Lauri Jääskeläinen

PALVELINKESKUSTEN ENERGIATEHOKKUUDEN PARANTAMINEN OHJELMISTORATKAISUILLA

JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2023

TIIVISTELMÄ

Jääskeläinen, Lauri
Palvelinkeskusten energiatehokkuuden parantaminen ohjelmistoratkaisuilla
Jyväskylä: Jyväskylän yliopisto, 2022, 32 s.
Tietojärjestelmätiede, Kandidaatin tutkielma
Ohjaaja(t): Clements, Kati

Palvelinkeskukset ovat kriittisiä infrastruktuureita verkkopohjaisille palveluille ja niiden tarpeen ennustetaan kasvavan entisestään tulevaisuudessa esimerkiksi pilvipalveluiden suosion lisääntyessä. Palvelinkeskukset kuluttavat paljon energiaa ja energiankulutus on yksi keskuksien suurimmista kulueristä. Energiatehokkuuden parantaminen on palveluntarjoajille edullista kulujen vähentyessä ja ympäristöystävällisyyden vuoksi. Tässä tutkielmassa tarkastellaan kirjallisuuskatsauksen muodossa kirjallisuudesta löytyviä ohjelmistoratkaisuita ja menetelmiä, joilla voidaan vähentää palvelinkeskusten energiankulutusta. Tärkeimmiksi menetelmiksi tunnistetaan virtuaalikoneiden sijoittaminen ja työkuorman tasapainottaminen. Fyysisten palvelimien resurssit jaetaan palvelinkeskuksissa usein sovelluksia ajavien virtuaalikoneiden kesken. Virtuaalikoneiden oikeanlaisella sijoittamisella voidaan maksimoida isäntäkoneen resurssienkäyttö ja fyysisten koneiden lepoaika ja säästää energiankulutuksessa. Työkuorman tasapainottamisella voidaan jakaa palvelinkeskukselle tuleva työkuorma tehokkaasti virtuaalikoneiden välille välttäen isäntäkoneiden ylikuormitus tai alikuormitus. Tarvittaessa virtuaalikoneita siirretään fyysisten koneiden välillä. Virtuaalikoneiden sijoittamiseen ja työkuorman tasapainottamiseen on kehitetty useita menetelmiä, joilla toteuttaa toiminnallisuudet mahdollisimman tehokkaasti.

Asiasanat: palvelinkeskus, virtualisaatio, energiatehokkuus

ABSTRACT

Last name, First name
Improving the energy efficiency of data centers with software solutions
Jyväskylä: University of Jyväskylä, 2022, 32 p.
Information Systems, Bachelor's Thesis
Supervisor(s): Clements, Kati

Data center are critical infrastructures for internet-based services and the need for data centers is predicted to increase in the future for the rise of cloud services and other similar technologies popularity. Data centers use lots of energy and energy usage is one of the greatest expenses for data centers. The improvement of data centers energy efficiency is beneficial to service providers for the decrease in the data center's running expenses and for the environmental benefits. This thesis is a literature review which studies software solutions presented in literature that can reduce the energy usage of a data center. The most prominent solutions presented in literature are virtual machine placement and load balancing. The resources of physical servers are shared between virtual machines that run applications. With efficient placement of the virtual machines the resource usage of the host machine and the sleep time of the physical machines, can be maximized and savings achieved in energy usage. Load balancing methods distribute the data center's workload efficiently between the virtual machines to avoid overloading or underloading the physical machines. If needed the virtual machines can me migrated between the physical machines. There have been numerous methods developed to place virtual machines and to execute load balancing efficiently in data centers.

Keywords: data center, virtualization, energy efficiency

KUVIOT

KUVIO 1 Yksittäisen palvelimen arkkitehtuuri (Mishra, Sahoo & Parida, 2020) 15
KUVIO 2 Käyttäjien palvelupyyntöjen saapuminen työkuorman tasapainottajalle ja tasapainottajan sijainti palvelinkeskusarkkitehtuurissa (Afzal & Kavitha, 2019)..... 16

TAULUKOT

Taulukko 1 Energiankulutusta vähentävät ohjelmistoratkaisut 25

SISÄLLYS

1 JOHDANTO 7

2 PALVELINKESKUKSET 9
2.1 Palvelinkeskukset 9
2.2 Palvelinkeskuksen arkkitehtuuri - palvelinkeskusverkko 9
2.3 Palvelinkeskuksen arkkitehtuuri - ohjelmistot 11

3 PALVELINKESKUSTEN ENERGIANKULUTUS 13
3.1 Palvelinkeskusten energiankulutus 13
3.2 Järjestelmien energiankulutuksen ratkaisuita 13

4 PALVELINKESKUSTEN ENERGIANKULUTUKSEN VÄHENTÄMINEN OHJELMISTORATKAISUILLA 17
4.1 Virtuaalikoneiden sijoittaminen ja siirtäminen 17
4.2 Työkuorman tasapainottaminen 21

5 YHTEENVETO 28

6 LÄHTEET 30

1 Johdanto

Palvelinkeskukset ovat infrastruktuureita, joka sisältävät suuria määriä tietokoneita, joilla ajetaan palveluita. Ajettavia palveluita tarjotaan käyttäjille verkon ylitse. Palvelinkeskusten kriittisyys yhteiskunnalle kasvaa jatkuvasti pilvipalveluiden ja verkkopohjaisten sovellusten suosion kasvaessa. Palvelut tarvitsevat mm. laskentatehoa ja datan säilytystilaa, joita palvelinkeskukset tarjoavat. (Dayarathna, Wen & Fan, 2015)

Energiankulutus on yksi palvelinkeskusten suurimmista kulueristä (Dayarathna, Wen & Fan, 2015). Energiankulutuksen vähentäminen on tärkeää sekä taloudellisista syistä, että ympäristöystävällisistä syistä. Tutkimusten mukaan palvelinkeskukset hyödyntävät keskimäärin vain 30 % resursseistaan. Tyhjäkäynnillä oleva palvelin kuluttaa 70 % energiasta, mitä se kuluttaisi ollessaan täydellä käytöllä (Zhang ym., 2019). Tästä syystä on tärkeää maksimoida kunkin palvelimen resurssien käyttö ja sammuttaa tarpeettomat palvelimet energian säästämiseksi.

Palvelinkeskukset kuluttavat suuren määrän energiaa jopa kokonaisenergiankulutukseen nähden. Yhdysvaltojen palvelinkeskusten energiankulutusraportin (Shebabi ym. 2016) mukaan palvelinkeskusten energiankulutuksen osuus vuonna 2014 oli 1,8 % Yhdysvaltojen kokonaisenergiankulutuksesta. Palvelinkeskusten, ja samalla niiden energiankulutuksen, edelleen kasvaessa palvelinkeskusten energiatehokkuuden parantamisen merkitys nousee entisestään. Palvelinkeskusoperaattorit voivat pienentää energian kulutustaan 10–15 % optimoimalla resurssien hyödyntämisestä ja hallinnasta vastaavat toiminnot (Rong ym., 2016).

Tämän tutkimuksen tarkoituksena on tarkastella palvelinkeskusten energiankulutusta. Palvelinkeskuksen energiankulutukseen vaikuttavat laitteisto ja ohjelmistot. Tässä tutkimuksessa käsitellään erilaisten ohjelmistojen vaikutusta palvelinkeskusten energiatehokkuuteen ja laitteistoa käsitellään vain lyhyesti palvelinkeskuksen arkkitehtuuria tarkasteltaessa. Tarkoituksena on löytää ohjelmistoratkaisuita, joita käytetään palvelinkeskuksissa energiankulutuksen pienentämiseen. Tutkimuksessa käydään läpi ohjelmistojen toimintaperiaate, miten ohjelmistot vaikuttavat energiankulutukseen ja mahdollisuuksien mukaan, miten

paljon ne vaikuttavat palvelinkeskuksen energiatehokkuuteen. Tutkimuskysymykseksi muodostuu täten:

Miten palvelinkeskuksen energiankulutukseen voidaan vaikuttaa ohjelmistoratkaisuilla?

Tutkielma on toteutettu kirjallisuuskatsauksena. Lähdeaineisto on haettu Google Scholarista ja IEEE Xploresta. Hakusanoina on käytetty mm. data center, energy, cloud, load balancing, virtual machine ja architecture ja sanojen eri yhdistelmiä. Lähdeaineistoksi pyrittiin valitsemaan mahdollisimman uusia artikkeleita ja muutamaa poikkeusta lukuun ottamatta kaikki artikkelit ovat vuonna 2016 tai sen jälkeen julkaistuja. Kaikki artikkelit ovat julkaisuista, jotka ovat julkaisufoorumissa vähintään tasoa 1. Lähdeaineistoksi on valittu mahdollisuuksien mukaan artikkeleita, joilla on mahdollisimman paljon viittauksia, mutta aiheen tarkan rajauksen ja joidenkin artikkelien tuoreuden vuoksi tästä vaatimuksesta on joustettu.

Palvelinkeskusten energiankulutusta käsittelevässä kirjallisuudessa usein esiintyviä ohjelmistoratkaisuita ovat virtuaalikoneiden sijoittaminen ja siirtäminen ja työkuorman tasapainottaminen. Näitä tutkimuskysymykseen vastaavia ohjelmistoratkaisuita käsitellään neljännessä luvussa.

Tutkielman sisältö jatkuu ensimmäisessä sisältöluvussa palvelinkeskuksen määritelmällä ja palvelinkeskusten arkkitehtuurin läpikäynnillä. Vaikka tutkielma on rajattu palvelinkeskusten energiatehokkuuttaparantaviin ohjelmistoratkaisuihin, palvelinkeskusten arkkitehtuurissa käsitellään myös keskuksissa toimiva laitteisto, jotta voidaan ymmärtää ympäristöä, joissa ohjelmistot toimivat. Kolmannessa luvussa määritellään palvelinkeskusten energiankulutus ja esitellään lyhyesti kirjallisuudessa esiintyviä ratkaisuita energiatehokkuuden parantamiseen. Neljännessä luvussa tarkastellaan syvällisesti energiankulutusta vähentäviä ohjelmistoratkaisuita, eli virtuaalikoneiden sijoittamista ja työkuorman tasapainottamista. Aluksi selitetään menetelmien toiminta yleisesti ja lopuksi tarkastellaan tarkemmin eri tutkimuksissa kehitettyjä algoritmeja, joiden tarkoitus on parantaa energiatehokkuutta. Tutkielma päättyy yhteenvetoon, jossa kootaan tutkielmassa käsiteltyjä aiheita ja ehdotetaan mahdollisia tulevaisuuden tutkimusaiheita.

2 PALVELINKESKUKSET

Luvussa käsitellään palvelinkeskuksia yleisesti, jotta tarkasteltava toimintaympäristö on lukijalle tuttu. Ensin määritellään palvelinkeskus, jonka jälkeen käydään läpi palvelinkeskuksen arkkitehtuuria. Arkkitehtuuri käsitellään kahdessa osassa, joista ensimmäisenä käsitellään palvelinkeskusverkkoa ja siihen kuuluvia laitteita. Vaikka tutkielma on rajattu palvelinkeskusten ohjelmistoihin, on tärkeää ymmärtää laitteiden ja ohjelmistojen väliset suhteet, koska ohjelmistot toimivat laitteissa ja hallitsevat niiden toimintaa. Viimeisessä alaluvussa kerrotaan lyhyesti palvelinkeskuksen toimintaa ohjaavista ohjelmistoista.

2.1 Palvelinkeskukset

Palvelinkeskukset ovat infrastruktuureita, jotka tarjoavat internetpohjaisia palveluita suuressa mittakaavassa (Dayarathna, Wen & Fan, 2016). Palvelinkeskus sisältää siis resurssit, jotka ovat käytettävissä verkon ylitse maksua vastaan (Imran ym., 2022). Palvelinkeskukset sisältävät tietoa prosessoivan elektroniikan eli palvelimet, tiedon tallennuslaitteet ja kommunikaatio- eli verkkolaitteet (Shehabi ym. 2016).

Yhteiskunnan digitalisoitumisen myötä ja digitalisaation edelleen yleistyessä palvelinkeskusten määrä ja niiden koot kasvavat jatkuvasti (Shehabi ym. 2016). Kysyntä keskusten palveluille on kasvanut nopeasti ja uusien dataintensiivisten teknologioiden, kuten tekoälyn, kehitys lisännee kysyntää myös tulevaisuudessa (Masanet ym. 2020). Palvelinkeskuksia tarvitaan lähes kaikilla talouden osa-alueilla ja ne ovat välttämättömiä mm. kommunikaatio- ja liiketoimintajärjestelmille ja hallinnollisille järjestelmille (Shehabi ym. 2016). Palvelinkeskusarkkitehtuurissa on eroteltavissa kaksi tärkeää komponenttia, laitteisto ja ohjelmisto (Katal, Dahiya & Choudhury, 2022).

2.2 Palvelinkeskuksen arkkitehtuuri - palvelinkeskusverkko

Palvelinkeskuksen toiminnassa on tärkeässä roolissa palvelinkeskusverkko (Data center network). Palvelinkeskusverkolla on tärkeä osa keskuksen laskennassa ja laitteiden välisessä kommunikaatiossa. Palvelinkeskusverkko yhdistää palvelinkeskuksen fyysiset komponentit toisiinsa kyseiseen keskukseen määritellyn topologian mukaan ja palvelinkeskuksen internettiin. Keskuksen tehokkuus ja toiminallisuus riippuvat paljon palvelinkeskusverkon toiminnasta (Cheng, Gao & Cheng, 2016).

Palvelinkeskusympäristö asettaa verkolle tiettyjä haasteita ja vaatimuksia. Modernien palvelinkeskusten suuri mittakaava aiheuttaa vaikeuksia verkon

suunnittelussa. Palvelinkeskusverkkoon täytyy pystyä liittämään tuhansia verkkolaitteita ja rakenteen tulee skaalautua tarvittaessa uusia verkkolaitteita liitettäessä (Wang ym., 2015). Palvelinkeskukset isännöivät monipuolisesti erilaisia sovelluksia, jotka aiheuttavat vaihtelevaa liikennettä palvelinkeskusverkkoon, mikä pitää kyetä hallitsemaan. Verkkoliikenne pitää pystyä välittämään nopeasti ja luotettavasti (Wang ym., 2015). Palvelinkeskusverkon täytyy pystyä tarjoamaan palveluita luotettavasti ympäri vuorokauden. Verkon toimintaa täytyy tarkkailla jatkuvasti ja häiriötilanteista pitää pystyä toipumaan nopeasti. (Xia, Zhao, Wen & Xie, 2016)

Palvelinkeskusverkossa toimivia laitteita ovat kytkimet, palvelimet, tiedon tallennusjärjestelmät, telineet (rack) ja kaapelit. Kytkimet kytkevät laitteita samaan verkkoon ja välittävät liikennettä. Palvelimet tarjoavat alustan virtuaalikoneille ja sovelluksille ja käsittelevät dataa. Tallennusjärjestelmät tallentavat palvelimien tietoa. Tiedon tallennukselle on tavallisesti kaksi vaihtoehtoa, network-attached-storage (NAS), johon tallennetaan tietoa verkon palvelinten yhteiskäyttöön ja storage area network (SAN), joka on palvelimiin yhdistettävä tallennustila. Telineet sisältävät edellä mainitut laitteet niiden hallinnan helpottamiseksi ja tilan säästämiseksi. Kaapelit yhdistävät palvelinkeskusverkon laitteet toisiinsa. (Cheng, Gao & Cheng)

Zhang ym. (2018b) käsittelevät tutkimuksessaan useita eri palvelinkeskusarkkitehtuureita, jotka voidaan jakaa kategorioihin kytkinkeskeiset-, palvelinkeskeiset- ja hybridiarkkitehtuurit. Kytkinkeskeiset ja palvelinkeskeiset arkkitehtuurit eroavat toisistaan siinä missä liikenteen reitittäminen ja verkon hallinta suoritetaan. Kytkinkeskeisissä arkkitehtuureissa kytkimet muodostavat reittejä palvelimien välille ja suorittavat pakettien eteenpäin lähetyksen. Palvelinkeskeisessä arkkitehtuurissa sen sijaan palvelimet suorittavat sekä laskennan, että toimivat kytkiminä yhdistyen muihin palvelimiin. Hybridiratkaisussa sekä kytkimiä, että palvelimia käytetään pakettien välittämiseen. (Zhang ym., 2018b) Palvelinkeskusverkon tarkemmalle rakenteelle on kehitetty kymmenittäin erilaisia arkkitehtuureja, joilla on omat etunsa ja heikkoutensa. Verkkoa suunniteltaessa on otettava huomioon sovellusten verkkoliikenne ja tehtävä kompromisseja kyseiseen keskukseen sopivan rakenteen suunnittelemiseksi. (Cheng, Gao & Cheng, 2016) Tutkimuksissa on havaittu palvelinkeskusten verkkoliikenteellä olevan tiettyjä ominaispiirteitä liittyen datavirtojen kokoon ja määrään, joista on hyötyä palvelinkeskusverkon suunnittelussa. Myös linkkien käyttöasteen ja ruuhkapaikkojen tutkimisesta on hyötyä. (Zhang ym., 2018b)

Palvelinkeskusverkon pohjimmainen tarkoitus on ohjata verkkoliikennettä lähteestä oikeaan kohteeseen. Perinteisesti viestin eteenpäin lähetys on toteutettu hajautetusti siten, että kukin kytkin oppii verkon rakenteen toisten verkkosolmujen kanssa vaihdettujen viestien avulla ja rakentaa tiedoista viestien välitystaulun, jota se käyttää tiedon eteenpäin lähettämiseen (Xia, Zhao, Wen & Xie, 2017). Käytännössä viestien välitykseen käytetään useita eri protokollia ja menetelmiä, jotka ovat tämän tutkielman aiheen ulkopuolella.

Tulevissa luvuissa tarkastellaan ohjelmistoratkaisuiden vaikutusta palvelinkeskusten energiankulutukseen. Palvelinkeskusten energiankulutukseen

vaikuttavia ohjelmistoja, kuten työkuorman tasapainottajia kehitettäessä, on otettava huomioon palvelinkeskuksen topologia ja sovellusten verkkoliikenne (Zhang ym., 2018b), joten on hyvä ymmärtää keskuksen laitteiden toiminnasta ja liikenteen hallinnasta erilaisissa arkkitehtuuriratkaisuissa.

2.3 Palvelinkeskuksen arkkitehtuuri - ohjelmistot

Palvelinkeskusten ohjelmistokomponentit vaativat paljon energiaa tehtävien suorittamiseen. Vaikka palvelinkeskuksen fyysiset komponentit varsinaisesti kuluttavat energian, luovat ohjelmistot energiankulutukselle tarpeen. Kehitettävien ohjelmistojen tulisi pystyä hyödyntämään laitteistossa tapahtuvaa kehitystä, jotta laitetasolla tapahtuva edistys ei valuisi hukkaan ja energiankulutus pysyisi kehityksestä huolimatta korkeana. Eniten energiaa kuluttava komponentti riippuu suoritettavan tehtävän tyypistä. Laskentaa vaativat tehtävät kuluttavat paljon prosessorin tehoa, dataa käsittelevät tehtävät käyttävät järjestelmän tiedon tallennusjärjestelmiä ja paljon kommunikaatiota vaativat tehtävät kuluttavat verkkolaitteiden resursseja. (Katal, Dahiya & Choudhury, 2022)

Palvelinkeskuksen ohjelmistokerros on jaettu Katalin, Dahiyan & Choudhuryn (2022) tutkimuksessa kolmeen alikerrokseen, joita tarkastellaan energiankulutuksen vähentämisen näkökulmasta. Alikerrokset ovat käyttöjärjestelmä, virtualisaatiokerros ja sovellukset. Käyttöjärjestelmä on ohjelmisto, joka toimii arkkitehtuurisesti palvelimella laitteiston ja sovellusten välissä. Käyttöjärjestelmän tehtävä on hallita palvelimen resursseja ja tarjota resursseja sovelluksille. Virtualisaatiolla tarkoitetaan fyysisen koneen resurssien jakamista ohjelmiston avulla useisiin isäntäkoneen käyttöjärjestelmässä toimiviin virtuaalikoneisiin. Virtuaalikoneiden luomisesta ja hallinnasta vastaa ohjelmisto nimeltään *Virtual Machine Monitor* tai *Hypervisor*. Sovellukset toimivat virtuaalikoneissa ja sisältävät palveluiden varsinaiset toiminnallisuudet. (Katal, Dahiya & Choudhury, 2022)

Viimeaikaisessa kirjallisuudessa paljon käsiteltyjä palvelinkeskusten energiankulutusta pienentäviä ohjelmistoratkaisuita, joita tässä tutkielmassa tarkastellaan, ovat virtualisaatio (Mishra ym., 2018) ja työkuorman tasapainottaminen (Zhang ym. 2018b). Virtualisaatiota pidetään yhtenä tärkeimmistä tekniikoista moderneiden pilvipalveluiden toiminnassa (Katal, Dahiya & Choudhury, 2022). Virtualisaation tehokas hyödyntäminen edellyttää palvelinkeskukselta algoritmia parhaiden virtuaalikone-isäntäkone- ja tehtävä-virtuaalikone-parien löytämiseksi (Mishra ym., 2018). Yksi virtualisaation mahdollistamista eduista on virtuaalikoneiden siirtäminen fyysisten koneiden välillä. Suurin osa hypervisoreista tukee virtuaalikoneiden siirtämistä (Noshy, Ibrahim & Ali, 2018). Syynä virtuaalikoneen siirrolle voi olla esimerkiksi vähän käytettyjen fyysisten koneiden palveluiden siirtäminen yhteen fyysiseen koneeseen tai resurssien käytön muuttuessa ylikuormittuneen koneen kuorman keventäminen. (Katal, Dahiya & Choudhury, 2022)

Työkuorman tasapainottaja on ohjelmisto, joka toimii palvelimilla ja tasaa työkuorman virtuaalikoneiden välillä. Tasapainottaja saa palvelupyynnöt palvelinkeskuksen kontrollerilta, johon ne tulevat käyttäjiltä palvelinkeskuksen ulkopuolelta. (Kumar & Kumar, 2019) Työkuorman tasapainottajan tarkoitus on ohjata tehtäviä maksimoiden verkon hyödyntäminen ja parantaa samalla sovellusten suorituskykyä (Zhang ym., 2018b).

3 Palvelinkeskusten energiankulutus

Tässä luvussa määritellään palvelinkeskuksen energiankulutus ja esitellään lyhyesti viimeaikaisessa kirjallisuudessa esiintyviä energiankulutusta vähentäviä ohjelmistoratkaisuita.

3.1 Palvelinkeskusten energiankulutus

Palvelinkeskuksen energiankulutuksesta suurin osa muodostuu palvelinten käynnissä pitämisestä ja niiden jäähdyttämisestä. Vaikka fyysiset komponentit kuluttavat energian on tärkeää huomioida energiankulutusta vähennettäessä myös ohjelmistojen osuus, koska ohjelmistot ovat se osa arkkitehtuurista, joka käyttää laitteiston tarjoamia resursseja ja asettaa tarpeen energialle. Energiatehokkaiden ohjelmistojen kehittäminen ja mallintaminen on tärkeää, koska ohjelmistot kuluttavat energiaa tehtävien suorittamiseen. Kehitettävien ohjelmistojen tulee myös kyetä hyödyntämään laitteistotason kehitystä, jotta tekninen edistys laitteissa ei mene hukkaan. (Katal, Dahiya & Choudhur, 2022)

Eniten resursseja kuluttava komponentti riippuu siitä, minkä tyyppinen tehtävä on kyseessä. Prosessori on eniten resursseja kuluttava komponentti paljon laskentatehoa vaativissa tehtävissä ja tiedonsäilytysresurssit dataintensiivisissä tehtävissä. Liitetyt laitteet eli verkkokortit, reitittimet ja kytkimet vaativat paljon energiaa kommunikaatiota vaativissa tehtävissä. (Katal, Dahiya & Choudhury, 2022)

3.2 Järjestelmien energiankulutuksen ratkaisuita

Katal, Dahiya & Choudhury (2022) nimeävät tutkimuksessaan ohjelmistoratkaisuja ja tulevaisuuden lisätutkimusaiheita palvelinkeskusten energiankulutuksen vähentämiseksi. Nimetyt ratkaisut ongelmaan ovat virtuaalikoneiden siirtäminen, kuorman tasapainottaminen ja työkuorman kategorisoiminen. Lisäksi tutkimuksessa tarkastellaan uutta nousevaa teknologiaa, kontteja (containerization). Kontit ovat vaihtoehto virtualisoimiselle. Virtualisoinnissa isäntäkoneella käytetään useita virtuaalikoneita, jotka käyttävät omaa käyttöjärjestelmäänsä, jossa käytetään sovellusta. Kontteja käytettäessä isäntäkoneessa ajetaan konttimoottoria, jonka sisällä sovellukset toimivat erillisissä konteissa (container). Kontti ei vaadi useita erillisiä virtuaalikoneita ja käyttöjärjestelmiä ja on siksi helpompi siirtää, kuin perinteinen virtuaalikone. (Katal, Dahiya & Choudhury, 2022, Zhang, Liu, Fu & Yahyapour, 2018) Huomattavan paljon kirjallisuudessa esiintyviä menetelmiä palvelinkeskusten energiankulutuksen hallintaan ovat virtuaalikoneiden sijoittaminen ja siirtäminen ja työkuorman tasapainottaminen. Katal, Dahiya ja Choudhury (2022) nimeävät työkuorman kategorisoimisen erillisenä

menetelmänä, mutta useissa tutkimuksissa menetelmä käsitellään osana työkuorman tasapainotusalgoritmia, jos algoritmi käyttää menetelmässään töiden kategorisoimista.

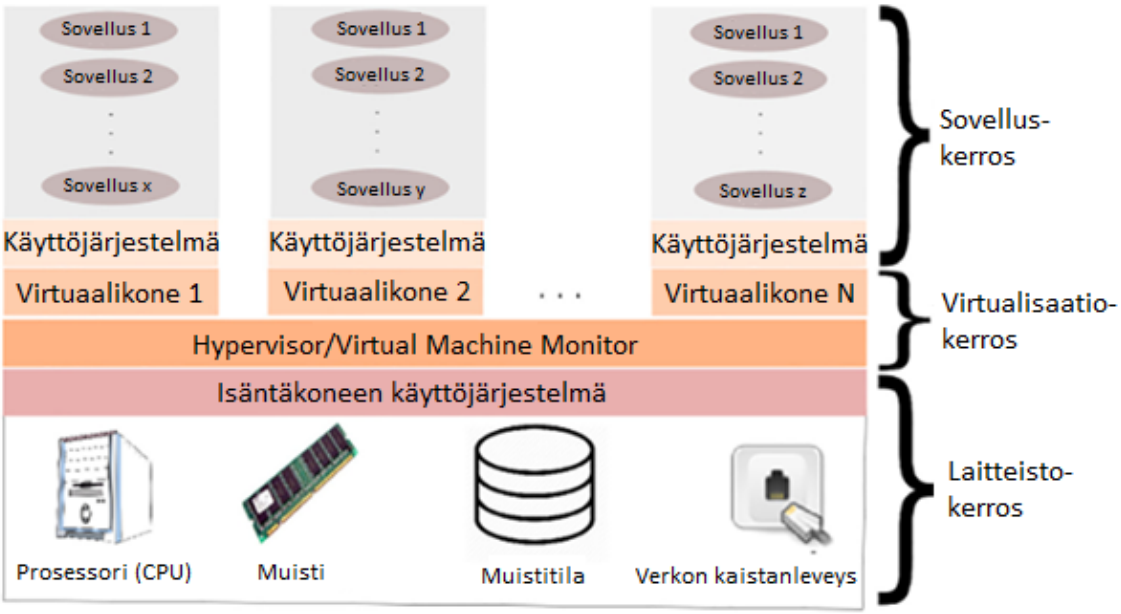
Virtuaalikoneiden siirtämisellä voidaan saavuttaa säästöä energian kulutuksessa. Aiheen tutkimuksessa on kehitetty useita eri algoritmeja virtuaalikoneiden tehokkaalle sijoittamiselle. Algoritmit pyrkivät optimoimaan energian käyttöä, palveluiden laatua ja resurssien käyttöä. Käytännössä virtuaalikoneet pyritään sijoittamaan mahdollisimman pieneen määrään fyysisiä koneita ja suoritettaviin tehtäviin mahdollisimman hyvin soveltuviin fyysisiin koneisiin, resurssien säästämiseksi. Myös koneiden siirtäminen vaatii resursseja ja täytyy tehdä mahdollisimman tehokkaasti. (Katal, Dahiya & Choudhury, 2022)

Virtuaalikoneiden siirtämisalgoritmin laatua voidaan arvioida erilaisilla mittareilla. Arvioinnissa on käytetty mittareina siirtämiseen kuluvaa aikaa, palvelun häiriöaikaa, siirretyn datan määrää ja palvelun laadun laskemista. Virtuaalikoneen siirtäminen voi aiheuttaa häiriötä toimintoon osallistuville komponenteille ja siirtämismenetelmän pitää pystyä sietämään ja minimoimaan häiriöt. Aiheutuvia häiriöitä ovat laskentatehon kulutus, verkon kaistanleveyden kulutus ja tallennustilan kulutus. Virtuaalikoneen siirtäminen kuluttaa hetkellisesti isäntäkoneen ja kohdekoneen resursseja aiheuttaen mahdollisesti häiriötä muiden virtuaalikoneiden toimintaan. Virtuaalikoneen siirtäminen kuluttaa palvelinkeskuksen verkon kaistanleveyttä, aiheuttaen mahdollisesti häiriöitä verkon ruuhkautuessa. (Zhang, Liu, Fu & Yahyapour, 2018)

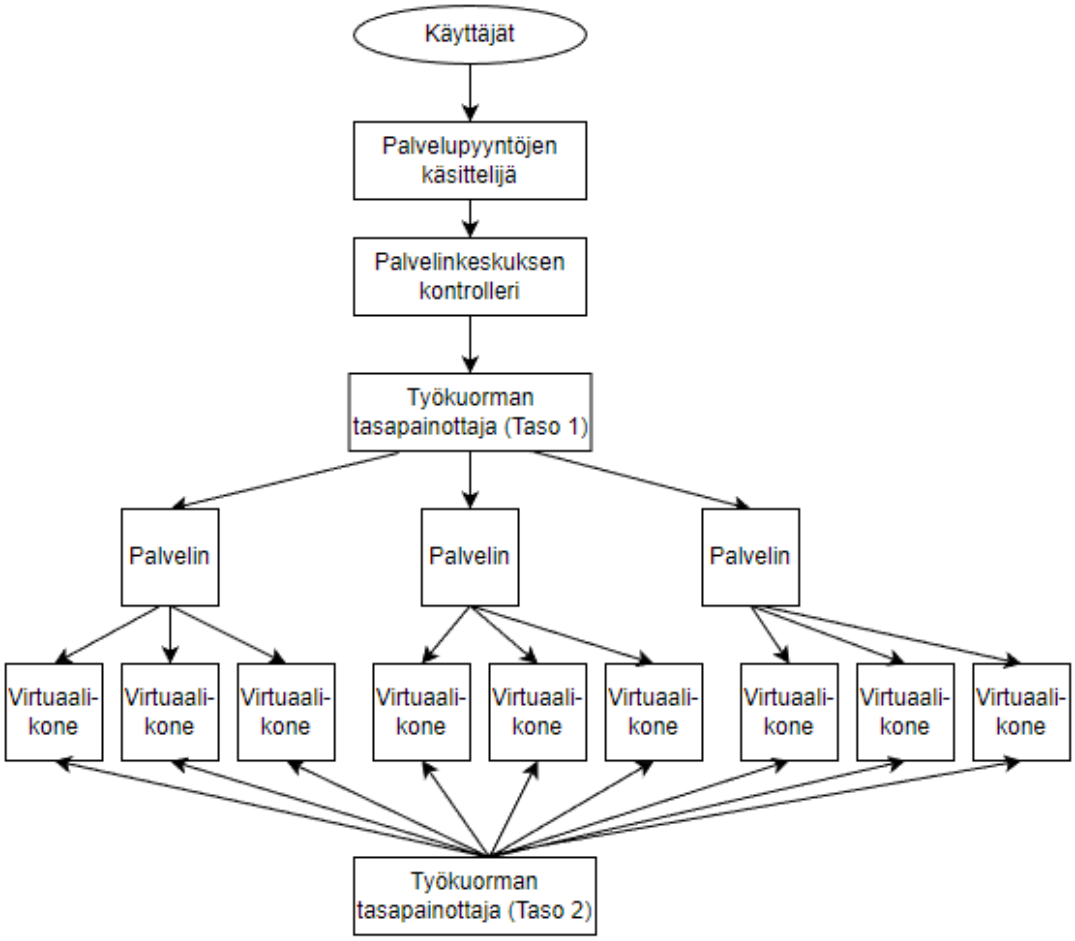
Työkuorman tasapainottaminen on yksi pilvipalveluiden merkityksellisimmistä haasteista (Ghomi, Rahmani & Qader, 2017) ja yksi vaikeimmista ja arvokkaimmista tutkimusaiheista palvelinkeskusten virtuaalikoneisiin liittyen (Katal, Dahiya & Choudhury, 2022). Työkuorman tasapainottamisella tarkoitetaan kuorman, eli palvelinkeskukseen saapuvien käyttäjien palvelupyyntöjen, jakamista ja uudelleenjakamista virtuaalikoneiden välillä tavoitteena maksimoida suoritusteho ja resurssienkäyttö, minimoida vasteaika ja parantaa sovellusten suorituskykyä (Ghomi, Rahmani & Qader, 2017). Työkuorman tasapainottaminen voi tapahtua joko palvelimien tai virtuaalikoneiden välillä, kuten esitetty kuviossa 2 (Afzal & Kavitha, 2019). Työkuorman tasapainottamisella pyritään välttymään ylikuormitetuilta ja alikuormitetuilta palvelimilta (Kumar & Kumar, 2019). Tasapainottamisessa toimii apuna virtuaalikoneiden siirtomenetelmät. Tutkimukset ovat osoittaneet, että energiatehokkuus on parhaimmillaan koneen resurssien käytön olleessa maksimissa. Tasapainotusalgoritmilla siis maksimoidaan resurssien käyttö, kuitenkin välttäen palveluiden laadun kärsiminen ja palvelutasosopimusten rikkominen työkuorman vaihdellessa (Ghomi, Rahmani & Qader, 2017). Tasapainottajaohjelmisto käsittelee käyttäjiltä tulevat pyynnöt ennen niiden saapumista palvelimelle ja ohjaa tehtävät oikeille virtuaalikoneille työkuorman vaatimusten ja virtuaalikoneiden vapaiden resurssien perusteella (Katal, Dahiya & Choudhury, 2022). Lisäksi saapuvat palvelupyynnöt asetetaan jonoon ja ajoitetaan jollakin tehtävään kehitetyistä useista mahdollisista menetelmistä (Dey & Gunasekhar, 2019).

Työkuorman tasapainottamiseen on kehitetty kirjallisuudessa useita eri algoritmeja ja menetelmiä (Katal, Dahiya & Choudhury, 2022). Algoritmin päätavoitteet ovat kulujen pienentäminen, skaalautuvuus ja tärkeiden tehtävien priorisoiminen. Algoritmin toimivuutta mitataan onnistuneiden tehtävien määrällä, prosessien kustannuksilla, virheidensietokyvyllä, tehtävien siirtoajalla, vasteajalla, resurssien käyttöasteella, skaalautuvuudella ja järjestelmän suorituskyvyllä. (Ivanisenko & Radivilova, 2015)

Työkuorman kategorisoiminen on tärkeä osa useita työn tasapainotusalgoritmeja ja työkuorman ennakointimenetelmiä. Kategorisoimisella tarkoitetaan tehtävän luokittelua sen vaatimien resurssien mukaan ennen tehtävän siirtämistä virtuaalikoneelle. Töiden luokittelulla voidaan pienentää todennäköisyyttä palvelinten ylikuormittumiselle ja välttää virtuaalikoneiden tarpeetonta siirtämistä. Kategorisoinnilla voidaan suunnitella palvelinkeskuksen resurssien tarvetta etukäteen ja siten parantaa palveluiden laatua ja resurssien käyttöastetta (Katal, Dahiya & Choudhury, 2022).



KUVIO 1 Yksittäisen palvelimen arkkitehtuuri (Mishra, Sahoo & Parida, 2020)



KUVIO 2 Käyttäjien palvelupyyntöjen saapuminen työkuorman tasapainottajalle ja tasapainottajan sijainti palvelinkeskusarkkitehtuurissa (Afzal & Kavitha, 2019)

4 Palvelinkeskusten energiankulutuksen vähentäminen ohjelmistoratkaisuilla

Luvussa käsitellään syvällisesti aikaisemmin esiteltyjä ohjelmistoratkaisuita palvelinkeskusten energiankulutuksen vähentämiseen. Luku on jaettu kahteen osaan, jotka ovat kirjallisuudessa eniten esiintyneet menetelmät energiatehokkuuden parantamiseen eli virtuaalikoneiden sijoittaminen ja työkuorman tasapainottaminen. Kumpikin ratkaisu käsitellään ensin yleisellä tasolla, minkä jälkeen tarkastellaan esimerkkejä tutkimuksissa esitetyistä menetelmistä ja menetelmillä saavutettuja tuloksia.

4.1 Virtuaalikoneiden sijoittaminen ja siirtäminen

Energiankulutus on suuri menoerä palvelinkeskuksille. Energiankulutuksen vähentämiseksi on tärkeää hyödyntää palvelinten resurssit mahdollisimman tehokkaasti. Ajamalla useita virtuaalikoneita fyysisen koneen, eli isäntäkoneen, sisällä, voidaan hyödyntää fyysisen koneen resursseja tehokkaammin, kuin käyttämällä palveluita suoraan fyysisen koneen käyttöjärjestelmässä. (Zhao ym., 2018) Resurssien jakamista virtuaalikoneiksi kutsutaan virtualisaatioksi.

Virtualisaatiosta vastaa ohjelmistokerros, joka toimii isäntäkoneen käyttöjärjestelmän ja laitteiston välillä. Ohjelmistosta käytetään nimitystä *virtual machine monitor* tai *hypervisor*. Hypervisor hallitsee alustalla toimivia virtuaalikoneita (Choudhary ym., 2017). Hypervisor tarkkailee virtuaalikoneiden vaatimuksia fyysisen koneen resursseista ja pitää huolen, että yksittäisen virtuaalikoneen häiriö ei aiheuta ongelmia muille virtuaalikoneille (Imran ym., 2022).

Isäntäkoneen resurssien riittämättömyys saattaa aiheuttaa vaikeuksia. Virtuaalikoneet jakavat isäntäkoneen resurssit, jolloin resurssien käyttö voi olla epätasapainoista. Epätasapainon takia palveluiden laatu ja palvelinkeskuksen tehokkuus voivat kärsiä. Jos esimerkiksi kaksi samassa isäntäkoneessa toimivaa virtuaalikonetta vaatii paljon prosessoritehoa, prosessoritehosta on käytössä suuri osa, mutta muista resursseista, kuten muistista on käytössä vain vähän. Fyysisten koneiden resurssien käytön maksimoimiseksi on tärkeää sijoittaa uudet virtuaalikoneet isäntäkoneisiin mahdollisimman tehokkaasti ja tarvittaessa siirtää virtuaalikoneita fyysisten koneiden välillä. (Zhao ym., 2018) Virtuaalikoneiden energiatehokkaaseen sijoittamiseen ja keskittämiseen on kehitetty useita algoritmeja.

Virtuaalikoneiden liiallinen keskittäminen voi kuitenkin johtaa palveluiden laadun heikentymiseen. Laadun kärsiessä palveluntarjoaja ei välttämättä pysty täyttämään palvelutasosopimuksen ehtoja ja joutuu maksamaan sakkoja, joten keskittämisestä vastaavan algoritmin tulee ottaa energiankulutuksen lisäksi palveluiden laatu huomioon. (Khoshkholghi ym., 2017)

Virtuaalikone voidaan siirtää uuteen isäntäkoneeseen livesiirtona tai nonlivesiirtona. Livesiirrossa virtuaalikoneen tarjoama palvelu ei katkea siirron aikana, kun taas nonlivesiirron aikana palvelu saa katketa. (Zhang, Liu, Fu & Yahapour, 2018) Virtuaalikoneiden livesiirto on paljon tutkittu aihe ja tekniikan on havaittu olevan tehokas palvelinkeskusten energiankulutuksen hallinnassa (Kansal & Chana, 2016). Energiankulutuksen vähentämisen lisäksi tavoitteena voi olla siirtää keskenään paljon kommunikoivia virtuaalikoneita samaan isäntäkoneeseen verkkoliikenteen vähentämiseksi (Zhang, Liu, Fu & Yahapour, 2018). Toisaalta myös virtuaalikoneen siirtäminen kuluttaa resursseja, kuten energiaa, verkon kaistanleveyttä ja laskentatehoa, joten on tärkeää minimoida siirtämiseen kuluva aika ja resurssien käyttö (Imran ym., 2022). Livesiirtäminen on viimeaikaisessa kirjallisuudessa huomattavasti enemmän käsitelty aihe kuin nonlivesiirtäminen, koska se mahdollistaa palveluiden paremman jatkuvuuden, mikä on tärkeää pilvipalveluiden tarjonnassa.

Zhangin ym. (2018) tutkimuksen mukaan virtuaalikoneita siirtämällä voidaan saavuttaa seuraavia etuja. Palvelinkeskuksen laitteistoa voidaan huoltaa ilman palvelukatkoja, kun palveluita tarjoavat virtuaalikoneet siirretään huollettavasta laitteesta toiseen. Palvelinten kuormaa voidaan tasapainottaa siirtämällä palveluita ylikuormittuneilta palvelimilta muihin palvelimiin, joka parantaa palveluiden laatua. Palveluita keskitetään ja siirretään pois alikuormitetuilta palvelimilta, jotta käynnissä olevien palvelimien määrä voidaan minimoida energian säästämiseksi. Palveluita voidaan siirtää palvelinkeskusten välillä tavoitteena joko saada palvelut lähelle käyttäjiä viiveen pienentämiseksi, tai saada palvelut palvelinkeskukseen, joka sijaitsee viileämmässä ympäristössä, jolloin säästetään palvelinkeskuksen jäähdytykseen kuluvassa energiassa. Virtuaalikoneiden siirtäminen mahdollistaa yhteistyön palveluntarjoajien kanssa. Yhteistyössä toimien palveluiden laskentaympäristö voidaan siirtää lähelle sovellusten tarvitsemaa dataa, jotta voidaan välttää suurten datamäärien siirtäminen. Tämän tutkimuksen kannalta oleellisin etu, jota virtuaalikoneiden siirtämisellä saavutetaan, on palveluiden keskittäminen pieneen määrään fyysisiä koneita.

Keskittämällä virtuaalikoneet palvelinkeskuksen työkuorman mukaan mahdollisimman pieneen määrään fyysisiä koneita voidaan saavuttaa suuria säästöjä palvelinkeskuksen energiankulutuksessa, koska käynnissä olevien palvelimien määrä minimoidaan. Tutkimusten mukaan palvelinkeskukset käyttävät keskimäärin 30 % resursseistaan ja tyhjäkäynnillä oleva palvelin kuluttaa 70 % energiasta, jonka se kuluttaisi ollessaan maksimikäytöllä (Zhang ym., 2019), joten energiankäytön minimoimiseksi olisi tärkeää sammuttaa tyhjäkäyvät koneet tai siirtää niihin virtuaalikoneita ylikuormitetuista palvelimista hyödyntämään käyttämättömiä resursseja. Virtuaalikoneiden livesiirtäminen pois alhaisella käytöllä olevista isäntäkoneista ja koneiden sammuttaminen on hyvä tapa parantaa palvelinkeskuksen energiatehokkuutta. (Choudhary ym. 2017)

Energiatehokkuuden ja palveluiden laadun näkökulmasta virtuaalikoneiden sijoittamisessa täytyy ottaa huomioon fyysisten koneiden minimoimisen lisäksi toistensa kanssa kommunikoivien koneiden sijainti. Jos keskenään

kommunikoivat virtuaalikoneet sijaitsevat verkossa kaukana toisistaan verkon kytkimien energian- ja kaistanleveydenkulutus kasvavat (Masdari & Zangakani, 2019).

Virtuaalikoneen livesiirtämiseen kuuluu kolmen eri komponentin siirtäminen (Masdari & Zangakani, 2019). Jotta virtuaalikone voi jatkaa toimintaansa samasta pisteestä siirron jälkeen, on siirrettävä tieto koneen siirronaikaisesta tilasta uuteen isäntäkoneeseen. Tilatietojen siirtämisestä käytetään nimitystä ”memory data migration”. Jos sovellusten vaatimat tietovarastot eivät ole saavutettavissa uudesta isäntäkoneesta, myös tietovarastot on siirrettävä (storage data migration). Lisäksi virtuaalikoneen siirtäminen vaatii strategian, jonka avulla käyttäjät pääsevät käsiksi sen tarjoamiin palveluihin, joten verkkoyhteys täytyy siirtää uuteen isäntäkoneeseen (network connection continuity) ja livesiirron tapauksessa yhteys täytyy pitää avoimena virtuaalikoneen siirtämisen aikana. (Zhang ym., 2018) Eniten aikaa ja resursseja kuluttava vaihe virtuaalikoneen siirtämisessä on tilatietojen eli muistin (memory pages) siirtäminen (Noshy, Ibrahim & Ali, 2018).

Virtuaalikoneiden livesiirtämiset voidaan jakaa kolmeen kategoriaan tilatietojen siirtämisjärjestyksen perusteella. Kategoriat ovat ennakkokopioiminen (pre-copy), jälkikopioiminen (post-copy) ja hybridikopiointi (Masdari & Zangakani, 2019). Yleisin tapa muistin siirtämiseen on ennakkokopiointi (Noshy, Ibrahim & Ali, 2018). Ennakkokopioinnissa tilatietoja ja muistia kopioidaan toistuvasti alkuperäisestä isäntäkoneesta uuteen isäntäkoneeseen. Alkuperäinen virtuaalikone jatkaa kopioinnin aikana toimintaansa ja voi muuttaa jo siirrettyjä tietoja (Masdari & Zangakani, 2019). Ensimmäisessä iteraatiossa kopioidaan alkuperäisen virtuaalikoneen koko muisti uuteen koneeseen. Tämän jälkeen kopioidaan toistuvasti ”likaantunutta” muistia, eli tietoja, jotka ovat muuttuneet edellisen lähetyksen jälkeen, kohdekoneeseen. Kun jäljellä olevan datan määrässä saavutetaan ennalta määritelty ehto tai iteraatioita on suoritettu tietty määrä, loput muistista ja prosessorin tilatiedot siirretään kohdekoneeseen, alkuperäinen virtuaalikone sammutetaan ja kone aloittaa toimintansa uudessa isäntäkoneessa (Zhang ym., 2018, Noshy, Ibrahim & Ali, 2018). Ennakkokopiointi on kannattavaa, jos virtuaalikoneessa käytettävä sovellus lukee paljon dataa (Masdari & Zangakani, 2019). Ennakkokopiointimenetelmällä pyritään pienentämään palveluiden häiriöaikaa (downtime), siirtämällä vain vähän dataa palvelun ollessa pois käytöstä. Menetelmässä, joudutaan kuitenkin siirtämään toisteisesti samoja muistisivuja, mikä kuluttaa verkon kaistanleveyttä ja pidentää kokonaissiirtoaikaa. Ennakkokopiointimenetelmä on myös luotettava, jos siirto joudutaan perumaan häiriön vuoksi, koska tilatiedot ovat tallessa alkuperäisessä virtuaalikoneessa siirron päättymiseen asti (Noshy, Ibrahim & Ali, 2018).

Jos sovellus kirjoittaa dataa ja kirjoitusnopeus on suurempi kuin verkon yli voidaan siirtää, on kannattavampaa siirtää virtuaalikone jälkikopioinnilla (Masdari & Zangakani, 2019). Jälkikopiointimenetelmässä pysäytetään ensin alkuperäisen virtuaalikoneen toiminta ja prosessorin tilatiedot siirretään uuteen isäntäkoneeseen. Kun virtuaalikone on käynnistynyt uudessa isäntäkoneessa, siirretään muisti alkuperäisestä koneesta uuteen (Zhang ym., 2018). Jos kone tarvitsee muistia, jota ei ole vielä lähetetty, syntyy virhe ja muistisivu noudetaan

alkuperäisestä koneesta. Jälkikopiointimenetelmä pienentää siirtämiseen kuluvaa kokonaisaikaa ja verkon kaistanleveyden kulutusta välttämällä muistin toistuvan siirtämisen. Menetelmässä palvelun häiriöaika saattaa kasvaa ja palveluiden laatu kärsiä, jos ajon aikana joudutaan odottamaan tarpeellisten muistisivujen noutamista. Jälkikopiointimenetelmä ei ole yhtä luotettava, kuin ennakkokopiointimenetelmä, koska alkuperäisen virtuaalikoneen virhetilanteen sattuessa uusi virtuaalikone ei saa tuoreimpia tietoja koneen tilasta, eikä voi jatkaa toimintaansa. (Noshy, Ibrahim & Ali, 2018)

Hybridikopiointi on yhdistelmä ennakkokopiointi- ja jälkikopiointimenetelmistä. Hybridikopioinnissa siirretään aluksi rajoitettu määrä muistia uuteen isäntäkoneeseen, jonka jälkeen uuden virtuaalikoneen toiminta käynnistyy ja loput muistista siirretään jälkikopiointimenetelmää käyttäen (Zhang ym., 2018). Hybridimenetelmässä siirretään ensin eniten käytetyt muistisivut kohdekoneeseen ennakkokopiointimenetelmää käyttäen, alkuperäisen virtuaalikoneen toiminta keskeytetään ja prosessorin tilatiedot siirretään kohdekoneen käynnistämistä varten. Loput muistisivut siirretään jälkikopiointimenetelmää käyttäen. Hybridikopiointi kärsii samoista ongelmista luotettavuutensa kanssa, kuin jälkikopiointimenetelmä. (Noshy, Ibrahim & Ali, 2018)

Virtuaalikoneiden sijoittamiseen ja siirtämiseen mahdollisimman tehokkaasti on kehitetty useita tekniikoita ja algoritmeja sopivan isäntäkoneen ja siirtämistä vaativien virtuaalikoneiden tunnistamiseksi. Seuraavaksi käsitellään lyhyesti kolmea viimeaikaisessa kirjallisuudessa esiintyvää menetelmää virtuaalikoneiden sijoittamiseksi.

Tutkimuksessaan Zhang ym. (2019) esittelevät kehittämänsä menetelmän, joka simuloi virtuaalikoneiden sijoittamista ja etsii iteratiivisesti parhaan sijoituksen virtuaalikoneille varauspohjaisessa palvelinkeskuksessa. Algoritmi jakaa palvelimet sattumanvaraisiin pareihin, joiden välillä virtuaalikoneiden siirtämistä simuloidaan parhaan paikallisen ratkaisun löytämiseksi. Seuraavissa iteraatioissa etsitään paras globaali ratkaisu vaihtamalla palvelinpareja ja toistamalla prosessia. Globaalia ratkaisua käytetään virtuaalikoneiden varsinaista sijoittamista varten. Tavoitteena on maksimoida virtuaalikoneiden samanaikainen ajaminen kussakin fyysisessä koneessa ja samalla maksimoida fyysisten koneiden lepotila-aika. Testeissä pystyttiin pienentämään palvelimien energiankulutusta 26,8 %. (Zhang ym., 2019) Vertailukohtana testeissä käytettiin MBFD-algoritmia (Beloglazov, Abawajy & Buyya, 2012).

Khoshkholghin ym. (2017) virtuaalikoneiden sijoittamismenetelmään kuuluu neljä algoritmia, jotka toimivat yhdessä virtuaalikoneita keskitettäessä. Yksi algoritmeista mittaa ajoittain palvelinten prosessorin, muistin ja kaistanleveyden käyttöä. Resurssien käytön ylittäessä määrätyn kynnyksen, palvelin määritellään ylikuormitetuksi ja osa sen virtuaalikoneista sijoitetaan uudelleen. Seuraavassa vaiheessa algoritmi valitsee ylikuormitetun palvelimen virtuaalikoneista siirrettävät koneet. Kolmannessa vaiheessa algoritmi määrittelee siirrettäville virtuaalikoneille uudet isäntäkoneet. Siirrettävät virtuaalikoneet asetetaan järjestykseen niiden muistivaatimusten mukaan ja ne pyritään sijoittamaan uudelleen eniten muistia vaativasta koneesta lähtöisin. Ensisijaisesti virtuaalikoneet yritetään

sijoittaa normaalikuormalla toimiviin palvelimiin, seuraavaksi alikuormitettuihin palvelimiin ja viimeiseksi tarvittaessa käynnistetään uusi palvelin, jos virtuaalikoneen resurssivaatimuksia täyttävää isäntäkonetta ei löydy. Myös alikuormitettujen isäntäkoneiden virtuaalikoneet sijoitetaan uusiin kohteisiin vastaavalla tavalla. Alikuormitetut isäntäkoneet tunnistetaan resurssienkäytön alittaessa asetetun kynnysarvon. Testeissä menetelmällä pienennettiin palvelinten energiankulutusta parhaimmillaan 28 % vertailualgoritmeihin nähden. (Khoshkholghin ym., 2017)

Farahnakian ym. (2019) kehittämä algoritmi sijoittaa virtuaalikoneet palvelimille nykyisen resurssien käytön lisäksi ennakoimalla lähitulevaisuuden resurssien käyttöä. Algoritmi luo siirtosuunnitelman kahdessa vaiheessa. Ensimmäisessä vaiheessa tarkastellaan ylikuormittuneita ja ennakoituja ylikuormittuneita palvelimia. Siirrettäväksi valitaan virtuaalikoneet, joiden siirtämiseen kuluu vähiten aikaa. Siirrettäviä virtuaalikoneita valitaan riittävän monta, että isäntäkoneen kuormitus ja ennakoitu kuormitus saadaan normaalille tasolle.

Seuraavaksi siirrettäville koneille valitaan uudet isäntäkoneet siten, että kohdekoneiden kuorma ei kasva liian suureksi. Toisessa vaiheessa järjestelmän aktiiviset fyysiset koneet järjestetään niiden kuormituksen mukaan laskevaan järjestykseen. Algoritmi pyrkii siirtämään vähiten kuormitetusta fyysisestä koneesta alkaen sen sisältämiä virtuaalikoneita korkeasti kuormitettuihin fyysisiin koneisiin. Lähdekoneen virtuaalikoneet järjestetään niiden kuorman mukaan ja koneet pyritään sijoittamaan uudelleen alkaen korkeimman kuorman virtuaalikoneesta. Virtuaalikone pyritään sijoittamaan eniten kuormitettuun fyysiseen koneeseen, johon se on mahdollista sijoittaa ilman ylikuormittumista. Jos lähdekoneen kaikkia virtuaalikoneita ei pystytä siirtämään, mitään koneista ei siirretä. Jos kaikki virtuaalikoneet voidaan sijoittaa uudelleen, ne siirretään ja isäntäkone voidaan asettaa lepotilaan. Menetelmällä saavutettiin testeissä 71,6 % energiankulutuksen pienentäminen verrattuna heikoimpaan verrokkialgoritmiin. (Farahnakian ym., 2019)

4.2 Työkuorman tasapainottaminen

Käyttäjien palvelupyynnöt saapuvat palvelinkeskukselle palvelupyyntöjen välittäjän kautta. Välittäjä analysoi resurssien saatavuutta ja välittää pyynnön valitsemalleen palvelinkeskukselle, jossa palvelinkeskuksen kontrolleri vastaanottaa pyynnön jatkokäsittelyä varten. Pyyntö välitetään edelleen fyysiselle palvelimelle ja palvelimen työkuorman tasapainottaja (load balancer) ohjaa palvelupyynnön sopivalle virtuaalikoneelle. Prosessin aikana tasapainottaja päivittää virtuaalikoneiden tilakaaviota, josta se tunnistaa käytettävissä olevat virtuaalikoneet. (Kumar & Kumar, 2019)

Työkuorman tasapainottaja on ohjelmisto, joka käsittelee palvelimeen tulevat käyttäjien pyynnöt ja jakaa työt palvelimille tai palvelimella toimiville virtuaalikoneille (Afzal & Kavitha, 2019). Työkuorman tasapainottamisen tavoitteena

on, että mikään palvelin ei ole ylikuormittunut, alikuormittunut tai toimeton (Katal, Dahiya & Choudhury, 2022). Hyvä tasapainottaja parantaa järjestelmän resurssien käyttöä, palveluiden saavutettavuutta ja tehtävien suoritusnopeutta. Tasapainottaminen takaa jatkuvan palveluntarjoamisen komponenttien virhetilanteissa. Tasapainottamisella pyritään lisäksi mahdollistamaan tulevaisuudessa todennäköisesti kasvavien sovellusten skaalautuvuus ja priorisoimaan kiireellisiä tehtäviä. (Kumar & Kumar, 2019) Tasapainottamisen tutkimus on arvokasta, koska ohjelmistot tarvitsevat jatkuvasti parempia menetelmiä työkuorman tasaiseen jakamiseen ja sen kautta palvelinkeskuksen tehokkuuden parantamiseen (Katal, Dahiya & Choudhury, 2022). Työkuorman tasapainottaja valitsee tehtävään sopivan virtuaalikoneen tasapainotusalgoritmin perusteella (Ala'anzy & Othman, 2019).

Kumar & Kumar (2019) ovat tunnistaneet työkuorman tasapainottamiseen liittyviä haasteita. Palvelinkeskukset voivat sijaita maantieteellisesti kaukana toisistaan aiheuttaen viivettä niiden välisessä kommunikaatiossa, joten tasapainotusalgoritmin tulee ottaa huomioon toisistaan kaukana sijaitsevat verkon solmukohdat. Tasapainotuksen tulee tapahtua hajautetusti, jotta yhden laitteen kaatuminen ei voi kaataa koko järjestelmää. Fyysisten koneiden ylikuormittuessa tasapainottajan täytyy pystyä siirtämään virtuaalikoneita. Tasapainotustekniikoiden täytyy toimia erilaisissa laitteissa. Tasapainottajan täytyy pystyä sopeutumaan nopeisiin työmäärän muutoksiin. Tasapainotusalgoritmin tulee olla yksinkertainen, koska algoritmin monimutkaisuus heikentää järjestelmän suorituskykyä ja tehokkuutta. (Kumar & Kumar, 2019)

Ghomi, Rahmani & Qader (2017) luokittelevat työkuorman tasapainottamisstrategiat joko järjestelmän tilan mukaan tai prosessin aloittajan mukaan. Prosessin aloittajan mukaan strategiat voidaan luokitella lähettäjäaloitteisiin, vastaanottaja-aloitteisiin ja symmetrisiin strategioihin. Lähettäjäaloitteisessa tasapainottamisessa ylikuormitettu verkkosolmu hakee vähemmän kuormitettuja solmuja jakaakseen työkuormaa tehtävien saapuessa. Vastaanottaja-aloitteisessa strategiassa sen sijaan alikuormitetut solmut hakevat korkeasti kuormitettuja solmuja tehtävien valmistuessa. Symmetrinen strategia on yhdistelmä edellisistä.

Järjestelmän tilan mukaan työkuorman tasapainottamisstrategiat voidaan jakaa staattisiin ja dynaamisiin menetelmiin. Staattiset menetelmät noudattavat ennalta määrättyjä sääntöjä eivätkä huomioi järjestelmän kunkin hetkistä tilaa päätöksenteossa. Järjestelmän tilatietojen sijaan staattinen järjestelmä tekee päätöksensä järjestelmän resurssien perusteella (Mishra, Sahoo & Parida, 2020). Staattisen algoritmin perusteella toimivien tasapainottajien etuna on niiden yksinkertaisuus (Afzal & Kavitha, 2019), mutta ne eivät sovellu järjestelmiin, joiden tila vaihtelee paljon tai jotka suorittavat hajautettua laskentaa (Kumar & Kumar, 2019).

Staattiset menetelmät voidaan jakaa edelleen optimaalisiin ja sub-optimaalisiin strategioihin. Työkuorman tasapainottaja pyrkii ensin tekemään päätöksen tehtävän sijoittamisesta sen vaatimusten ja järjestelmän resurssien perusteella optimaalisesti. Jos optimaalinen sijoittaminen ei onnistu määrätyssä ajassa se sijoitetaan järjestelmään sub-optimaalisesti. Sub-optimaalinen sijoittaminen

voidaan tehdä arviointimenetelmää tai heuristista menetelmää käyttäen. Arviointimenetelmää käyttävässä tekniikassa tasapainottaja-algoritmi päättyy löydettyään hyväksi arvioimansa ratkaisun. Algoritmi ei siis tutki koko järjestelmää täydellisen sijainnin löytämiseksi tehtävälle. Lopuksi tehdyn ratkaisun laatu arvioidaan. Heuristisessa menetelmässä tasapainottaja-algoritmi tekee oletuksia jaettavista tehtävistä ja saatavilla olevista resursseista ja tekee päätöksen tehtävien sijoittamisesta. Algoritmi ei kuitenkaan ole rajoittunut määriteltyihin oletuksiin vaan pystyy sopeutumaan tilan muutoksiin. (Ghomi, Rahmani & Qader, 2017)

Staattinen tasapainotusmenetelmä ei salli resurssien ajonaikaista jakoa tai liitettyjen palvelinten tunnistamista tehtävien ajon aikana (Kumar & Kumar, 2019) ja kärsii tästä syystä tehtävien ruuhkautumisesta (Afzal & Kavitha, 2019). Koska ajonaikainen tehtävien ja virtuaalikoneiden siirto ei ole mahdollista, staattisia menetelmiä käyttävät työkuorman tasapainottajat soveltuvat vain pieniin järjestelmiin, joilla on vähän resursseja ja verkkopalvelimille, joiden käsittelemät pyynnöt ovat samankaltaisia keskenään ja työt hyvin ennustettavissa (Kumar & Kumar, 2019). Staattiset menetelmät eivät sovellu pilviympäristöön, jossa järjestelmän tila muuttuu jatkuvasti työkuorman muutosten ja töiden erilaisuuden vuoksi (Afzal & Kavitha, 2019).

Dynaamiset työkuorman tasapainottamismenetelmät ottavat päätöksenteossa huomioon järjestelmän päätöksenaikaisen tilan. Dynaaminen tasapainottaja tarkkailee järjestelmän työkuormaa tietyin aikavälein ja tarvittaessa jakaa kuormaa uudelleen (Kanakala & Reddy, 2015). Dynaaminen työkuorman tasapainotusalgoritmi pystyy tarvittaessa siirtämään virtuaalikoneita ja tehtäviä koneiden välillä, mutta on monimutkainen toteuttaa verrattuna staattiseen algoritmiin (Afzal & Kavitha, 2019). Suurin osa dynaamisista algoritmeista toimii nelivaiheisesti. Ensimmäisessä vaiheessa tarkkaillaan järjestelmän kuormaa ja resurssien tilaa. Seuraavaksi verkon solmut vaihtavat tietoa kuormasta ja resursseista. Kolmannessa vaiheessa tehdään laskelmia tietojen perusteella ja päätetään työkuorman uudelleenjakamisesta. Viimeisessä vaiheessa suoritetaan datan varsinainen siirto. (Ghomi, Rahmani & Qader, 2017)

Dynaamiset menetelmät jaetaan edelleen kahteen luokkaan, hajautettuihin ja hajauttamattomiin algoritmeihin (Ghomi, Rahmani & Qader, 2017). Hajautetussa algoritmissa kaikki verkon solmut kommunikoivat toistensa kanssa ja osallistuvat työkuorman tasapainottamiseen. Solmujen välinen kommunikaatio voi tapahtua yhteistyöhaluisesti (cooperative) tai yhteistyöstä kieltäytyen (non-cooperative). Yhteistyöhaluisesti kommunikoivat solmut pyrkivät saavuttamaan yhteisen tavoitteen, kuten pienentää kaikkien järjestelmän tehtävien vasteaikaa. Yhteistyöstä kieltäytyvät solmut pyrkivät itsenäisesti saavuttamaan oman paikallisen tavoitteensa, kuten pienentämään tietyn paikallisen tehtävän vasteaikaa. (Ghomi, Rahmani & Qader, 2017)

Hajauttamattomissa algoritmeissa yksi tai pieni määrä verkkosolmuja suorittaa työkuorman tasapainottamisen. Jos yksi solmu suorittaa algoritmia, puhutaan keskitetystä tasapainottamisesta ja jos vain osa solmuista suorittaa algoritmia puhutaan semi-hajautetusta tasapainottamisesta. Hajauttamattomien algoritmien heikkous on huono virheensietokyky. Jos työkuorman

tasapainottamisesta vastaavaan komponenttiin tulee vikaa, toimii koko järjestelmä heikosti. (Kumar & Kumar, 2019)

Ghomi, Rahmani & Qader (2017) ovat tutkimuksessaan koonneet aikaisemmassa kirjallisuudessa käytettyjä mittareita työntasapainottamisalgoritmin toiminnalle. Käytetyt mittarit ovat suoritusteho eli suoritettujen prosessien määrä aikayksikössä, vasteaika, maksimisuoritusaika tehtävälle, skaalautuvuus, virheidensietokyky, tehtävien ja virtuaalikoneiden siirtoaika, virtuaalikoneiden välinen epätasapaino, tehokkuus, energiankulutus ja hiilidioksidipäästöt.

Shafiq, Jhanuhi, Abdullah & Alzain (2021) ovat kehittäneet tutkimuksessaan algoritmin pilvipalvelinkeskusten työkuorman tasapainottamiseen. Lähtökohtana algoritmille on, että kaikille virtuaalikoneille jaetaan tasainen osa isäntäkoneen resursseista. Saapuvat tehtävät jaetaan aluksi sattumanvaraisesti virtuaalikoneille. Tehtävillä on suoritusaika, joka lasketaan tehtävän vaatimista ja virtuaalikoneen omaamista resursseista, ja määräaika, jossa tehtävä tulee olla suoritettu, jotta vältytään palvelutasosopimuksen rikkomiselta. Seuraavaksi tarkistetaan virtuaalikoneen kyky suorittaa tehtävä vähentämällä odotetusta suoritusajasta määräaika. Jos tehtävän suoritus ei onnistu määräajassa, isäntäkoneen resurssit jaetaan uudelleen mahdollisuuksien mukaan. Jos resurssit eivät yhäkään riitä, tehtävä siirretään uuteen virtuaalikoneeseen. Testeissä saavutettiin palvelinkeskussimulaatiossa 78 % resurssien käyttö, joka oli 2 %-yksikköä parempi kuin vertaillun menetelmän.

Zhang & Zhou (2018) esittävät tutkimuksessaan kaksivaiheisen strategian tehtävien aikatauluttamiselle. Ehdotetulla järjestelmällä on dataa järjestelmän aikaisemmasta työkuormasta. Ensimmäisessä vaiheessa järjestelmä määrittelee vanhan datan perusteella tarvittavat virtuaalikonetyypit ja niiden määrän ja luo ne isäntäkoneisiin. Toisessa vaiheessa käyttäjien pyyntöjä saapuu järjestelmään ja ne asetetaan jonoon. Saapuvat pyynnöt luokitellaan ja toimitetaan tehtävään soveltuvalle virtuaalikoneelle. Valmis virtuaalikone-tehtävä-pari siirretään valmiiden jonoon, ja tehtävä suoritetaan. Lopuksi prosessin tiedot tallennetaan järjestelmän historiatietoihin hyödynnettäväksi myöhemmässä aikatauluttamisessa.

Chen, Chen & Kuo (2017), ehdottavat käytettäväksi round robin-pohjaista algoritmia työkuorman tasapainottamiseen. Menetelmässä järjestelmän palvelimille annetaan prioriteettiarvo niiden resurssien ja kuormituksen perusteella 0,1 sekunnin välein. Käyttäjän palvelupyyntöjen saapuessa työkuorman tasapainottaja ottaa arvioitavaksi puolet palvelimista niiden prioriteettiarvon mukaan. Työkuorman tasapainottaja alkaa jakamaan palvelupyyntöjä kaikille virtuaalikoneille. Tasapainottaja käy virtuaalikoneita läpi järjestyksessä ja pudottaa seuraavilta kierroksilta koneet, joiden kuormitus on liian suuri uusille pyynnöille.

Taulukko 1 Energiankulutusta vähentävät ohjelmistoratkaisut

Menetelmä	Kuvaus	Lähde
Virtuaalikoneiden sijoittaminen/siirtäminen		
Livesiirto	Virtuaalikone siirretään uuteen isäntäkoneeseen palvelun katkeamatta.	(Zhang, Liu, Fu & Yahapour, 2018), (Noshy, Ibrahim & Ali, 2018), (Kansal & Chana, 2016)
Ennakkokopionti	Virtuaalikoneen tietoja siirretään toistuvasti uuteen isäntäkoneeseen, kunnes virtuaalikone on valmis aloittamaan toimintansa kohdekoneessa.	(Choudhary, ym., 2017), (Masdari & Zangakani, 2019), (Noshy, Ibrahim & Ali, 2018), (Zhang, Liu, Fu & Yahapour, 2018)
Jälkikopiointi	Virtuaalikoneen toiminta pysäytetään ja sen tilatiedot siirretään kohdekoneeseen. Virtuaalikone käynnistetään kohteessa ja muisti siirretään vasta käynnistyksen jälkeen.	(Choudhary, ym., 2017), (Noshy, Ibrahim & Ali, 2018), (Zhang, Liu, Fu & Yahapour, 2018)
Hybridikopiointi	Virtuaalikoneen eniten käyttämät muistisivut siirretään ennakkokopiointimenetelmää käyttäen. Seuraavaksi virtuaalikone käynnistetään kohdekoneessa ja siirtoa jatketaan jälkikopiointimenetelmää käyttäen.	(Choudhary, ym., 2017), (Noshy, Ibrahim & Ali, 2018), (Zhang, Liu, Fu & Yahapour, 2018)
Nonlivesiirto	Virtuaalikoneen tarjoama palvelu saa katketa siirron aikana.	(Noshy, Ibrahim & Ali, 2018), (Zhang, Liu, Fu & Yahapour, 2018)
Työkuorman tasapainottaminen		
Staattiset menetelmät	Tekevät päätöksiä järjestelmän resurssien perusteella ottamatta	(Afzal & Kavitha, 2019), (Mishra, Sahoo & Parida, 2020)

	huomioon järjestelmän kunkin hetkistä tilaa.	
Optimaalinen	Algoritmi määrittelee optimaalisen ratkaisun kuorman vaatimusten ja järjestelmän resurssien perusteella kohtuullisessa ajassa.	(Ghomi, Rahmani & Qader, 2017), (Kumar & Kumar, 2019)
Sub-optimaalinen	Jos tehtävien optimaalinen sijoittaminen epäonnistuu, tasapainottaja pysähtyy löydettyään hyvän ratkaisun. Algoritmi ei tutki koko järjestelmää ratkaisun löytämiseksi.	(Ghomi, Rahmani & Qader, 2017), (Kumar & Kumar, 2019)
Dynaamiset menetelmät	Ottavat päätöksenteossa huomioon järjestelmän päätöksenaikaisen tilan. Tarkkailevat järjestelmän työkuormaa tietyin aikavälein ja tarvittaessa jakavat kuormaa uudelleen.	(Kanakala & Reddy, 2015), (Mishra, Sahoo & Parida, 2020)
Hajautetut menetelmät	Kaikki verkon solmut kommunikoivat toistensa kanssa ja osallistuvat työkuorman tasapainottamiseen.	(Ghomi, Rahmani & Qader, 2017), (Kumar & Kumar, 2019)
Yhteistyöhaluiset (hajautetut)	Solmut pyrkivät saavuttamaan yhteisen tavoitteen.	(Ghomi, Rahmani & Qader, 2017), (Kumar & Kumar, 2019)
Yhteistyöstä kieltäytyvät (hajautetut)	Solmut pyrkivät itsenäisesti saavuttamaan oman paikallisen tavoitteensa.	(Ghomi, Rahmani & Qader, 2017)
Hajauttamattomat menetelmät	Yksi tai pieni määrä verkkosolmuja suorittaa työkuorman tasapainottamisen.	(Ghomi, Rahmani & Qader, 2017), (Kumar & Kumar, 2019)
Keskitetyt (hajauttamattomat)	Yksi solmu suorittaa työkuorman tasapainottamisen.	(Ghomi, Rahmani & Qader, 2017), (Kumar & Kumar, 2019)

Semi-hajautetut (hajauttamattomat)	Osa solmuista suorittaa työkuorman tasapainottamisen.	(Ghomi, Rahmani & Qader, 2017), (Kumar & Kumar, 2019)

5 Yhteenveto

Tässä tutkielmassa tarkasteltiin kirjallisuuskatsauksen muodossa palvelinkeskuksissa toteutettavia ohjelmistoratkaisuita, joilla voidaan vaikuttaa palvelinkeskuksen energiankulutukseen. Palvelinkeskusten energiankulutus muodostaa merkittävän osan maailman kokonaisenergiankulutuksesta. Palvelinkeskusten energiankulutuksen tutkiminen on tärkeää palveluntarjoajille taloudellisista syistä sekä ympäristöystävällisistä syistä.

Tutkielman tarkoituksena on tunnistaa ohjelmistoja ja ohjelmistopohjaisia ratkaisuita, joilla parannetaan palvelinkeskusten energiatehokkuutta. Tutkielma alkaa palvelinkeskusten ja niiden energiatehokkuuden yleisellä käsittelyllä, jonka jälkeen esitellään merkittävimpiä energiatehokkuutta parantavia menetelmiä. Menetelmien tunnistamisen jälkeen tarkastellaan menetelmien toimintaperiaatetta syvällisesti. Energiatehokkaat menetelmät käsitellään aluksi yleisesti ja tarkastellaan niiden toteuttamiseen liittyviä haasteita ja niiden toimivuutta mittaavia mittareita. Menetelmistä käydään läpi muutamia kirjallisuudessa esiintyviä esimerkkejä.

Tutkielma toteutettiin kirjallisuuskatsauksena hakemalla tutkimusaineistoksi tieteellisiä artikkeleita Google Scholarista ja IEEE Xploresta. Hakusanoina käytettiin mm. data center, energy, cloud, load balancing, virtual machine ja architecture ja näiden eri yhdistelmiä. Lähdeaineistoksi pyrittiin valitsemaan mahdollisimman uusia artikkeleita, siten että ennen 2016 ilmestyneet artikkelit hylättiin lähteistä muutamaa poikkeusta lukuun ottamatta. Lähteiltä vaadittiin tuoreutta, koska aiheena on nopeasti kehittyvä teknologia ja tarkoituksena on varmistaa tiedon olevan mahdollisimman ajankohtaista. Tuoreuden lisäksi valittujen lähdeartikkelien täytyy olla julkaistu lehdessä, joka on julkaisufoorumissa vähintään tasoa 1. Lisäksi lähteiksi pyrittiin valitsemaan artikkeleita, joilla oli mahdollisimman paljon viittauksia. Aiheen melko tarkan rajauksen vuoksi lähdeaineistoon hyväksyttiin myös artikkeleita, joilla on suhteellisen vähän viittauksia, jos artikkeli on uusi ja sopii aiheeseen hyvin.

Kirjallisuudessa eniten esiintyviä ja merkittävimpiä ohjelmistopohjaisia menetelmiä palvelinkeskusten energiankulutuksen pienentämiseen ovat sovelluksia ajavien virtuaalikoneiden sijoittaminen ja siirtäminen ja palvelinkeskuksen työkuorman tasapainottaminen. Palvelinkeskukset hyödyntävät keskimäärin 30 % laitteistonsa resursseista ja tyhjäkäynnillä oleva palvelin kuluttaa 70 % energiasta, jonka se kuluttaisi ollessaan maksimikäytöllä. Ydinajatus energiankulutuksen pienentämisen näkökulmasta työkuorman tasapainottamisessa ja virtuaalikoneiden sijoittamisessa, on lisätä palvelinkeskuksen resurssienkäyttöastetta keskittämällä virtuaalikoneet mahdollisimman pieneen määrään fyysisiä palvelimia ja asettaa käyttämättömät palvelimet lepotilaan. Lyhyesti sanottuna siis maksimoidaan yhtäaikaisesti ajettavien virtuaalikoneiden määrä kussakin käytössä olevassa palvelimessa. Samalla maksimoidaan palvelimien lepoaika ja säästetään energiaa. Tavoitteena on myös suorittaa virtuaalikoneiden siirtäminen mahdollisimman tehokkaasti vähän energiaa kuluttaen. Hyvän työkuorman

tasapainottajan tavoitteena on lisäksi välttää virtuaalikoneiden ja tehtävien tarpeeton siirtäminen ja siten vähentää energiankulutusta siirtämisestä kuormittuvissa verkkolaitteissa.

Tutkimukset osoittavat palvelinten energiankulutuksen riippuvan käytetyistä resurssienkäyttöä optimoivista algoritmeista. Heikkoutena tutkimuksissa on kuitenkin testien toteuttaminen palvelinkeskussimulaatioissa ja vertailukohtana käytetyt menetelmät. Vertailukohtana käytetään useissa artikkeleissa vanhemmista tutkimuksista peräisin olevia hyvin tunnettuja algoritmeja. On kuitenkin vaikeaa tietää kirjallisuuden perusteella mitä algoritmeja moderneissa palvelinkeskuksissa käytetään, joten vertailu vanhaan algoritmiin ei välttämättä anna arvokasta tietoa tutkijoille, jos algoritmi ei ole käytössä todellisissa palvelinkeskuksissa. Tarkempien ja arvokkaampien tutkimustulosten saavuttamiseksi olisi tärkeää implementoida kehitetyt menetelmät oikeisiin palvelinkeskuksiin ja vertailla tuloksia keskuksessa käytössä oleviin menetelmiin. Tämä edellyttäisi kuitenkin avoimuutta palvelinkeskuksilta keskuksissa käytössä olevista resurssienkäytönoptimointimenetelmistä. Tutkimus todellisessa palvelinkeskuksessa antaisi arvokasta tietoa kehitetyn menetelmän eduista verrattuna käytössä oleviin menetelmiin ja tutkimustulokset olisivat luotettavampia, kuin simulaatiotesteistä saadut tulokset. Tutkimus palvelinkeskuksessa antaisi kehittäjille myös tietoa kehitetyn menetelmän implementoinnin haasteista.

Ajankohtainen tutkimus palvelinkeskusten käyttämistä virtuaalikoneidensijoitus- ja työn tasapainotusmenetelmistä olisi hyödyllistä uusien menetelmien kehitykselle. Palvelinkeskukset saattavat kuitenkin olla vastahakoisia paljastamaan käyttämiään menetelmiä. Monissa tutkimuksissa kehitetyt resurssienhallinta-algoritmit on suunniteltu mielessä jonkin muun ominaisuuden, kuin energiatehokkuuden kehittäminen, kuten palveluiden laadun parantaminen. Tästä syystä artikkeleissa ei välttämättä ole mitattu ja ilmoitettu kehitetyn algoritmin energiatehokkuutta. Lisätutkimus viimeaikaisten menetelmien energiatehokkuudesta ja algoritmien suorituskyvyn vertailu voisi olla aiheellista. Kirjallisuudessa esitellään kymmenittäin algoritmeja ja menetelmiä, jotka lupaavat parantaa palvelinkeskusten tehokkuutta. Tutkimus siitä miksi kyseiset menetelmät eivät ole yleisesti palvelinkeskusten käytössä, voisi tuoda arvokasta näkemystä aihealueeseen. Tutkimus resurssienhallinta-algoritmien käyttöönotosta antaisi arvokasta tietoa käyttöönoton haasteista. Kirjallisuudessa esiintyy usein käsite sovellusten ajamisesta *konteissa*. Kontit ovat uudehko vaihtoehtoinen kevyempi teknologia virtuaalikoneille, joka vaatii lisätutkimusta.

6 LÄHTEET

Afzal, S., & Kavitha, G. (2019). Load balancing in cloud computing–A hierarchical taxonomical classification. *Journal of Cloud Computing*, *8*(1), 1-24.

Ala'Anzy, M., & Othman, M. (2019). Load balancing and server consolidation in cloud computing environments: a meta-study. *IEEE Access*, *7*, 141868-141887.

Assefa, Beakal Gizachew, and Öznur Özkasap. "A survey of energy efficiency in SDN: Software-based methods and optimization models." *Journal of Network and Computer Applications* 137 (2019): 127-143.

Beloglazov, A., Abawajy, J., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future generation computer systems*, *28*(5), 755-768.

Chen, S. L., Chen, Y. Y., & Kuo, S. H. (2017). CLB: A novel load balancing architecture and algorithm for cloud services. *Computers & Electrical Engineering*, *58*, 154-160.

Chen, T., Gao, X., & Chen, G. (2016). The features, hardware, and architectures of data center networks: A survey. *Journal of Parallel and Distributed Computing*, *96*, 45-74

Choudhary, A., Govil, M. C., Singh, G., Awasthi, L. K., Pilli, E. S., & Kapil, D. (2017). A critical survey of live virtual machine migration techniques. *Journal of Cloud Computing*, *6*(1), 1-41.

Dayrathna, M., Wey, Y. & Fan, R. (2015) Data Center Energy Consumption Modeling: A Survey. *IEEE Communications Surveys & Tutorials* 18(1), 732-794

Dey, N. S., & Gunasekhar, T. (2019). A comprehensive survey of load balancing strategies using hadoop queue scheduling and virtual machine migration. *IEEE Access*, *7*, 92259-92284.

Farahnakian, F., Pahikkala, T., Liljeberg, P., Plosila, J., Hieu, N. T., & Tenhunen, H. (2016). Energy-aware VM consolidation in cloud data centers using utilization prediction model. *IEEE Transactions on Cloud Computing*, *7*(2), 524-536.

Ghomi, E. J., Rahmani, A. M., & Qader, N. N. (2017). Load-balancing algorithms in cloud computing: A survey. *Journal of Network and Computer Applications*, *88*, 50-71.

Hammadi, A., Mhamdi, L. (2014). A survey on architectures and energy efficiency in Data Center Networks. Computer Communications (Volume 40, 1 March 2014, Pages 1-21)

Imran, M., Ibrahim, M., Din, M. S. U., Rehman, M. A. U., & Kim, B. S. (2022). Live virtual machine migration: A survey, research challenges, and future directions. *Computers and Electrical Engineering*, *103*, 108297.

Ivanisenko, I. N., & Radivilova, T. A. (2015, October). Survey of major load balancing algorithms in distributed system. In *2015 information technologies in innovation business conference (ITIB)* (pp. 89-92). IEEE.

Kanakala, V. R., Reddy, V. K., & Karthik, K. (2015, March). Performance analysis of load balancing techniques in cloud computing environment. In *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)* (pp. 1-6). IEEE.

Kansal, N. J., & Chana, I. (2016). Energy-aware virtual machine migration for cloud computing-a firefly optimization approach. *Journal of Grid Computing*, *14*(2), 327-345.

Katal, A., Dahiya, S., & Choudhury, T. (2022). Energy efficiency in cloud computing data centers: a survey on software technologies. *Cluster Computing*, 1-31.

Khoshkholghi, M. A., Derahman, M. N., Abdullah, A., Subramaniam, S., & Othman, M. (2017). Energy-efficient algorithms for dynamic virtual machine consolidation in cloud data centers. *IEEE Access*, *5*, 10709-10722.

Kumar, P., & Kumar, R. (2019). Issues and challenges of load balancing techniques in cloud computing: A survey. *ACM Computing Surveys (CSUR)*, *51*(6), 1-35.

Masanet, E., Shehabi, A., Lei, N., Smith, S., & Koomey, J. (2020). Recalibrating global data center energy-use estimates. *Science*, *367*(6481), 984-986.

Masdari, M., & Zangakani, M. (2020). Green cloud computing using proactive virtual machine placement: challenges and issues. *Journal of Grid Computing*, *18*(4), 727-759.

Mishra, S. K., Puthal, D., Sahoo, B., Jayaraman, P. P., Jun, S., Zomaya, A. Y., & Ranjan, R. (2018). Energy-efficient VM-placement in cloud data center. *Sustainable computing: informatics and systems*, *20*, 48-55.

Mishra, S. K., Sahoo, B., & Parida, P. P. (2020). Load balancing in cloud computing: a big picture. *Journal of King Saud University-Computer and Information Sciences*, *32*(2), 149-158.

Noshy, M., Ibrahim, A., & Ali, H. A. (2018). Optimization of live virtual machine migration in cloud computing: A survey and future directions. *Journal of Network and Computer Applications*, *110*, 1-10.

Rawat, Danda B., and Swetha R. Reddy. "Software defined networking architecture, security and energy efficiency: A survey." *IEEE Communications Surveys & Tutorials* 19, no. 1 (2016): 325-346.

Rong, H., Zhang, H., Xiao, S., Li, C., & Hu, C. (2016). Optimizing energy consumption for data centers. *Renewable and Sustainable Energy Reviews*, *58*, 674-691.

Shafiq, D. A., Jhanjhi, N. Z., Abdullah, A., & Alzain, M. A. (2021). A load balancing algorithm for the data centres to optimize cloud computing applications. *IEEE Access*, *9*, 41731-41744.

Shehabi, A., Smith, S., Sartor, D., Brown, R., Herrlin, M., Koomey, J., Masanet, E., Horner, N., Azevedo, I. & Lintner, W. (2016). United states data center energy usage report.

Xia, W., Zhao, P., Wen, Y., & Xie, H. (2016). A survey on data center networking (DCN): Infrastructure and operations. *IEEE communications surveys & tutorials*, *19*(1), 640-656.

Zhang, F., Liu, G., Fu, X., & Yahyapour, R. (2018a). A survey on virtual machine migration: Challenges, techniques, and open issues. *IEEE Communications Surveys & Tutorials*, *20*(2), 1206-1243.

Zhang, J., Yu, F. R., Wang, S., Huang, T., Liu, Z., & Liu, Y. (2018b). Load balancing in data center networks: A survey. *IEEE Communications Surveys & Tutorials*, *20*(3), 2324-2352.

Zhang, P., & Zhou, M. (2017). Dynamic cloud task scheduling based on a two-stage strategy. *IEEE Transactions on Automation Science and Engineering*, *15*(2), 772-783.

Zhang, X., Wu, T., Chen, M., Wei, T., Zhou, J., Hu, S., & Buyya, R. (2019). Energy-aware virtual machine allocation for cloud with resource reservation. *Journal of Systems and Software*, *147*, 147-161.

Zhao, H., Wang, J., Liu, F., Wang, Q., Zhang, W., & Zheng, Q. (2018). Power-aware and performance-guaranteed virtual machine placement in the cloud. *IEEE Transactions on Parallel and Distributed Systems*, *29*(6), 1385-1400.