

Hanna Kortetjärvi

**Reunalaskennan käyttö reaaliaikaisen massadatan
käsittelyssä mobiililaitteissa**

Tietotekniikan Kandidaatintutkielma

18. toukokuuta 2021

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Hanna Kortetjärvi

Yhteystiedot: hanna.k.kortetjarvi@student.jyu.fi

Ohjaaja: Tuomo Rossi

Työn nimi: Reunalaskennan käyttö reaaliaikaisen massadatan käsittelyssä mobiililaitteissa

Title in English: Use of edge computing in real-time big data processing on mobile devices

Työ: Kandidaatintutkielma

Opintosuunta: Tietotekniikka

Sivumäärä: 32+0

Tiivistelmä: Tässä kandidaatintutkielmassa tutkitaan reunalaskennan hyödyntämistä mobiililaitteilla kerätyn reaaliaikaisen massadatan käsittelyssä. Tutkielman tavoitteena on selvittää, kyetäänkö reaaliaikaisen massadatan käsittelyn haasteita ratkaisemaan käsittelemällä dataa lähempänä käyttäjää sekä kuinka reunalaskenta tukee mobiililaitteiden toimivuutta. Tutkielma on kirjallisuuskatsaus. Käymällä läpi artikkeleita ja tutkimuksia, kyettiin selvittämään reunalaskennan tarjoamia hyötyjä. Reunalaskenta kykenee muun muassa vähentämään datan käsittelyn viivettä, lisäämään käyttäjien yksityisyyttä, tukemaan mobiililaitteiden liikkuvuutta sekä vähentämään energiankulutusta.

Avainsanat: Reunalaskenta, Pilvilaskenta, Hajautettu laskentaparadigma, Massadata

Abstract: This bachelor's thesis researches the utilization of edge computing in the processing of real-time big data collected on mobile devices. The aim of the study is to find out whether the challenges of real-time big data processing can be solved by processing the data closer to the user and how edge computing supports the functionality of mobile devices. The study is a literature review. By going through articles and studies, it was possible to find out the benefits of edge computation. Among other things, edge computing is able to reduce data processing latency, increase user privacy, support mobile device mobility, and reduce energy consumption.

Keywords: Edge computing, Cloud computing, Distributed computing paradigm, Big Data

Termiluettelo

Datakeskus	Fyysinen rakennus, joka sisältää useita datan käsittelyä suorittavia tietokoneita.
Esineiden Internet	Internetiin liitettyjä teknisiä laitteita, jotka voivat Internet-yhteyden avulla kerätä ja siirtää tietoa toisilleen.
Hajautettu laskentaparadigma	Arkkitehtuurimalli, jossa datan käsittelyä tapahtuu useilla eri tietokoneilla. Nämä tietokoneet sijaitsevat eri paikoissa, mutta ovat yhteydessä samaan verkkoon.
Massadata	Toiselta nimeltään Big Data. Suuria datajoukkoja, jotka kasvavat jatkuvasti.
Mobiililaite	Kädessä pidettävä tietokone, jota pystytään kuljettamaan paikasta toiseen. Esimerkiksi älypuhelimet ja tabletit.
Pilvilaskenta	Palveluita, joihin käyttäjä pääsee käsiksi Internetin kautta. Datat prosessointi ja analysointi tapahtuu pilvessä tai datakeskuksissa.
Reunalaskenta	Laitteiden keräämän datan prosessointia joko laitteessa tai laitteen lähellä olevilla palvelimilla - eli verkon reunalla.

Kuviot

Kuvio 1. Reunalaskennan arkkitehtuurin kerrokset.....	4
---	---

Sisällys

1	JOHDANTO	1
2	REUNALASKENTA	3
	2.1 Yleistä reunalaskennasta	3
	2.2 Reunalaskennan arkkitehtuuri	3
	2.3 Reunalaskenta ja pilvilaskenta	5
3	REAALIAIKAINEN MASSADATA	7
	3.1 Massadata	7
	3.2 Datan käsittelyn haasteet ja huomioitavat piirteet	8
	3.2.1 Kerääminen	8
	3.2.2 Prosessointi	9
	3.2.3 Siirtäminen	10
	3.2.4 Varastointi	11
	3.2.5 Analysointi	11
	3.2.6 Turvallisuus ja yksityisyys	12
4	DATAN KERÄÄMINEN MOBIILILAITTEILLA	13
	4.1 Mobiililaitteilla kerättävä data	13
	4.2 Mobiililaitteiden ominaisuudet	14
5	REUNALASKENTA VASTAUKSENA MOBIILILAITTEILLA KERÄTYN RE- AALIAIKAISEN MASSADATAN KÄSITTELYN HAASTEISIIN	15
	5.1 Reunalaskennan ja mobiililaitteiden ominaisuuksien yhteensopivuus	15
	5.2 Datan käsittelyn reaaliaikaisuus ja tehokkuus	17
	5.3 Datan luonne	18
	5.4 Skaalautuvuus	19
	5.5 Kustannukset	20
	5.6 Reunalaskennan tarjoama tietoturva ja yksityisyys	21
	5.7 Palvelun laatu ja käyttäjäkokemus	22
6	TULEVAISUUDEN KEHITYSKOHEET	23
7	YHTEENVETO	24
	LÄHTEET	25

1 Johdanto

Mobiililaitteiden määrä kasvaa jatkuvasti, ja näistä pienikokoista tietokoneista on tullut osan monen ihmisen arkipäiväistä elämää. Vuonna 2018 maailmalla arvioitiin olevan 8.8 miljardia mobiililaitteita ja vuoteen 2023 mennessä tämän määrän ennustetaan nousevan 13.1 miljardiin (Cisco 2020). Laitteiden määrän kasvamisen lisäksi niitä myös kehitetään aktiivisesti ja niille luodaan koko ajan uusia sovelluksia. Laitteiden ja sovelluksien kehittyminen saa aikaan muutosta myös niiden tuottamassa datassa - suuria datajoukkoja eli massadataa syntyy vuorokauden ympäri yhä enemmän (Mohamed ja Al-Jaroodi 2014).

Pilvilaskenta (eng. Cloud Computing) on ollut suosittu valinta suurten datamäärien käsittelyssä, sillä siinä kerätty data siirretään käsiteltäväksi pilveen tai datakeskukseen, joissa resurssit riittävät hyvin laajamittaisiin laskentatehtäviin ja massadatan käsittelyyn (Hashem ym. 2015). Pilvellä tarkoitetaan internetin välityksellä saatavia palveluita, kun taas datakeskus on fyysinen rakennus, jossa useat tietokoneet suorittavat datan käsittelyä. Pilvilaskennalla on kuitenkin heikkoutensa. Kun siirretään suurta määrää käsittelemätöntä dataa, tiedon siirron nopeus laskee (Cao, Zhang ja Shi 2018). Reaaliaikaiset sovellukset vaativat kuitenkin viiveetöntä datan siirtämistä, joten hyvän suorituskyvyn ylläpitämiseksi täytyy keksiä vaihtoehtoisia ratkaisuja. Reunalaskenta (eng. Edge Computing) on mahdollinen vaihtoehtoinen laskentaparadigma reaaliaikaisen massadatan käsittelylle. Siinä datan prosessointi toteutetaan joko laitteen lähellä olevilla palvelimilla tai itse laitteessa, jolloin data saadaan nopeammin prosessoitavaksi (Cao, Zhang ja Shi 2018).

Tämä tutkielma on kirjallisuuskatsaus, jossa tutkitaan reunalaskennan käyttöä reaaliaikaisen massadatan käsittelyssä mobiililaitteissa huomioiden myös reunalaskennan haasteet. Esimerkiksi monissa langattomissa laitteissa - kuten mobiililaitteissa - on pienen koon vuoksi rajattu varasto ja laskennallinen teho (Dai ym. 2019), joka vaikuttaa massadatan prosessointiin. Tutkielmassa punnitaan siis reunalaskennan tarjoamia hyötyjä ja haasteita, ja tutkitaan 1) tarjoaako reunalaskenta ratkaisuja mobiililaitteilla kerätyn reaaliaikaisen massadatan käsittelyn haasteisiin ja 2) onko reunalaskennan käyttö yleisellä tasolla hyödyllistä reaaliaikaisen massadatan käsittelyssä mobiililaitteissa.

Ensimmäiseksi luodaan tutkielmalle teoriapohjaa, jotta ymmärretään paremmin tutkittavaa kohdetta. Koska reaaliaikaisen massadatan käsittelyn haasteissa perehdytään sellaisiin ongelmiin, joihin reunalaskennalla voidaan mahdollisesti vaikuttaa, luvussa 2 keskitytään tarkemmin ensiksi reunalaskentaan. Luvussa määritellään mitä reunalaskenta on ja minkälaisista osista sen arkkitehtuuri rakentuu.

Tämän jälkeen luvussa 3 määritellään mitä massadata on ja mitä ominaisuuksia reaaliaikaisuus tuo siihen mukaan. Reaaliaikaisen massadatan määrittelyn jälkeen esitellään datan käsittelyn eri vaiheet ja minkälaisiin haasteisiin niissä voidaan törmätä. Näitä datan käsittelyn vaihteita ovat kerääminen, siirtäminen, prosessointi, analysointi ja varastointi. Lisäksi huomioidaan tietoturva ja käyttäjän yksityisyyden turvaaminen sekä niiden haasteet. Tässä luvussa käsiteltävät haasteet ovat niitä, joihin myöhemmin pyritään löytämään vastauksia reunalaskennasta.

Luvussa 4 tuodaan tarkasteluun mukaan mobiililaitteet ja niiden ominaisuudet. Kaikilla laitteilla ovat omat ominaispiirteensä, jotka tulee huomioida laitteen dataa käsiteltäessä. Mobiililaitteilla näitä piirteitä ovat esimerkiksi laitteen sijainnin muuttuminen käyttäjän sijainnin mukaan ja laitteen pieni koko. Koska tässä tutkielmassa tutkitaan tarkemmin mobiililaitteilla kerättävää dataa, on reunalaskennan hyödyntämisessä otettava huomioon myös se, kuinka hyvin se toimii yhdessä mobiililaitteiden kanssa.

Luvussa 5 lähdetään etsimään vastauksia esitettyihin tutkimuskysymyksiin. Eli luvussa tutkitaan kuinka hyvin reunalaskenta sopii yhteen mobiililaitteiden ominaisuuksien kanssa ja olisiko reunalaskenta potentiaalinen ratkaisu reaaliaikaisen massadatan käsittelyn haasteisiin. Luvussa 6 esitellään kehityskohteita niiden datan käsittelyn haasteiden pohjalta, joihin reunalaskennalla ei ole vielä ratkaisua. Lopuksi yhteenvedossa esitetään tiiviisti käsitellyt teoriaosuudet sekä löydetyt tulokset.

2 Reunalaskenta

Tässä luvussa määritellään reunalaskenta ja esitellään reunalaskennan arkkitehtuurin rakenne. Luvussa perehdytään myös hieman pilvilaskentaan, jotta voidaan paremmin ymmärtää reunalaskennan tärkeys.

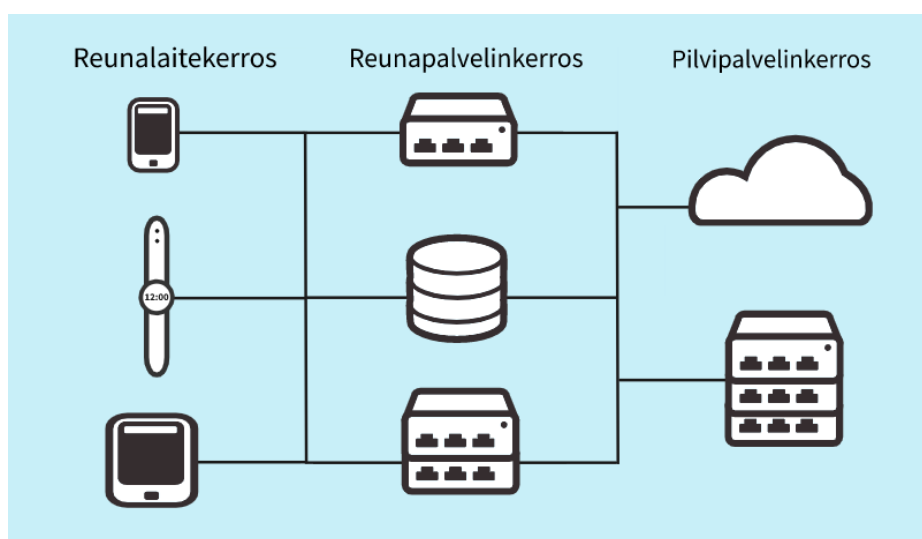
2.1 Yleistä reunalaskennasta

Reunalaskenta on hajautettu laskentaparadigma (eng. distributed computing paradigm), joka mahdollistaa datan käsittelyn joko itse laitteessa tai laitteen lähellä sijaitsevilla palvelimilla (Ahmed ym. 2017). Näitä palvelimia ja käyttäjien käyttämiä laitteita voidaan myös kutsua myös verkon reunaksi, josta reunalaskenta on saanut nimensä. Hajautettu laskentaparadigma viittaa arkkitehtuurimalliin, jossa datan käsittelyä tapahtuu monilla eri tietokoneilla (Coulouris ym. 2011). Nämä tietokoneet sijaitsevat eri paikoissa, mutta ovat kuitenkin yhteydessä samaan verkkoon. Reunalaskennassa laitteen keräämää dataa kyetään siis käsittelemään tarpeen mukaan yhdellä tai useammalla eri alustalla. Tämän lisäksi reunalaskennassa dataa voidaan myös siirtää haluttaessa pilveen.

Hajautuneisuuden lisäksi reunalaskennalla on myös muita tunnusomaisia ominaisuuksia. Reunalaskennan ominaisuuksiin kuuluvat sen tiheä jakautuminen eri alueille maantieteellisesti, laitteiden liikkuvuuden tukeminen ja lyhyt etäisyys käyttäjiin (Ahmed ym. 2017). Näiden ominaispiirteiden olemassaolo on mahdollista reunalaskennan arkkitehtuurin ansiosta. Jotta reunalaskentaa voitaisiin ymmärtää paremmin syvemmällä tasolla, on siis ymmärrettävä mistä osista sen arkkitehtuuri rakentuu.

2.2 Reunalaskennan arkkitehtuuri

Kuviossa 1. on visualisoitu reunalaskennan arkkitehtuuria, eli minkä osien välillä dataa liikkuu. Xiao, Jia, Liu, Cheng (2019) määrittelivät reunalaskennan arkkitehtuurin koostuvan pääosin kolmesta seuraavasta kerroksesta: reunalaittekerroksesta, reunapalvelinkerroksesta ja pilvipalvelinkerroksesta.



Kuvio 1. Reunalaskennan arkkitehtuurin kerrokset, kuvaan otettu mallia (Xiao ym. 2019).

Ensimmäiseksi Xiao ja hänen ryhmänsä määrittivät *reunalaitekerroksen* koostuvan sellaisista teknisistä laitteista, jotka kykenevät esimerkiksi keräämään informaatiota laitteen ulkopuolisesta maailmasta tai kontrolloimaan jotakin siihen yhdistettyä toista laitetta. Reunalaitteet voidaan jakaa kahteen ryhmään: esineiden internetiin (eng. Internet of Things, IoT) eli IoT-laitteisiin ja tässä tutkielmassa käsiteltäviin mobiililaitteisiin.

Esineiden internet koostuu sellaisista laitteista, jotka voidaan liittää langattomasti internetiin ja ne kykenevät keräämään ja siirtämään dataa (Xia ym. 2012). Älykodin laitteisto ja terveyden seurantalaitteet ovat esimerkkejä IoT-laitteista. Mobiililaitteet ovat pienikokoisia laitteita, joita voidaan kuljettaa helposti paikasta toiseen. Mobiililaitteissa on usein kehittyneempi käyttöjärjestelmä verrattuna IoT-laitteisiin, kuten Android ja iOS. Käytössä oleva käyttöjärjestelmä tarjoaa ohjelmointirajapinnan mobiilisovellusten kehittäjille helpottaakseen sovellusten kehitysprosessia (Xiao ym. 2019). Esimerkkejä mobiililaitteista ovat älypuhelimet ja tabletit.

Reunapalvelinkerros sisältää useita erilaisia reunapalvelimia. Nämä palvelimet mahdollistavat aikaisemmin mainitut reunalaskennan tunnusomaiset piirteet. Reunapalvelimia on sijoitettu maantieteellisesti eri alueille ja osa niistä sijaitsee lähempänä käyttäjiä. Käyttäjän käyttämä laite yhdistyy aina lähimpään löydettävään reunapalvelimeen, jolloin laitteen liikkuvuutta on mahdollista tukea (Satyanarayanan ym. 2009). Tältä lähimmältä reunapalvelimel-

ta dataa siirretään eteenpäin kauempana sijaitseville palvelimille. Liikkuvuuden tukemiseen perehdytään enemmän luvussa 5.

Data siirretään yleensä ensimmäisenä tukiasemalle (eng. access point, AP), jonka tehtävänä on seurata ja kontrolloida palvelimien välistä kommunikointia ja datan siirtymistä kohteesta toiseen (Xiao ym. 2019). Tukiasemalta data siirretään seuraavaksi muille reunapalvelimille, joilla taas hoidetaan esimerkiksi datan prosessointia, analysointia ja varastointia. Jos datan määrä on liian suuri eikä yksi reunapalvelin kykene käsittelemään sitä yksin, voi palvelin jakaa työmäärän pienempiin osiin muiden eri reunapalvelimien kanssa. Datan käsittelyn lisäksi reunapalvelinkerros hoitaa myös suurimman osan todennuksista ja lupien antamisesta (eng. authentication and authorization), kun joku yrittää päästä käsiksi dataan (Xiao ym. 2019).

Pilvipalvelinkerroksella sijaitsee pilvi ja datakeskukset. Tämän kerroksen palvelimet sijaitsevat reunapalvelimiin verrattuna pidemmän matkan päässä käyttäjistä. Koska pilvipalveluilla ja datakeskuksilla on riittävät resurssit laajamittaisten laskentatehtävien tekoon, pilvipalvelinkerros hoitaa eniten työtä vaativat työtehtävät ja suurimpien datamäärien varastoinnin. Lisäksi pilvipalvelinkerros käsittelee korkeimman tason todennukset ja lupien antamiset ulkopuolisille toimijoille (Xiao ym. 2019).

2.3 Reunalaskenta ja pilvilaskenta

Reunalaskenta on lähivuosina kasvattanut suosiotaan ja herättänyt tutkijoiden mielenkiinnon. Reunalaskentaa edeltänyt pilvilaskenta ei kykene tyydyttämään kaikkia datan käsittelyn vaatimuksia, jonka takia datan käsittelyä verkon reunalla on tutkittu entistä enemmän. Tästä johtuen reunalaskennan määrittelyn yhteydessä on hyvä myös ymmärtää, miksi reunalaskentaa tarvitaan pilvilaskennan rinnalle.

Pilvilaskennassa datan prosessointi, analysointi ja varastointi toteutetaan joko pilvessä tai datakeskuksissa (Cao, Zhang ja Shi 2018). Eli toisin kuin reunalaskennassa, pilvilaskennassa suurin osa datan käsittelystä on keskitetty yhteen paikkaan. Tämä on mahdollista, sillä pilvessä ja datakeskuksissa on käytettävissä hyvin resursseja intensiivisten prosessointi- ja analysointitehtävien suorittamiseen (Hashem ym. 2015). Pilven ja datakeskusten resurssien takia pilvilaskenta on suosittu vaihtoehto massadatan käsittelyssä.

Toisin kuin reunalaskennassa, jonka arkkitehtuurissa on mukana myös reunapalvelinkerros, pilvilaskennassa kommunikointi tapahtuu suoraan reunalaitekerroksen ja pilvipalvelinkerroksen välillä. Käyttäjän ja datan käsittelyn toteuttavan alustan välimatka on siis pidempi verrattuna reunalaskentaan. Pitkän välimatkan seurauksena pilvilaskennalla on ongelmia tiedonsiirron kanssa: reunalaitteiden tuottaman datan määrä on kasvanut ajan myötä niin suureksi, että dataa ei saada siirrettyä tarpeeksi nopeasti (Cao, Zhang ja Shi 2018). Tämä johtaa siihen, että reaaliaikaiset sovellukset eivät saa käyttöönsä tarvitsemaansa dataa riittävän nopeasti, jolloin sovelluksen toiminta häiriintyy.

Tiedonsiirron nopeuden lisäksi käyttäjän ja pilvipalvelimen välisen etäisyyden takia pilvilaskenta ei kykene tukemaan laitteiden liikkuvuutta yhtä hyvin kuin reunalaskenta kykenee (Mansouri ja Babar 2021). Reunalaskennassa saadaan tieto siitä, minkä reunapalvelimen läheisyydessä käyttäjä on, jolloin se pystyy myös tiedostamaan laitteen liikkeen. Pilvilaskennassa tätä ominaisuutta ei ole, sillä laite kommunikoi koko ajan saman pilvipalvelimen kanssa.

Datan jatkuva kasvu aiheuttaa myös painetta pilvilaskennalle. Internetiin liitettyjä laitteita - kuten kodinkoneita, kodin turvajärjestelmiä ja tehdaslaitteita - kehitetään koko ajan lisää. Näiden IoT-laitteiden kasvun myötä myös verkon reunalla tuotettu data kasvaa jatkuvasti. Jos vain yksi alusta on vastuussa kaiken tämän datan käsittelystä, niin vaikka pilvellä ja datakeskuksilla onkin tällä hetkellä hyvät resurssit datan käsittelyä varten, IoT-laitteiden tuottama data voi aiheuttaa ajan myötä painetta alustalle ja hidastaa sen toimintaa (Cao, Zhang ja Shi 2018).

Mobiililaitteilla kerätyn reaaliaikaisen massadatan näkökulmasta pilvilaskennan nykyisiä ongelmia ovat siis reaaliaikaisuuden takaaminen ja laitteen liikkuvuuden tukeminen. Lisäksi datan määrä voi tulevaisuudessa kasvaa niin suureksi, että pilvipalvelimet eivät yksin kykene käsittelemään dataa tarpeeksi tehokkaasti. Reunalaskenta on kehitetty vastaukseksi näihin haasteisiin. Luvussa 5 perehdytään tarkemmin siihen, kuinka reunalaskenta kykenee ratkaisemaan näitä ongelmia.

3 Reaaliaikainen Massadata

Tässä luvussa käsitellään reaaliaikaista massadataa ja sen käsittelyn haasteita yleisellä tasolla. Datan käsittelyn haasteissa keskitytään ongelmiin, joihin eri laskentaparadigman valinnalla voidaan vaikuttaa.

3.1 Massadata

Massadatalla eli Big Datalla tarkoitetaan suuria datajoukkoja, joita perinteiset tietokantajärjestelmät eivät kykene käsittelemään tarpeeksi tehokkaasti datan suuren määrän takia (McAfee ja Brynjolfsson 2012). Massadataa voidaan kuvata monien eri ominaisuuksien avulla, joista Hariri, Fredericks ja Bowers (2019) määrittivät käytetyimmiksi ominaisuuksiksi viisi V:tä eli määrän, valikoiman, nopeuden, arvon ja totuudenmukaisuuden:

1. *Määrällä (eng. Volume)* viitataan syntyneen datan kokoon, joka kasvaa jatkuvasti.
2. *Valikoima (eng. Variety)* kuvaa datan eri tietomuotoja ja -tyyppejä. Data voi olla jäsenneltyä, lähes jäsenneltyä tai jäsentymätöntä riippuen siitä, kuinka järjestelmällistä ja helposti ymmärrettävissä se on.
3. *Nopeudella (eng. Velocity)* tarkoitetaan sitä, kuinka nopeasti dataa syntyy.
4. Datan *arvo (eng. Value)* kertoo kuinka hyödyllistä data on päätöksenteossa ja mikä datan konteksti on.
5. *Todenmukaisuus (eng. Veracity)* eli datan laatu voidaan jakaa hyvään, huonoon ja määrittelemättömään. Huonossa tilanteessa data voi olla esimerkiksi epäluotettavaa.

Tässä tutkielmassa käsiteltävä mobiililaitteiden reaaliaikainen massadata voidaan määritellä näiden viiden ominaisuuden avulla. Data on suurikokoista, reaaliajassa syntyvää ja ainakin käsittelyprosessin alussa usein jäsentymätöntä. Siinä voi myös esiintyä useita erilaisia tietotyyppiejä. Datan arvo määräytyy sen käyttötarkoituksen mukaan, joten tarkan arvon määrittely kaikille mobiililaitteiden datoilta on vaikeaa monien eri käyttökohteiden takia. Kerätty data on kuitenkin usein suoraa informaatiota käyttäjistä ja käyttäjän käyttäytymisestä, joka nostattaa datan arvoa. Myös todenmukaisuus voi vaihdella datojen välillä. Koska datassa voi esiintyä monia datatyyppiejä sekä dataa voidaan kerätä monipuolisesti erilaisista kohteista,

datan tarkkuus voi olla heikkoa. Tästä seuraten massadata on yleensä normaalia epäluotettavampaa (Hariri, Fredericks ja Bowers 2019).

Reaaliaikaisen massadatan käyttäminen tarjoaa hyötyjä sekä kehittäjälle että käyttäjälle, sillä reaaliaikaisten valintojen avulla sovellukset kykenevät tehostamaan käyttäjille tarjoamia palveluita ja kasvattamaan samalla voittojansa (Mohamed ja Al-Jaroodi 2014). Reaaliaikaisen massadatan käsittelyssä voidaan kuitenkin törmätä erilaisiin ongelmatilanteisiin. Näiden haasteiden tiedostaminen on tärkeää, jotta sovelluksen tai laitteen kehitysprosessin aikana voidaan tehdä päteviä valintoja kyseisten haasteiden välttämiseksi. Seuraavassa alaluvussa käsitellään datan käsittelyn haasteita tarkemmin.

3.2 Datan käsittelyn haasteet ja huomioitavat piirteet

Kun puhutaan reaaliaikaisesta massadatasta, on tärkeä huomioida käsittelyn nopeus. Koska sovelluksen tulee toimia reaaliajassa, täytyy kaikki datan käsittelyn vaiheet toteuttaa nopeasti ja tarkkaa aikataulua seuraten (Mohamed ja Al-Jaroodi 2014; Hariri, Fredericks ja Bowers 2019). Pienikin ylimääräinen viive kykenee tekemään hyvästä sovelluksesta huonon, sillä sovellus ei silloin enää vastaa tarjoamiaan odotuksia. Nopeuden rinnalla datan käsittelyssä pyritään myös kustannusten (Marjani ym. 2017) ja energiakulutuksen minimointiin (Dai ym. 2019). Viiveen, kustannusten ja energiankulutuksen minimointi ovat siis yleisiä tavoitteita datan käsittelyssä. Näiden lisäksi datan käsittelyn eri vaiheilla on myös omia huomiota vaativia asioita.

3.2.1 Kerääminen

Ennen kuin datalle voidaan tehdä mitään, täytyy se ensiksi kerätä. Dataa voidaan kerätä itse laitteesta tai laitteen ulkopuolisesta maailmasta esimerkiksi antureiden avulla.

Sen lisäksi että datan kerääminen on toteuttava reaaliajassa, täytyy myös huomioida, kuinka usein datan keräämistä suoritetaan. Eli vähäisen viiveen lisäksi myös datan aktiivinen kerääminen on tärkeää. Dataa syntyy jatkuvasti lisää, joten keräämisprosessin on kyettävä pysymään datatuotannon perässä (Marjani ym. 2017).

Keräämisessä on tärkeää huomioida datan luonne, sillä data voi olla epäyhtenäistä (eng. heterogeneous) ja usein vasta kerätty data on käsittelemätöntä (Dai ym. 2019; Marjani ym. 2017). Epäyhtenäisyydellä viitataan siihen, kuinka datan sisällössä voi esiintyä eroavaisuuksia, jos keräämiseen käytetyt laitteet poikkeavat toisistaan tai kerätyn datan sisällössä esiintyy erilaisia datatyyppisiä. Käsittelemättömässä datassa taas voi olla todenmukaisuutta alentavia tekijöitä, kuten epävalideja tai toistuvia arvoja. Nämä molemmat ominaisuudet voivat vaikeuttaa ja hidastaa datan käsittelyä mahdollisten satunnaisten ja virheellisten data-arvojen kautta.

Datan luonteen huomioimisen lisäksi keräämisen on oltava tehokasta ja pätevästi toteutettua (Dai ym. 2019). Kerääminen on tärkeää toteuttaa pätevästi, sillä ongelmat keräämisessä eli heti datan käsittelyn alkuvaiheessa voivat saada aikaan ketjureaktion ja negatiivisia vaikutuksia myös myöhemmille vaiheille. Pätevyyteen voivat vaikuttaa erilaiset häiriöt ja häiriötekijät. Esimerkiksi langattomien laitteiden yhteydessä datan keräämistä voi estää sensoreiden akun loppuminen (Dai ym. 2019).

3.2.2 Prosessointi

Datan prosessointi on datan muuntamista järkevämpään muotoon. Prosessointi voidaan halutessa jakaa esikäsittelyyn ja lopulliseen prosessointiin. Esikäsittelyllä tarkoitetaan alustavaa prosessointia, jonka tarkoituksena on vähentää lopullisen prosessoinnin työmäärää.

Keräämisen yhteydessä mainittu epäyhtenäinen ja jäsentymätön data voi aiheuttaa ongelmia datan prosessoinnissa. Datan sisällössä voi esiintyä odottamattomia arvoja, joiden prosessointi vaatii enemmän tarkkaavaisuutta. Prosessoinnissa täytyy siis kyetä löytämään järkevä tapa, jolla voidaan käsitellä erityyppisiä dataa tehokkaasti (Dai ym. 2019).

Esikäsittelyn yhteydessä voidaan suodattaa virheellisiä ja toistuvia arvoja sekä saadaan muuttettua dataa tiiviimpään muotoon (Dai ym. 2019). Näin datan määrä vähenee hieman, jonka ansiosta verkkoliikenteen viive vähentyy. Lisäksi lopullisen prosessoinnin tekevän alustan työmäärää saataisiin pienennettyä eikä alustan tarvitsisi olla niin suuren paineen alla kuin mitä se alun perin olisi ollut. Esikäsittelyn ja varsinaisen prosessoinnin toteuttamiseksi vaaditaan kuitenkin riittävät resurssit. Ongelmia esiintyy esimerkiksi langattomissa laitteissa, joissa on käytettävän laitteiston takia rajallinen laskennallinen teho (Dai ym. 2019).

Resursseja tarvitaan myös prosessoinnille sopivien alustojen kehittämiseen. Koska massadataa syntyy jatkuvasti lisää, prosessointia suorittavien alustojen täytyy kyetä olemaan skaalautuvia (Mohamed ja Al-Jaroodi 2014), eli niiden tulee kyetä mukautumaan esimerkiksi datan määrän tai energiankulutuksen muutoksille. Alustoilla ei kuitenkaan saata aina olla käytössä tarpeeksi päteviä resursseja skaalautuvuuden toteuttamiseksi.

3.2.3 Siirtäminen

Käytettävän laskentaparadigman arkkitehtuurin osien - kuten laitteen, varaston ja palveluntarjoajan - välillä voi tapahtua kommunikointia datan eri käsittelyvaiheiden välillä. Kun data on kerätty, voidaan se siirtää paikkaan, jossa prosessointi suoritetaan. Tai kun data on prosessoitu ja analysoitu, lähetetään analysoinnin tulos sovellukselle ja data siirretään varastoon tallennettavaksi.

Koska sovellukset käsittelevät reaaliaikaista dataa, tulee informaation siirtämisen tapahtua reaaliajassa (Mohamed ja Al-Jaroodi 2014). Tiedonsiirrossa esiintyvä viive on siis minimoitava joka kerta, kun dataa siirretään paikasta toiseen. Siirrettävä informaatio voi kuitenkin vaikuttaa datan siirtämisen nopeuteen. Tiedonsiirrossa on rajattu kapasiteetti (Dai ym. 2019), eli yhdellä kerralla voidaan siirtää vain tietty määrä dataa. Liian suuri datan määrä johtaa siihen, että tiedon siirtäminen tapahtuu toivottua hitaammin.

Nopeuden lisäksi on huomioitava myös tiedonsiirron tehokkuus. Tehokkuuteen vaikuttavat kaistanleveys eli kuinka nopeasti viesti saadaan välitettyä vastaanottajalle sekä kuinka paljon energiaa tiedonsiirtoon kuluu (Dai ym. 2019). Energiankulutusta on tärkeä seurata varsinkin langattomissa verkoissa, sillä langattomien verkkojen laitteilla ja sensoreilla on usein rajattu määrä energiaa käytössä. Rajatun energiamäärän takia langattomille verkoille pyritään kehittämään energiatehokkaita vaihtoehtoja tiedonsiirrolle (Dai ym. 2019), mutta energiankulutus on hyvä huomioida myös muissakin verkoissa. Datan siirtämisessä pyritään siis vähentämään energiankulutusta, jotta siirtäminen saataisiin toteutettua tehokkaasti.

3.2.4 Varastointi

Varastoinnissa data siirretään talteen myöhempää käyttöä varten. Tietovarastoina voivat toimia esimerkiksi laitteen muisti, pilvipalvelu tai datakeskus. Haasteita esiintyy varastoinnissa erityisesti varastojen kapasiteetin ja hallinnan kanssa (Mohamed ja Al-Jaroodi 2014). Koska tietokannoilla on rajattu kapasiteetti, ne kykenevät varastoimaan vain tietyn määrän dataa. Suuren määrän lisäksi dataa syntyy koko ajan lisää, jonka takia eri hallintatehtävien suorittaminen, kuten datajoukkojen järjestäminen ja halutun datan hakeminen on vaikeampaa.

Mohamed ja Al-Jaroodi (2014) toivat myös esille sen, kuinka käyttäjien määrä voi vaikuttaa varastointiin. Vaikka yhden käyttäjän tuottama data ei olisi itsessään suurta, voi sovelluksen suuri käyttäjämäärä nostaa käsiteltävän datan kokonaismäärää. Tällöin useamman henkilön datan organisoinnissa voi esiintyä haasteita varsinkin, kun kaikkien käyttäjien datan organisointi täytyy toteuttaa samanaikaisesti ja reaaliajassa.

Koska varasto sisältää suuren määrän dataa, voi varasto joutua käsittelemään useita samanaikaisesti tapahtuvia sisäänpääsyjä ja kyselyitä (Dai ym. 2019). Jotta tämä saataisiin toteutettua sujuvasti, varastoinnissa on tärkeä huomioida varaston kolme ominaisuutta: skaalautuvuus, luotettavuus ja tehokkuus. Isot datamäärät voivat aiheuttaa haasteita skaalautuvuuden kanssa, sillä datamäärät voivat kasvaa loputtomasti (Hariri, Fredericks ja Bowers 2019). Käytettävissä oleva tietokantatyökalu ei kuitenkaan välttämättä kykene mukautumaan jatkuvasti kasvavaan datamäärään. Varaston täytyy myös kyetä olemaan luotettava, ja varmistaa datan pysyvyys varastossa. Haasteena on luotettavuuden varmistaminen niin, että kustannukset eivät ole liian korkeat (Guerra ym. 2011). Viimeiseksi kuten datan käsittelyn muutkin vaiheet, on varastoinninkin tapahduttava tarpeeksi tehokkaasti.

3.2.5 Analysointi

Datan analysointi sisältää datan tutkimista ja arviointia, jotta sovellukset ja laitteet voivat tehdä päteviä päätöksiä parantaakseen toimintaansa. Tässä vaiheessa datan käsittelyä datasta voidaan löytää esimerkiksi normaalista poikkeavia muutoksia sovelluksen toiminnassa, joten analysointi on suoritettava nopeasti ja luotettavasti (Mohamed ja Al-Jaroodi 2014). Lisäksi reaaliaikaiset sovellukset vaativat nopeaa analysointia (Dai ym. 2019), joten analysoinnis-

sa ei saisi esiintyä viivettä. Analysointiin vaadittava aika on kuitenkin yleensä pitkä, sillä tarkasteltavan datan määrä on suuri ja päätöksenteko on tärkeä toteuttaa huolellisesti.

Nopeuden lisäksi analysoinnin suorittavalta alustalta odotetaan tehokasta toimintaa. Tehokkuutta voivat kuitenkin laskea erilaiset tekijät. Koska käsitellään suurta määrää dataa reaaliajassa, voi analysointi olla intensiivinen vaihe ja vaatia alustalta korkeaa suorituskykyä (Mohamed ja Al-Jaroodi 2014). Hyvän suorituskyvyn takaamiseksi on otettava huomioon alustan käytettävissä olevat resurssit, kuten laitteisto ja ohjelmisto.

Datan määrän lisäksi datan valikoima on huomiota vaativa ominaisuus analysoinnissa. Jäsentymätöntä dataa, eli esimerkiksi kun se sisältää useita vaikeasti luettavia datatyyppisiä, on vaikeampaa käydä läpi (Hariri, Fredericks ja Bowers 2019). Analysoinnin tehokkuus voi siis laskea teknologisten resurssien lisäksi myös datan sisällön takia.

3.2.6 Turvallisuus ja yksityisyys

Kaikissa mainituissa datan käsittelyn vaiheissa on tärkeää, että ne huolehtivat tietoturvasta ja suojaavat käyttäjän yksityisyyttä. Jotta laitteiden käyttö olisi turvallista, niiden täytyy kyetä suojelemaan käyttäjän dataa erilaisilta turvallisuus- ja yksityisyysuhilta (Marjani ym. 2017).

Laitteiden lisäksi datan käsittelyä tapahtuu muillakin alustoilla, kuten reuna- ja pilvipalvelimillä. Koska datan käsittelyä tapahtuu monissa eri paikoissa, on tietoturvasta pidettävä huolta kaikilla käsittelyä suorittavilla alustoilla reaaliaikaisesti (Dev Mishra ja Beer Singh 2016). Varastoinnissa täytyy esimerkiksi kyetä estämään luvaton pääsy tietokantaan. Myös datan siirtäminen eri alustojen välillä on otettava huomioon, sillä kommunikointia voi seurata ulkopuoliset salakuuntelijat (Dai ym. 2019).

Laitteet voivat sisältää käyttäjän arkaluontoista informaatiota. Henkilökohtaista dataa on kyettävä suojelemaan ulkopuolisilta henkilöiltä (Dai ym. 2019). Lisäksi jotkut sovellukset voivat vaatia toimiakseen lupaa tiettyjen tietojen käyttöön, jolloin on tärkeää turvata käyttäjän anonymisuus. Mishra ja Singh (2016) tuovat kuitenkin esille anonymisuuden varmistamisen haasteen. Vaikka data pyrkisikin olemaan anonymiä, voi se silti sisältää vihjeitä käyttäjän henkilöllisyydestä. Tällöin analysoidusta datasta olisi mahdollista selvittää käyttäjän henkilöllisyys.

4 Datan kerääminen mobiililaitteilla

Tässä luvussa käsitellään tarkemmin mobiililaitteita. Ensimmäiseksi käsitellään, minkälaista dataa mobiililaitteilla voidaan kerätä. Sen jälkeen perehdytään tarkemmin mobiililaitteiden tyypillisimpiin ominaisuuksiin.

4.1 Mobiililaitteilla kerättävä data

Mobiililaitteet ovat pienikokoinen laite, jota käyttäjä voi kuljettaa helposti paikasta toiseen. Mobiililaitteet ovat vaikuttaneet suuresti massadatan määrän kasvuun (Mohamed ja Al-Jaroodi 2014), sillä niillä voidaan kerätä paljon erilaista dataa. Laitteista kyetään keräämään informaatiota itse laitteesta eli järjestelmädataa sekä informaatiota laitteen käyttäjästä eli käyttäjädataa (Dai ym. 2019). Mobiililaitteet keräävät dataa laitteen sisältä sekä laitteen ympäristöstä erilaisten antureiden (eng. sensor) avulla. Anturit mahdollistavat esimerkiksi kasvojen tunnistuksen älypuhelimissa, laitteen nopeuden määrittämisen liikkeessä ja laitteen sijainnin löytämisen.

Anturit ovat hyödyllisiä, sillä niillä voidaan parantaa laitteiden käytettävyyttä ja käyttäjäkokemusta. GPS (eng. Global Positioning System) on hyvä esimerkki antureiden hyödyllisyydestä. GPS on navigaatiojärjestelmä, jonka avulla voidaan paikantaa laitteen sijainti. Monista mobiililaitteista löytyy antureita, jotka ottavat vastaan GPS-radiosignaaleja. GPS:sää voidaan hyödyntää muun muassa väkijoukon hallinnassa onnettomuuksien ja katasrofiien aikana (Mohamed ja Al-Jaroodi 2014), sillä sen avulla voidaan paikantaa vaara-alueen lähistöllä olevat ihmiset ja viestittää heille informaatiota tilanteesta. Myös videopelit voivat hyödyntää GPS:sää parantaen pelaajien pelikokemusta. Pokémon Go on 2016 julkaistu Niantic-yrityksen kehittämä mobiilipeli, joka käyttää hyödykseen laitteen sijaintitietoja. Sijaintitietojen avulla Pokémon Go luo pelaajalle mukaansatempaavan pelikokemuksen (Premsankar, Di Francesco ja Taleb 2018), sillä pelaaja kykenee omalla liikkumisellaan vaikuttamaan pelin etenemiseen.

4.2 Mobiililaitteiden ominaisuudet

Datan käsittelyssä on tärkeää ottaa myös huomioon dataa käyttävän laitteen ominaisuudet. Tämän tutkielma käsittelee reaaliaikaista massadataa mobiililaitteen yhteydessä, joten tämä alaluku keskittyy tarkemmin mobiililaitteiden huomioitaviin ominaisuuksiin.

Mobiililaitteen käsitteen määritelmästä kyetään tunnistamaan kaksi mobiililaitteiden tunnusomaista piirrettä: *laitteen pienikokoisuus* ja käyttäjän kyky kuljettaa laitetta helposti mukansa eli *laitteen liikkuvuus*. Mobiililaitteet kykenevät vaihtamaan sijaintia kulkiessaan käyttäjän mukana, joten laskentaparadigman on kyettävä tukemaan laitteen liikkuvuutta (Ahmed ja Rehmani 2017). Mobiililaitteen pieni koko taas tarkoittaa, että käytettävissä olevaa laitteistoa on rajoitettu (Dai ym. 2019) eikä laitteessa voi käyttää osia, jotka kykenevät suorittamaan kaikkein vaativimpia laskentatehtäviä. Laitteisto rajaa myös laitteen sisäisen muistin tilaa ja akkua. Laitteessa voi olla intensiivisiä sovelluksia, jotka vaativat paljon energiaa ohjelman suorittamiseen ja samalla syövät laitteen akkua (Ahmed ym. 2017).

Vaikka liikkuvuus ja laitteen koko ovat näkyvimmit ominaisuudet mitä mobiililaitteissa on, näiden lisäksi on tärkeää kiinnittää huomiota myös laitteella käsiteltävään dataan. Mobiililaitteiden kommunikointi muiden laitteiden ja tietokannan kanssa on monipuolista ja aktiivista (Cao, Zhang ja Shi 2018). Laite on samaan aikaan *datan tuottaja sekä käyttäjä*: käyttäjän laitteelle siirretään informaatiota ja käyttäjä kykenee laitteellansa myös tuottamaan dataa. Mobiililaitteet vaativat siis sen, että tiedonsiirrossa kyetään samanaikaisesti ottamaan vastaan pyydettyä dataa sekä siirtämään dataa toiselle alustalle ilman verkon rasittumista.

Laitteissa sijaitsevassa datassa korostuu myös *epäyhtenäisyys*, sillä esimerkiksi älypuhelimilla käsitellään paljon erilaisia datatyyppisiä (Dai ym. 2019). Esimerkkejä näistä datan eri tyypeistä ovat video-, kuva- ja tekstitiedostot. Laitteet ovat myös hyvin ihmisläheisiä, ja keräävät käyttäjästä henkilökohtaista informaatiota. Ne voivat siis sisältää paljon *arkaluontoista dataa* (Dai ym. 2019) lisäten yksityisyyden ja tietoturvan tärkeyttä. Anshari ja Alas (2015) toivat myös esille käyttäjäkokemuksen tärkeyden. Koska mobiililaitteet ovat osa ihmisten arkipäiväistä elämää ja ovat aktiivisesti käytössä, on tärkeää luoda käyttäjille paras mahdollinen kokemus laitteesta. Käyttäjäkokemusta voidaan parantaa esimerkiksi nopeammilla ja tehokkaammilla palveluilla.

5 Reunalaskenta vastauksena mobiililaitteilla kerätyn reaaliaikaisen massadatan käsittelyn haasteisiin

Tässä luvussa tarkastellaan, onko reunalaskenta pätevä vaihtoehto ratkaisemaan datan käsittelyn haasteita sekä kykeneekö reunalaskenta tukemaan mobiililaitteiden toimintaa. Datan käsittelyn haasteissa huomataan, että samoja haasteita voi esiintyä joko datan käsittelyn eri vaiheissa tai arkkitehtuurin eri osilla. Eli sen sijaan, että datan käsittelyn vaiheita käytäisiin yksi kerrallaan läpi, tässä luvussa haasteita tarkastellaan omina kokonaisuuksinaan.

5.1 Reunalaskennan ja mobiililaitteiden ominaisuuksien yhteensopivuus

Ennen kun tarkastellaan tarkemmin reunalaskennan käyttöä reaaliaikaisen massadatan käsittelyssä, on hyvä ensimmäisenä selvittää kuinka yhteensopivia reunalaskenta ja mobiililaitteet ovat keskenänsä. Mobiililaitteiden huomioitavia ominaisuuksia ovat edellisessä luvussa mainitut ominaisuudet, eli laitteen liikkuvuus, laitteen pieni koko sekä laitteen kyky tuottaa ja käyttää dataa samanaikaisesti. Datan epäyhtenäisyys ja arkaluonteisuus ovat suuria haasteita reaaliaikaisen massadatan käsittelyn vaiheissa ja käyttäjän yksityisyyden turvaamisessa, joten näitä käsitellään tarkemmin seuraavassa alaluvussa.

Reunalaskentaa käsittelevässä luvussa mainittiin lyhyesti, että reunalaskentaa käyttävät alustat tukevat laitteiden liikkuvuutta reunal palvelimien avulla. Cloudletit ja mikrodatakeskukset (eng. Micro Data Center, MDC) ovat esimerkkejä niistä reunal palvelimista, joita voidaan hyödyntää liikkuvuuden tukemisessa (Cao, Zhang ja Shi 2018). Cloudlet ja mikrodatakeskus ovat pienikokoisemmat versiot pilvipalveluista ja datakeskuksista, joten ne voivat hoitaa osaa pilvipalveluiden ja datakeskuksien työtehtävistä pienemmillä resursseilla. Mobiililaitteen liikkussa alueelta toiselle, se yhdistyy aina lähimpään cloudlettiin (Satyanarayanan ym. 2009). Jos cloudlet ei löydy tarpeeksi lähellä, laite voi jatkaa toimintaansa kauempana sijaitevan pilvipalvelun tai omien resurssiensa avulla. Cloudlet ja mikrodatakeskus ovat osa reunal palvelinkerrosta, joten ne sijaitsevat lähellä käyttäjiä. Lyhyen välimatkan takia käyttäjät pääsevät nopeasti käyttämään näitä palveluita ilman viivettä.

Mobiililaitteen pienen koon takia laitteella on rajatut resurssit laskentatehtävien suorittamiseen. Työtehtävät olisi kyettävä suorittamaan ilman, että viivettä esiintyisi eikä laite kuluttaisi liikaa energiaa ja akkua. Vaikka reunapalvelimilla ja -laitteilla ei yksinään ole välttämättä riittäviä resursseja intensiivisten tehtävien suorittamiseen, voidaan tehtäviä jakaa osiin hajautetun järjestelmän eri osien välille. Ha, Chen, Hu, Richter, Pillai ja Satyanaryanan (2014) suorittivat tutkimuksen, jossa he seurasivat, kuinka kauan cloudleteilla kesti suorittaa pienempiin osiin hajautettu työmäärä. Lisäksi he suorittivat cloudlettien kesken jaetun työmäärän myös pilvessä, ja vertasivat cloudlettien ja pilven laskentanopeuksia keskenään.

Tutkimuksesta nähtiin, että cloudlettien avulla saatiin aikaan 80 ms - 200 ms parannus viiveen vähentämisessä verrattuna pilveen. Samassa tutkimuksessa Ha ja hänen tiiminsä tutkivat myös energiankulutusta. Heidän tutkimustuloksensa perusteella jakamalla työmäärä cloudlettien kesken sen sijaan, että työmäärä siirrettäisiin tehtäväksi pilveen, energiankulutusta saatiin vähennettyä 30-40 %. Mobiililaitteet voivat hyödyntää tätä tietoa, sillä vähentämällä energiankulutusta saadaan parannettua akunkestoa. Eli tutkimuksesta nähdään, että reunalaskennalla on arkkitehtuurinsa avulla resurssit myös suurten työtehtävien suorittamiseen (Ha ym. 2014).

Tiedonsiirrossa voi esiintyä ongelmia, jos mobiililaitte keskustelee suoraan pilven kanssa ilman reunapalvelimia. Laite on datan tuottaja sekä käyttäjä, joten tiedonsiirtoa tapahtuisi molempiin suuntiin pilven ja laitteen välillä. Tiedonsiirron kapasiteetti on kuitenkin rajattu, eli haluttuja datamääriä ei kyetä siirtämään tarpeeksi tehokkaasti ja nopeasti. Lisäksi välimatka laitteen ja pilven välillä on pitkä. Cao, Zhang ja Shi (2018) esittivät, miksi reunalaskenta olisi hyvä ratkaisu tiedonsiirron nopeuttamiseen ja tehostamiseen. Reunalaskennan hajautettu arkkitehtuuri mahdollistaa sen, että dataa voidaan käsitellä verkon reunalla ennen tiedonsiirtoa. Kun data käsitellään laitteen lähellä ja käsittelyn jälkeen vasta siirretään pilveen, data pystyy kulkemaan välimatkan nopeammin, koska käsittelyn jälkeen siirrettävää dataa on vähemmän. Kuitenkin jos kaikkien pyydettyjen työtehtävien suorittaminen käy verkon reunalla liian raskaaksi, niin työmäärää voidaan hajauttaa osiin reunapalvelimille.

Reunalaskenta kykenee siis tukemaan hyvin edellä käsiteltyjä mobiililaitteiden ominaisuuksia. Seuraavana vaiheena on ottaa mukaan tarkasteltavaksi mobiililaitteella kerätty reaaliaikainen massadata, ja tutkia kuinka hyvin reunalaskenta kykenee toteuttamaan datan käsittelyä.

5.2 Datan käsittelyn reaaliaikaisuus ja tehokkuus

Reunalaskenta kykenee arkkitehtuurinsa ja ominaisuuksiensa avulla tukemaan sovellusten reaaliaikaisuutta. Reunalaskennan ja mobiililaitteiden ominaisuuksien yhteensopivuus -alalu- vussa käsiteltiin esikäsittelyn ja työmäärän jakamisen eri palvelimien kesken vaikutuksesta nopeuteen. Näiden lisäksi palvelimien fyysisen sijainnin avulla voidaan vaikuttaa nopeuteen. Jos datan käsittely suoritetaan laitteen sijasta palvelimella, reunapalvelimen lyhyen etäisyy- den takia viive saadaan minimoitua (Satyanarayanan 2017), sillä myös datan kulkema matka on lyhyt. Reunapalvelin saa tällöin datan nopeasti käyttöönsä ja pääsee aloittamaan työteh- tävänsä toivotussa aikataulussa. Näin samalla saadaan tehostettua käsittelyprosessia (Cao, Zhang ja Shi 2018).

Tiedonsiirtoa tapahtuu reunalaskennassa usean eri alustan välillä. Tämän takia on varmis- tettava, että nopeuden lisäksi data saadaan siirrettyä järkevästi ilman, että verkon toiminta heikentyy tiedonsiirron aikana. Esikäsittelyn suorittamisella ennen tiedonsiirtoa voidaan vä- hentää tiedonsiirron aikana esiintyviä ongelmia. Kun data saadaan tiivistettyä pienemmäksi tiedostoksi, niin tiedonsiirron nopeuden kasvamisen lisäksi sen siirtäminen aiheuttaa pie- nempää painetta verkolle (Cao, Zhang ja Shi 2018).

Kun käsitellään reaaliajassa suuria datajoukkoja, suoritettavat laskentatehtävät voivat olla laajamittaisia ja kuluttaa paljon energiaa. Laitteilla voi kuitenkin olla käytössä rajatut ener- gialähteet. Mobiililaitteiden yhteydessä kerrottiin, että jakamalla työmäärää lähellä olevien reunapalvelimien kanssa saatiin vähennettyä energiankulutusta. Kun vältetään liian laaja- mittaisten ja monimutkaisten työtehtävän toteuttamista itse laitteella ja siirretään työtehtä- vä suoritettavaksi reunapalvelimelle, saadaan pidennettyä reunalaitteen akunkestoa (Prem- sankar, Di Francesco ja Taleb 2018). Ennen työmäärän siirtämistä on kuitenkin huomioita- va laskentatehtävän siirtoon kuluva energiamäärä, sillä siirtämiseenkin kuluu energiaa (Shi ym. 2016). Jos joudutaan siirtämään suurta työmäärää paikasta toiseen, voi energiankulu- tus nousta suuremmaksi kuin mitä se olisi ollut laitteessa toteutettavassa datan käsittelyssä. Eli energiankulutuksen näkökulmasta on tärkeä selvittää, kuluisiko energiaa enemmän datan käsittelyssä laitteessa vai laskentatehtävän siirtämisessä palvelimelle.

Datan käsittelyn toteuttamisella eri palvelimien kesken on hyötyä myös tehokkuuden paran-

tamisella yleisellä tasolla. Intensiivisten työtehtävien suorittamisen lisäksi tehtävien jakamisella reunal palvelimien kesken saadaan myös maksimoitua laitteen tai alustan suoritettavien tehtävien määrä (Ahmed ym. 2017). Datan käsittelyssä voidaan myös priorisoida tehtäviä niiden kiireellisyyden mukaan. Erityisesti hätätilanteissa data täytyy saada mahdollisimman nopeasti käsiteltyä, jolloin datan käsittely kannattaa suorittaa lähellä käyttäjää. Analyysin suorittaminen voi olla raskas tehtävä, mutta laite ei välttämättä aina tarvitse tarkkoja analyysin tuloksia reaaliajassa tietoonsa. Tällöin tehtävä voidaan siirtää pilveen asti, jossa se voidaan suorittaa mahdollisen viiveen kanssa (Ahmed ym. 2017).

5.3 Datan luonne

Datan luonne voi hidastaa ja vaikeuttaa datan käsittelyä. Artikkeleita ja tutkimuksia läpikäymällä huomattiin, että reunalaskennalla ei ole suoraa vastausta helpottamaan epäyhtenäisen datan käsittelyä. Reunalaskennan nopeus ja työtehtävien hajauttaminen voivat yleisesti parantaa reaaliaikaisen massadatan käsittelyä, kun taas palvelimien ja laitteiden vähäiset resurssit voivat hidastaa datan käsittelyä. Voidaan siis olettaa, että nopeus, hajautettu arkkitehtuuri sekä resurssit vaikuttavat myös epäyhtenäisen datan käsittelyyn.

Käytettävillä algoritmeilla voidaan kuitenkin parantaa epäyhtenäisen datan käsittelyä. Reunalaskenta tarvitsee hajautettuja algoritmeja, joiden avulla datan käsittelyä voidaan tehdä jokaisella reunalaitteella ja -palvelimella itsenäisesti niin, että jokaisella alustalla käsitellyt tiedot ovat mukana lopullisessa päätöksenteossa (Mansouri ja Babar 2021). Näiden hajautettujen algoritmien tulee kyetä huomioimaan epäyhtenäinen data sekä niiden suoritus aika ja -tehokkuus tulee olla tarpeeksi hyviä. Eli epäyhtenäisen datan käsittelyn parantamiseksi olisi hyvä kehittää hajautettuja algoritmeja, jotka ovat luotettavia ja kykenevät mukautumaan jatkuvasti kasvavaan massadatan määrään.

Datan käsittelyn luotettavuuden ja pätevyyden takaamisessa voi esiintyä haasteita reunalaskennassa. Tämä johtuu datan todenmukaisuutta alentavista tekijöistä sekä reunalaskennan hajautetusta arkkitehtuurimallista. Reunalaskennassa on käytössä useita eri alustoja, jotka toimivat yhteistyössä tarjotakseen palvelun tai palveluita käyttäjälle. Nämä alustat kuitenkin suorittavat itsenäisesti oman osuutensa laskentatehtävistä eivätkä välttämättä ole tietoisia da-

tan käsittelyn kokonaiskuvasta, ellei käytössä ole hajautettuun järjestelmään sopivia algoritmeja (Mansouri ja Babar 2021). Epäyhtenäisen datan lisäksi myös pätevyuden takaamiseksi tarvitaan siis hajautettuihin järjestelmiin kehitettyjä algoritmeja.

Käsittelyn luotettavuus voi laskea, jos itse käsiteltävä data on epäluotettavaa tai epätodennukaista. Datan luotettavuus voi heikentyä, jos datan keräämisessä esiintyy häiriötekijöitä, kuten antureiden toimintaa häiritsevät tekijät tai heikko langaton yhteys (Cao, Zhang ja Shi 2018). Langattoman yhteyden pettämisen aiheuttamat ongelmat voidaan kuitenkin välttää, jos informaatiota tallennetaan tietyn väliajoin ja siitä seuranneiden tehtävien epäonnistumiset suoritetaan uudelleen (Ahmed ym. 2017). Näiden häiriötilanteiden huomioinnilla ja ennaltaehkäisyllä voidaan parantaa reunalaskennan luotettavuutta.

5.4 Skaalautuvuus

Skaalautuvuutta voidaan tarkastella reunalaskennassa monesta eri näkökulmasta. Verkoilla voi olla erilaisia kuuluvuusalueita. Verkko voi esimerkiksi olla keskitetty pienemmän yhteisön ympärille, tai vastakohtana verkon alueena voi olla koko kaupunkialue. Reunalaskenta kykenee skaalautumaan erikokoisille alueille (Shi ym. 2016), riippuen siitä kuinka reunal palvelimia sijaitetaan maantieteellisesti. Huomiota skaalautuvuudessa kuitenkin vaativat verkkoon yhdistettyjen mobiililaitteiden määrä ja niiden tuottama massadata.

Verkkoon voi olla yhdistettynä suuri määrä mobiililaitteita ja niiden liikkuvuuden takia verkkoon yhdistettyjen laitteiden lukumäärä voi jatkuvasti muuttua. Laitteiden muuttuva määrä tulisi huomioida järjestelmien kehityksessä, mutta esimerkiksi suunniteltaessa reunalaskennassa käytettäviä tietoturvajärjestelmiä saatetaan usein jättää huomioimatta mobiililaitteiden liikkuvuus ja järjestelmän skaalautuvuus aktiivisesti muuttuvaan laitemäärään (Liu ym. 2019). Laitteiden määrä voi kuitenkin vaikuttaa laitteen tai järjestelmän suorituskykyyn, joten se tulisi jollain tapaa ottaa kehitysvaiheessa huomioon.

Reunalaskentaa käyttävien alustojen tulisi kyetä mukautumaan myös muuttuvaan datamäärään kaikissa datan käsittelyn vaiheissa. Alustojen rajalliset laitteistot ja laskennallinen teho vaikeuttaa tämän ehdon toteutumista. Vaikka työmäärää voidaan jakaa osiin, laitteet ja palvelimet eivät kykene mukautumaan reunalaskennassa yhtä hyvin suureen työmäärään

verrattuna pilvilaskentaan (Ahmed ja Ahmed 2016).

Skaalautuvuutta voidaan kuitenkin parantaa reunalaskennassa eri keinoin. Reunalaskennan monet ominaisuudet ovat olemassa sen reunapalvelimien hajautuneisuuden seurauksena. Joten lisäämällä uusia reunapalvelimia lähelle käyttäjiä, saadaan tarjottua reunalaskentaa käytävälle sovellukselle tai laitteelle uusia resursseja käyttöön (Ahmed ym. 2017). Myös pilvipalvelinkerrosta voidaan hyödyntää skaalautuvuuden takaamiseksi. Koska pilvellä ja datakeskuksilla on käytössä paremmat laskentatehot reunapalvelimiin verrattuna, laskentatehtävien kasvaessa voidaan käyttää hyödyksi pilvipalvelimien resursseja (Ahmed ym. 2017). Lisäksi reaaliaikaisuuden ja tehokkuuden yhteydessä mainittu palvelimien resurssien lisääminen mahdollistaa intensiivisempien laskentatehtävien suorittamisen. Eli paremmilla resursseilla alustat kykenevät paremmin myös mukautumaan kasvaviin datamääriin.

5.5 Kustannukset

Sovelluksen ja laitteen kehittäjien näkökulmasta on hyvä huomioida myös reunalaskennan kustannukset. Käsiteltäessä reaaliajassa kasvavaa massadataa kustannukset voivat nousta korkeiksi tarvittavien resurssien takia. Reunalaskennan avulla voidaan vähentää tiedonsiirtoon meneviä kuluja reunapalvelimien läheisyyden ja lyhyiden välimatkojen avulla (Shi ym. 2016). Lisäksi esikäsittelyllä tuotetut tiivistetyt datapaketit eivät aiheuta tiedonsiirrolle niin suurta työmäärää kuin mitä ne olisivat alkuperäisessä koossa tuottaneet.

Reunapalvelinkerroksella sijaitsevia tukiasemia voidaan hyödyntää myös välimuistina mobiililaitteille. Käyttäjän käyttäessä mobiililaitetta verkossa, laitteella avatut sisällöt voidaan tallentaa tukiaseman välimuistiin alentaen palveluntarjoajan verkkokustannuksia (Ahmed ja Rehmani 2017). Lisäksi reunalaskentaa hyödyntämällä voidaan myös nostaa sovelluksen tai laitteen tuottamia voittoja. Vähentämällä viivettä ja energiankulutusta voidaan vaikuttaa positiivisesti käyttäjäkokemukseen ja parantaa laitteen toimivuutta (Shi ym. 2016), jotka vuorostaan mahdollisesti kasvattavat tuottoja.

Yksi huomioon otettava asia reunalaskennassa on se, että hajautettu arkkitehtuuri voi vuorostaan olla kustannuksia nostattava tekijä. Koska pilvilaskennassa datan käsittely on keskittynyt joko pilveen tai datakeskukseen, sitä on kokonaisuutena helpompi hallita ja vaatii

pienemmät kustannukset hallintajärjestelmien suhteen (Satyanarayanan 2017). Eli kustannuksien näkökulmasta täytyy tarkkailla eri tekijöitä, jotka voivat vaikuttaa kustannuksiin, sekä miettiä niitä suhteessa mahdollisiin tuottoihin.

5.6 Reunalaskennan tarjoama tietoturva ja yksityisyys

Reunalaskenta kykenee huomioimaan käyttäjän yksityisyyden niin, että käyttäjän henkilökohtainen informaatio pysyy turvassa. Esikäsittelemällä tai prosessoimalla data käyttäjän lähellä sen sijaan, että käsittelemätöntä dataa kuljetettaisiin pilveen, kyetään suojaamaan käyttäjän yksityisyyttä (Cao, Zhang ja Shi 2018). Tämä onnistuu muuttamalla dataa niin, että henkilökohtaiset tiedot piilotetaan datan sisällöstä. Esikäsitteilyssä saadaan siis poistettua arkaluontoinen ja henkilökohtainen sisältö ennen pilveen lähettämistä (Satyanarayanan ym. 2015). Myös varastoinnissa on tärkeä suojella käyttäjän yksityisyyttä. Reunalaskennassa dataa voidaan varastoida pilven lisäksi käyttäjän laitteelle, jolloin käyttäjä kykenee hallitsemaan dataansa paremmin (Shi ym. 2016). Kontrolloimalla omaa dataansa käyttäjä kykenee tekemään päätöksen esimerkiksi siitä, saavatko palveluntarjoajat käyttää käyttäjän dataa.

Vaikka reunalaskenta kykenee turvaamaan käyttäjän yksityisyyden hyvin, reunalaskennan tietoturva on vielä tässä vaiheessa turhan heikkoa. Monet reunalaskennan tietoturvaasteista johtuvat siitä, miten reunalaskentaa käyttävät järjestelmät ovat suunniteltu. Järjestelmien suunnittelijat keskittyvät järjestelmän yleiseen hyödyllisyyteen ja käyttäjäkokemukseen, ja työستävät parempaa tietoturvaa vain tarvittaessa (Xiao ym. 2019).

Tietoturvaasteet voivat lisäksi johtua hajautetusta arkkitehtuurimallista sekä käytössä olevien resurssien heikkoudesta. Pilvilaskenta kykenee resurssiensa avulla vahvistamaan fyysisistä kyberturvallisuutta datakeskuksissa ja lisäksi se kykenee käyttämään vahvoja tietoturvajärjestelmiä. Reunalaskennalla taas on käytössä joustavia mutta kevyitä turvallisuusjärjestelmiä, joiden tehtävänä on hyökkäysten ja riskitilanteiden selvittäminen mahdollisimman vähässä ajassa (Mansouri ja Babar 2021). Turvallisuusjärjestelmän kevyiden ja ongelmien nopean selvityksen seurauksena järjestelmä ei kuitenkaan heti kykene täydellisesti korjaamaan ongelmaa, vaan se pyrkii korjaamaan ongelman siihen pisteeseen asti, että se ei enää aiheuta kummempin vahinkoa. Näin ollen reunalaskenta on yleisesti herkempi olemassa-

leville hyökkäyksille verrattuna pilvilaskentaan (Xiao ym. 2019). Mobiililaitteilta voidaan esimerkiksi kerätä informaatiota käyttäjistä seuraamalla reunal palvelimien välistä kommunikaatiota tai varastamalla laitteista antureiden tuottamaa dataa.

5.7 Palvelun laatu ja käyttäjäkokemus

Kuten tietoturva käsiteltäessä todettiin, reunalaskentaa käyttävissä järjestelmissä käyttäjäkokemus on yksi asia, johon pyritään panostamaan. Palvelun laatu on siis reunalaskennassa usein hyvä. Nopeiden palveluiden tarjoaminen on suurin tekijä, jolla käyttäjäkokemusta saadaan reunalaskennassa parannettua. Koska reunalaskennalla voidaan nopeuttaa sovelluksen ja laitteen toimintaa, käyttäjät saavat nopeammin palveluita käyttöön. Minimoimalla viivettä saadaan siis vaikutettua positiivisesti käyttäjien kokemuksen laatuun (Ahmed ym. 2017).

Reunalaskennan hajautettu arkkitehtuuri on myös käyttäjäkokemusta parantava tekijä. Reunal palvelinkerroksella sijaitsevien tukiasemien hyödyntäminen välimuistina voi vaikuttaa kustannuksien lisäksi positiivisesti myös käyttäjäkokemukseen. Tallentamalla mobiililaitteella avattu websisältö tukiasemalle välimuistiin saadaan vähennettyä web-palvelimella esiintyvää viivettä, joka vaikuttaa positiivisesti palvelun laatuun (Ahmed ja Rehmani 2017). Myös palvelimien sijainnilla on merkitystä käyttäjäkokemuksen näkökulmasta. Koska palvelimet sijaitsevat lähellä käyttäjiä, palvelimet pystyvät selvittämään käyttäjistä erilaisia tietoja, kuten esimerkiksi käyttäjän sijainnin (Ahmed ym. 2017). Palveluntarjoaja voi hyödyntää näitä käyttäjistä kerättyjä tietoja ja tarjota niiden avulla käyttäjälle parempia palveluita.

Videopelien on tärkeä kyetä tarjoamaan pelaajille hyvä käyttäjäkokemus, jotta peli voisi menestyä. Erityisesti moninpeleissä viiveellä on suuri vaikutus pelin laatuun. Käyttäjän valitseman toiminnon suorittamisessa esiintyvä viive voi tuottaa käyttäjälle negatiivisia tuntemuksia (Jarschel ym. 2011). Mobiilipelien yhteydessä reunalaskenta on pätevä valinta, sillä reunalaskennalla kyetään vähentämään peleissä esiintyvää viivettä (Prensankar, Di Francesco ja Taleb 2018). Viiveen vähentämisen lisäksi reunalaskennan avulla saadaan parannettua pelin pyörittämisen nopeutta, eli pelin kuvataajuutta (eng. frame rate) voidaan parantaa (Prensankar, Di Francesco ja Taleb 2018). Kuvataajuudella tarkoitetaan sitä, kuinka monta kuvaa näytöllä voidaan esittää sekunnissa.

6 Tulevaisuuden kehityskohteet

Vaikka reunalaskenta kykenee ratkaisemaan monia mobiililaitteilla kerätyn reaaliaikaisen massadatan käsittelyn haasteita, on sillä myös kehitystä vaativia kohteita. Tässä luvussa tuodaan vielä esille ne asiat, joita reunalaskennassa voitaisiin kehittää.

Käytössä olevat resurssit vaikuttavat kaikkiin käytössä olevien laitteiden ja palvelimien toimintaan. Vaikka reunalaskenta kykenee hajautuneen arkkitehtuurinsa avulla mukautumaan intensiivisiin työtehtäviin, on tärkeä huomioida myös hajauttamisen rajat. Massadatan määrä voi tulevaisuudessa kasvaa niin suureksi, että työtehtävien jakaminen osiin ei mahdollisesti riitä enää vastaukseksi massadatan käsittelylle. Jos laskentatehtävän työmäärä on suurempi kuin mitä kaikki käytössä olevat reunaluonnetimet kykenevät suorittamaan, on hyödynnettävä pilvipalvelimia. Datan käsittelyn suorittaminen pilvipalvelimella taas lisää sitä viivettä, joka tulisi kyetä välttämään reaaliaikaisissa sovelluksissa. Tulevaisuutta ajatellen täytyisi siis keksiä ratkaisuja, joilla voitaisiin lisätä reunaluonnetimien ja -palvelimien käytössä olevia resursseja. Lisäämällä resursseja alustat voisivat skaalautua paremmin kasvavaan datamäärään ja reaaliaikaisen massadatan käsittelyn laatua saataisiin parannettua.

Reunalaskennalla itsellään ei ollut vastausta epäyhtenäisen datan käsittelylle. Lisäksi reunalaskennalla voi olla vaikeuksia pätevyuden takaamisessa datan käsittelyssä. Reunalaskennalle voidaan kuitenkin kehittää hajautettuja algoritmeja, jotka huomioivat epäyhtenäisen datan ja takaavat datan käsittelyn pätevyuden. Kun näitä algoritmeja kehitetään lisää, olisi hyvä huomioida myös niiden skaalautuvuus. Hajautettujen algoritmien olisi hyvä pyrkiä olemaan skaalautuvia, jotta datamäärien kasvaessa niiden suoritusaika ei nousisi liian korkeaksi.

Artikkeleita ja tutkimuksia läpikäymällä selvitettiin, reunalaskennan tietoturva tarvitsee vielä paljon parannusta. Tulevaisuutta ajatellen olisi siis hyvä kehittää tietoturvajärjestelmiä, jotka järjestelmän hyödyllisyyden ja käyttäjäkokemuksen priorisoinnin lisäksi keskittyisivät enemmän myös käyttäjien turvallisuuden takaamiseen. Parempien turvallisuusjärjestelmien toiminnan takaamiseksi alustoille tulisi tarjota enemmän resursseja käyttöön, sillä nykyisillä resursseilla alustat eivät kykene ylläpitämään tarpeeksi vahvoja tietoturvajärjestelmiä.

7 Yhteenveto

Tässä tutkielmassa toteutettiin kirjallisuuskatsaus, jossa selvitettiin mobiililaitteilla kerätyn reaaliaikaisen massadatan haasteita ja näiden haasteiden ratkaisemista reunalaskennalla. Lisäksi tutkittiin, kuinka yhteensopivia reunalaskenta ja mobiililaitteet ovat keskenään.

Ensimmäiseksi perehdyttiin tutkielman teoriaosuuteen. Tutkielman toisessa luvussa käsiteltiin reunalaskentaa yleisellä tasolla ja esiteltiin reunalaskennassa käytettävää arkkitehtuurimallia. Reunalaskennan arkkitehtuurissa oleva reunapalvelinkerros kykenee tarjoamaan laskentaparadigmalle ominaisuuksia, joita pilvilaskenta ei tue yhtä tehokkaasti. Kolmannessa luvussa tuotiin esille reaaliaikaisen käsittelyn haasteita sekä huomiota vaativia ominaisuuksia, ja neljännessä luvussa käsiteltiin mobiililaitteiden tyypillisimpiä ominaisuuksia. Reunalaskennan tulisi kyetä ratkaisemaan esiteltyjä datan käsittelyn haasteita sekä tukemaan esiteltyjä datan käsittelyn ja mobiililaitteiden ominaispiirteitä.

Varsinainen tutkimus suoritettiin viidennessä luvussa. Ensimmäiseksi verrattiin reunalaskennan tarjoamia hyötyjä mobiililaitteiden ominaisuuksiin, ja voitiin huomata, että mobiililaitteet hyötyvät reunalaskennan tarjoamasta liikkuvuuden tuesta, kyvystä sopeutua pienikokoisille laitteille sekä verkon vähäisestä rasituksesta. Seuraavaksi perehdyttiin tarkemmin datan käsittelyn haasteiden ratkaisemiseen. Reunalaskenta mahdollistaa reaaliaikaisen ja energiatehokkaan datan käsittelyn. Reunalaskenta kykenee myös skaalautumaan erikokoisille alueille maantieteellisesti, vähentää osan kustannuksista sekä tarjoaa käyttäjälle yksityisyyttä ja hyvän palvelun laadun.

Lopuksi tuotiin vielä esille tulevaisuuden kehityskohteita. Rajallisten resurssien takia reunalaskennalla voi tulevaisuudessa olla vaikeuksia skaalautumisessa ja laskentatehtävien toteuttamisessa, jos datamäärät kasvavat liian suuriksi. Lisäksi hajautuneisuuden ja datan luonteen takia reunalaskenta tarvitsee päteviä algoritmeja datan käsittelyn parantamiseksi. Suurin kehityskohde on kuitenkin reunalaskennan tietoturva. Tietoturvajärjestelmät ovat kevyitä sekä priorisoivat hyödyllisyyttä ja käyttäjäkokemusta. Reunalaskennalle täytyy siis kehittää vahvempia turvallisuusjärjestelmiä, jotta käyttäjä voi turvallisesti jatkaa laitteensa käyttämistä.

Lähteet

- Ahmed, A., ja E. Ahmed. 2016. “A survey on mobile edge computing”. Teoksessa *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, 1–8. <https://doi.org/10.1109/ISCO.2016.7727082>.
- Ahmed, E., A. Ahmed, I. Yaqoob, J. Shuja, A. Gani, M. Imran ja M. Shoaib. 2017. “Bringing Computation Closer toward the User Network: Is Edge Computing the Solution?” *IEEE Communications Magazine* 55 (11): 138–144. <https://doi.org/10.1109/MCOM.2017.1700120>.
- Ahmed, Ejaz, ja Mubashir Husain Rehmani. 2017. “Mobile Edge Computing: Opportunities, solutions, and challenges”. *Future Generation Computer Systems* 70:59–63. ISSN: 0167-739X. <https://doi.org/https://doi.org/10.1016/j.future.2016.09.015>. <https://www.sciencedirect.com/science/article/pii/S0167739X16303260>.
- Cao, Jie, Quan Zhang ja Weisong Shi. 2018. *Edge Computing: A Primer*. Springer Briefs in Computer Science. Springer. ISBN: 978-3-030-02082-8. <https://doi.org/10.1007/978-3-030-02083-5>. <https://doi.org/10.1007/978-3-030-02083-5>.
- Cisco. 2020. *Cisco Annual Internet Report, 2018–2023*. <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>.
- Coulouris, George, Jean Dollimore, Tim Kindberg ja Gordon Blair. 2011. *Distributed Systems: Concepts and Design*. 5th. USA: Addison-Wesley Publishing Company. ISBN: 0132143011.
- Dai, Hong-Ning, Raymond Chi-Wing Wong, Hao Wang, Zibin Zheng ja Athanasios V. Vasilakos. 2019. “Big Data Analytics for Large-Scale Wireless Networks: Challenges and Opportunities”. *ACM Comput. Surv.* (New York, NY, USA) 52, numero 5 (syyskuu). ISSN: 0360-0300. <https://doi.org/10.1145/3337065>. <https://doi.org/10.1145/3337065>.
- Dev Mishra, A., ja Y. Beer Singh. 2016. “Big data analytics for security and privacy challenges”. Teoksessa *2016 International Conference on Computing, Communication and Automation (ICCCA)*, 50–53. <https://doi.org/10.1109/CCAA.2016.7813688>.

- Guerra, J., H. Pucha, Joseph S. Glider, W. Belluomini ja R. Rangaswami. 2011. “Cost Effective Storage using Extent Based Dynamic Tiering”. Teoksessa *FAST*.
- Ha, Kiryong, Zhuo Chen, Wenlu Hu, Wolfgang Richter, Padmanabhan Pillai ja Mahadev Sathanarayanan. 2014. “Towards wearable cognitive assistance”. *MobiSys 2014 - Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services* (kesäkuu). <https://doi.org/10.1145/2594368.2594383>.
- Hariri, R. H., Erik M. Fredericks ja Kate M. Bowers. 2019. “Uncertainty in big data analytics: survey, opportunities, and challenges”. *Journal of Big Data* 6:1–16.
- Hashem, Ibrahim Abaker Targio, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani ja Samee Ullah Khan. 2015. “The rise of “big data” on cloud computing: Review and open research issues”. *Information Systems* 47:98–115. ISSN: 0306-4379. <https://doi.org/https://doi.org/10.1016/j.is.2014.07.006>. <https://www.sciencedirect.com/science/article/pii/S0306437914001288>.
- Jarschel, M., D. Schlosser, S. Scheuring ja T. Hofffeld. 2011. “An Evaluation of QoE in Cloud Gaming Based on Subjective Tests”. Teoksessa *2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 330–335. <https://doi.org/10.1109/IMIS.2011.92>.
- Liu, D., Z. Yan, W. Ding ja M. Atiquzzaman. 2019. “A Survey on Secure Data Analytics in Edge Computing”. *IEEE Internet of Things Journal* 6 (3): 4946–4967. <https://doi.org/10.1109/JIOT.2019.2897619>.
- Mansouri, Yaser, ja M. Ali Babar. 2021. “A review of edge computing: Features and resource virtualization”. *Journal of Parallel and Distributed Computing* 150:155–183. ISSN: 0743-7315. <https://doi.org/https://doi.org/10.1016/j.jpdc.2020.12.015>. <https://www.sciencedirect.com/science/article/pii/S0743731520304317>.
- Marjani, M., F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa ja I. Yaqoob. 2017. “Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges”. *IEEE Access* 5:5247–5261. <https://doi.org/10.1109/ACCESS.2017.2689040>.

- McAfee, Andrew, ja Erik Brynjolfsson. 2012. “Big Data: The Management Revolution”. *Harvard business review* 90 (lokakuu): 60–6, 68, 128.
- Mohamed, N., ja J. Al-Jaroodi. 2014. “Real-time big data analytics: Applications and challenges”. Teoksessa *2014 International Conference on High Performance Computing Simulation (HPCS)*, 305–310. <https://doi.org/10.1109/HPCSim.2014.6903700>.
- Premsankar, G., M. Di Francesco ja T. Taleb. 2018. “Edge Computing for the Internet of Things: A Case Study”. *IEEE Internet of Things Journal* 5 (2): 1275–1284. <https://doi.org/10.1109/JIOT.2018.2805263>.
- Satyanarayanan, M. 2017. “The Emergence of Edge Computing”. *Computer* 50 (1): 30–39. <https://doi.org/10.1109/MC.2017.9>.
- Satyanarayanan, M., P. Bahl, R. Caceres ja N. Davies. 2009. “The Case for VM-Based Cloudlets in Mobile Computing”. *IEEE Pervasive Computing* 8 (4): 14–23. <https://doi.org/10.1109/MPRV.2009.82>.
- Satyanarayanan, M., P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, W. Hu ja B. Amos. 2015. “Edge Analytics in the Internet of Things”. *IEEE Pervasive Computing* 14 (2): 24–31. <https://doi.org/10.1109/MPRV.2015.32>.
- Shi, W., J. Cao, Q. Zhang, Y. Li ja L. Xu. 2016. “Edge Computing: Vision and Challenges”. *IEEE Internet of Things Journal* 3 (5): 637–646. <https://doi.org/10.1109/JIOT.2016.2579198>.
- Xia, Feng, Laurence T Yang, Lizhe Wang ja Alexey Vinel. 2012. “Internet of things”. *International journal of communication systems* 25 (9): 1101.
- Xiao, Y., Y. Jia, C. Liu, X. Cheng, J. Yu ja W. Lv. 2019. “Edge Computing Security: State of the Art and Challenges”. *Proceedings of the IEEE* 107 (8): 1608–1631. <https://doi.org/10.1109/JPROC.2019.2918437>.