi vuodeksi jaettuna viiden kehittäjän kesken. Ratkaisu takaa yritykselle kolmen vuoden rauhallisen kasvuajan ilman, että palvelinympäristön kustannuksista joutuisi huolehtimaan.

Yrityksen palvelinpalvelut siirrettiin kaikki Azureen ja DNS-nimipalvelut käännettiin yhdistämään Azuren palvelimiin. Tämä toteutettiin aikaan, jolloin asiakkaat käyttivät mahdollisimman vähän palveluita. Tästä huolehdittiin myös kertomalla asiakkaille tulevasta palvelukatkoksesta ja palveluiden sujuvasta jatkumisesta siirtymän jälkeen. Tästä lähtien yrityksen on helppo lisätä uusia palveluita, eikä heidän tarvitse käyttää enää palvelinkäyttöjärjestelmää, elleivät he itse sitä halua. Azuresta he pystyvät seuraamaan sovelluksien resurssien käyttöä ja hallinnoimaan helpommin palveluita sovelluskohtaisesti. Azure-toteutukseen oli tavanomaisia palvelimia helpompi tutustua ja näin ollen yrityksellä on jatkossa helpompi kouluttaa käyttäjiä ja kehittäjiä.

7 PALVELINKEHITYKSEN TULEVAISUUS

Palvelimet ovat kehittyneet tekniikan läpimurtojen vaikutuksesta ja yleensä korkeelliset konesalit ovat olleet ensimmäisiä uusien tekniikoiden testaajia. Tämä piirre tuskin katoaa, koska palvelimien rakentamiseen käytetään enemmän rahaa kuin yhteenkään muuhun tietokonejärjestelmään. Palvelimien kehittyminen on alkanut saada vaikutteita muista järjestelmistä, kuten näytönohjaimista, pelikonsoleista ja mikrokontrollereista. IBM:n Sony Playstation 3 pelikonsoleille valmistama PowerXCell 8i -mikroprosessori liitettiin supertietokone IBM Roadrunnerin toiseksi pääprosessorisirujen perheeksi AMD:n opteron prosessorien kanssa (IBM Roadrunner. 2015). IBM:n Roadrunner oli ensimmäinen supertietokone, joka ylitti yhden petaflopsin (10^{15}) suorituskyvyn. Sen jälkeen on nähty sekä korkeellisia, että aktiiviseen käyttöön päätyneitä sulautettuja- ja klusterijärjestelmiä, joilla on perinteisistä järjestelmistä poikkeavat tekniset ratkaisut. Onnistuneiden Playstation 3 klustereiden kokeilut johtivat mikrokontrolleri klustereihin ja ne taas erikoistuneiden sirujen klustereihin kuten bitcoin-klustereihin.

Ympäristö ja käyttäjien vaikutteet ovat viime vuosina muokanneet käsityksiä palvelimista. Ensimmäistä kertaa palvelimia toteutetaan kuluttajien tarpeiden mukaisesti tai erikoispalvelujen toteuttamista varten. Esimerkiksi pelikonsoleiden peleissä yksi pelikonsoli voi ilmoittaa itsensä pelipalvelimeksi, johon muut liittyvät pelaamaan. Peliyhtiöiden palvelimien ei enää tarvitse toteuttaa pelipalvelinta, vaan ne tarjoavat yhdistämispalveluita pelikonsoleiden välille ja pelaajarekisterien ylläpitoa. Täten peliyhtiöt säästävät valtavasti rahaa ja resursseja.

Pitkään aikaan ei tulla näkemään yritysten kaikkien palvelinten siirtymistä suur-yritysten palvelinsaleihin, mutta pilvipalvelimista tulee kiihtyvällä tahdilla perinteisten palvelimien korvaava ratkaisu. Tätä kehityssuuntaa jarruttaa pääsääntöisesti tietoturvaan liittyvät pelot ja epäluulot suuryrityksiä kohtaan. Toisaalta myös palvelinten monipuolisempi räätälöinti käyttäjän tarpeita varten ja palvelimen fyysinen omistaminen vaikuttavat yritysten palvelinratkaisuista tehtäviin päätöksiin. Euroopan Unionin ja valtioiden lakipykälät palveluista, tietojen säilyttämisestä ja henkilörekistereistä ovat tiukasti määritettyjä, eikä edes saman yrityksen sisällä

ole sallittua siirtää kaikkea tietoa valtioiden rajojen yli, henkilötietolain pykälien §22 ja §23 mukaan (L 22.4.1999/523, 1999).

Tulevaisuudessa koneissa ei enää käytetä perinteisiä tekniikoita. Tarjolla on kvanttietokoneita, parempia puolijohteita, tietoliikenneyhteyksiä ja kenties suprajohtavia materiaaleja. Komponenttikehitys saavuttaa nykyisten tekniikoiden teoreettisen maksimiviivanleveyden piakkoin, joten uudet tekniikat tulevat muuttamaan melkoisesti nykyisiä tunnettuja järjestelmiä teknisestä näkökulmasta. Uusia teknisiä ratkaisuihin on lukuisia, mutta kaikki ovat enemmän tai vähemmän nykyjärjestelmistä poikkeavia.

Tämä tuo mielenkiintoisen kysymyksen, pystytäänkö teknisten ratkaisujen muuttumisesta huolimatta pitämään ohjelmointi nykyisen kaltaisena ja säilyttämään käytetyt järjestelmät. Tämä riippuu siitä, miten hyvin rajapinnat saadaan toteutettua ja säilyykö komponenttien hyödyntäminen tehokkaana. Voi olla, että tekniikan muuttumisen takia koetaan muutokset aina sovellustasolla asti, mutta todennäköisempää on, että nykyiset sovellukset vain yksinkertaisesti jäävät käytöstä pikkuhiljaa pois.

8 POHDINTA

Opinnäytetyön tavoitteena oli hakea erilaisia palvelinratkaisuja pienten- ja keski-suurtenyritysten käyttöön. Työ ei varsinaisesti ole opas yritykselle palvelinympäristöntekemiseen. Tästä työstä voisi jatko kehittää yksityiskohtaisemman ja laajemman oppaan tai jopa teoksen pienille ja keskisuurille yrityksille.

Työn tutkimusosuus ja kerronta ovat suhteellisen suppea kokonaisuus kaikesta palvelinratkaisujen toteuttamisen mahdollisuuksista. Tässä työssä pyrittiin kuitenkin tuomaan mahdollisimman laaja käsitys palvelinten toiminnasta ja helpottamaan ratkaisun etsimistä omaan ja yrityksen käyttöön. Tekniikka kehitty ja vaihtoehdot muuttuvat, siksi ei kannata kirjoittaa liian tarkkaa teknistä ohjetta, vaan on parempi jos pystytään laajentamaan lukijan ymmärrystä aiheesta. Toivon, että mahdollisimman moni tätä työtä lukenut innostuu kokeilemaan oman palvelimensa tekemistä. Kenties lukija innostuu mukaan palvelinten kehittämiseen tai ainakin ymmärrys laajenee ja palvelimet eivät tunnu enää vierailta. Aiheesta löytyy paljon tietoa internetistä ja siitä on kirjoitettu paljon eri vaikeustasoisia kirjoja. Suosittelenkin tutustumaan Micheal J. Kaviksen kirjoittamaan teokseen *Architecting the Cloud*, jos pilvipalvelut ja niiden toteuttaminen kiinnostaa.

Vain reilu puolivuotta ennen työni aloittamista, minulla ei ollut juuri kokemusta palvelimista tai niiden toteuttamisesta. Olen valtavan nopeasti pystynyt omaksumaan tietoa palvelinten toiminnasta, käytettävistä protokollista ja standardeista. Omalla kohdallani omaksumisessa auttoi vahva tausta tietotekniikasta ja siirtojärjestelmistä sekä oma uteliaisuuteni ja tiedonhalu. Vastuunottaminen loi paineita, mutta antoi perääntymättömän motivaation edetä tehtävässä. On rohkeaa lähteä näin pienellä kokemuksella mukaan lähes tuntemattomaan ja ottaa vastuu onnistumisesta. Voinkin hyvillä mielin kehottaa tekstiä lukevia herättämään omaa uteliaisuuttaan, kokeilun halukkuuttaan ja rohkeuttaan uskaltaa.

Työn toteuttaminen oli mahtava tilaisuus tutustua lukuisiin tekniikkoihin palvelinpuolella ja kokeilemaan asioita, joista osa oli niin uusia, ettei niistä ollut vielä edes kirjoja olemassa. Tämä tilaisuus antoi innostuksen uusiin vapaa-ajan harrastuk-

siin huomattuani, kuinka mielenkiintoista palvelinten toteuttaminen on. Palvelinala on haasteellinen ja siinä mestariksi tulemiseen on pitkä taival, mutta kynnys lähtee mukaan palvelinalalle on pieni.

LÄHTEET

- AMD Server Processors. 2015. AMD. Saatavissa: <http://www.amd.com/en-gb/products/server>. Hakupäivä 1.12.2015.
- Barroso, L. A. – Clidaras, J. – Hölzle, U. 2013. The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines. Madison: Morgan & Claypool Publishers.
- Bartolini, C. – Gasparly, L. P. 2009. Integrated Management of Systems, Services, Processes and People in IT. Venice: Springer Science & Business Media.
- Bikeborg 2015. File:Cloud computing layers.png. Wikimedia commons.Saatavissa: https://en.wikipedia.org/wiki/Cloud_computing#/media/File:Cloud_computing_layers.png. Hakupäivä 19.11.2015.
- Brouwer, A. 2013. 2. Properties of partition tables. Saatavissa: http://www.win.tue.nl/~aeb/partitions/partition_types-2.html. Hakupäivä 2.12.2015.
- Brown, Michael. How to choose a server for your small business. PCworld. Saatavissa: http://www.pcworld.com/article/251993/how_to_choose_a_server_for_your_small_business.html. Hakupäivä 2.12.2015.
- Chevance, R. J. 2004. Server Architectures: Multiprocessors, Clusters, Parallel Systems, Web Servers, Storage Solutions. Burlington: Digital Press.
- Dostálek, L. – Kabelová, A. 2006. Understanding TCP/IP. Birginham: Packt Publishing.
- Erkens, R. – Haustein, N. – Müller-Friedt, W. – Troppens, U. – Wolafka, R. 2009. Storage Networks Explained: Basics and Application of Fibre Channel SAN, NAS, iSCSI, Infiniband and FCoE. Heidelberg: John Wiley & Sons.

Hausman, K. – Cook, S. L. – Sampaio, T. 2013. Cloud Essentials: CompTIA Authorized Courseware for Exam CLO-001. New Jersey: Wiley.

IBM Roadrunner. 2015. Wikipedia. Saatavissa:

https://en.wikipedia.org/wiki/IBM_Roadrunner. Hakupäivä 19.11.2015.

Kamula, M. 2010. Towards Green IT. Luentomateriaali opintojaksolla. Oulu: Oulun seudun ammattikorkeakoulu, liiketalouden yksikkö.

Kavis, M. J. 2014. Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS). New Jersey: Wiley.

L 18.9.2015/1141. 2015. Tilintarkastuslaki. Saatavissa:

<http://www.finlex.fi/fi/laki/alkup/2015/20151141>. Hakupäivä 19.11.2015.

L 22.4.1999/523. 1999. Henkilötietolaki. Saatavissa:

<http://www.finlex.fi/fi/laki/ajantasa/1999/19990523>. Hakupäivä 19.11.2015.

L 30.12.1997/1336. 2015. Kirjanpitolaki. Saatavissa:

<http://www.finlex.fi/fi/laki/ajantasa/1997/19971336>. Hakupäivä 19.11.2015.

Microsoft takes .NET open source and cross-platform, adds new development capabilities with Visual Studio 2015, .NET 2015 and Visual Studio Online. 2014. Microsoft News Center.

Saatavissa: <http://news.microsoft.com/2014/11/12/microsoft-takes-net-open-source-and-cross-platform-adds-new-development-capabilities-with-visual-studio-2015-net-2015-and-visual-studio-online/>. Hakupäivä 16.12.2015.

Palkkalaskuri. 2015. Yrittäjäjien liitto. Saatavissa: <http://www.yrittajat.fi/fi-FI/palkkalaskuri/>. Hakupäivä 19.11.2015.

Portnoy, M. 2012. Essentials: Virtualization Essentials. Hoboken: Sybex.

Processors. Intel ARK. Saatavissa: <http://ark.intel.com/>. Hakupäivä 1.12.2015.

Rackley, S. 2007. Wireless Networking Technology: From Principles to Successful Implementation. Oxford: Newnes.

- Seitz, J. 2014. Black Hat Python: Python Programming for Hackers and Pentesters. San Fransisco: No Starch Press.
- Shostack, A. 2014. Threat Modeling: Designing for Security. Indiana: John Wiley & Sons.
- Singh, V. P. – Singh, M. 2009. Computer Hardware Course. New Delhi: Computech Publications Limited.
- Själänder, M. – Martonosi, M. – Kaxiras, S. 2014. Power-Efficient Computer Architectures: Recent Advances. Madison: Morgan & Claypool Publishers.
- Spafford, G. 2009. Greening the Data Center: A Pocket Guide. Cambridgeshire: IT Governance Publishing.
- Tietotekniikan palvelualan työehtosopimus. 2013. Teknologiateollisuus ry, Ylemmät toimihenkilöt YTN ry, Tietoalan toimihenkilöt ry. Saatavissa: http://tietoala.fi/rauta/wp-content/uploads/2015/11/TES_2013_fi_verkkoversio.pdf. Hakupäivä 10.12.2015.
- Tuloslaskelma. 2015. Wikipedia. Saatavissa: <https://fi.wikipedia.org/wiki/Tuloslaskelma>. Hakupäivä 19.11.2015.
- Tuotepaketit. 2015. OVH-hosting.fi. Saatavissa: <https://www.ovh-hosting.fi/dedicated-cloud/tuotepaketit/>. Hakupäivä 19.11.2015.
- Young, R. 2009. How Computers Work: Processor and Main Memory. Roger Young: How Computers Work.

Opinnäytetyön selvitys

Tilaaaja:

GroupBuilder OY

Harri Majala

Marko Savolainen

Opinnäytetyön aihe:

GB mahdollistaa ihmisille joko uuden tai olemassaolevan kodin suunnittelun, sisustamisen ja ylläpidon virtuaalisessa 3D maailmassa.

GB:n tarve palvelin resursseista kasvaa jatkuvasti ja meidän täytyy pystyä rakentamaan elastinen helpposti skaalautuva virtuaalipilvi, joka pystyy hostaamaan valtavan suuret volyymit koteja. BimVault (3d-mallit), Homebook (materiaalit, 3d-assetit, tietomalli), GB Core (viestintä, liitteet, kotien sen hetkinen "status" ja historia) pääkomponentteina. Lopputyön tarkoitus olisi löytää tekninen ratkaisu, joka pystyisi mahdollisimman kattavasti nyt ja tulevaisuuden näkökulmasta toteuttamaan parhaan ratkaisun GB tarpeisiin.

Eli syvemmin aiheesta: tutkia palveluiden tarjoamiseen internetin ylitse erilaisilla metodiikoilla ja vertailla niiden mahdollisuuksia sekä rajoitteita. Palvelimet ja proxy; Ohjelmointi kielet, hallittavuus, käytettävyys ja sulautettavuus; Käytössä olevat sovelluksien liittäminen ja mahdollisten tulevien sovellusten toteuttaminen; GB tuote perheen laajentaminen muun muassa IoT (Internet of Thing) ja kodin hallinta webgl tai vastaavassa web GUI ympäristöstä.

Tiedossa olevat tutkittavat aiheet:

- Fyysiset palvelimet
 - OS
 - *Linux*
 - *Windows*
 - Yritykselle oma palvelin
 - *räkki*
 - *koteloitu*
 - Domain ympäristö pohjat
 - *Oracle*
 - *Apache*
 - *IIS*
- Elastiset pilvi palvelimet:
 - Azure
 - Jelastic
 - OuluDC
- web palvelin ohjelmistot
 - Apache
 - Lighttpd
 - Nginx
 - Bitnami
- sovellus palvelin ohjelmistot
 - Glassfish
 - Ruby
- Vaadittava tuettavuus, nykyiselle sovelluksille ja sivustoille
 - ASP.NET
 - Vaadin
 - Wordpress
 - Unity