

Joonas Juuso

**SUORITUSKYKYMITTAUSTEN VISUALISOINTI TUKIASEMIEN RF-
MODUULEISSA**

SUORITUSKYKYMITTAUSTEN VISUALISOINTI TUKIASEMIEN RF- MODUULEISSA

Joonas Juuso

Opinnäytetyö

Syky 2022

Tietotekniikan tutkinto-ohjelma

Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma, ohjelmistokehityksen suuntautumisvaihtoehto

Tekijä: Joonas Juuso
Opinnäytetyön nimi: Suorituskykymittausten visualisointi tukiasemien RF-moduuleissa
Työn ohjaaja: Janne Kumpuoja
Työn valmistumislukukausi ja -vuosi: Syksy 2022
Sivumäärä: 35 + 5 liitettä

Opinnäytetyössä tutkittiin langattomia radioverkkoteknologioita, suorituskykymittauksia sekä visualisointia. Työn toimeksiantajalla eli Nokiella oli tarve visualisoida langattomassa mobiiliverkossa käytettävän radio-osan antennin ohjelmiston suorituskykymittauksia.

Työn tavoitteena on tutustua tarkemmin työkaluihin ja kehysympäristöihin, joilla voidaan suorittaa suorituskykymittauksia ja rakentaa visualisointeja. Pääpaino kuitenkin on työn käyttötapauksessa, jossa tavoitteena on lähettää EM-jäljityksestä saadut suorituskykymittaukset valittuun visualisointiympäristöön ja rakentaa kojelautaa, josta nähdään kaikki tärkeät aspektit datasta.

Opinnäytetyö rakentuu langattomien mobiiliverkkojen teorian ja työkalujen sekä kehysympäristöjen dokumentaation ympärille. Käyttötapaus tehdään suoraan toimeksiantajan käyttöön, käyttäen toimeksiantajan tarjoamia työkaluja.

Käyttötapauksen rakentaminen ja analyysi sujui odotettavasti. Tiedettiin, että visualisointi auttaa näkemään tarkemmin trendit ja ongelmakohtat datasta, ja rakentamalla kojelaudan suorituskykymittauksesta saatua dataa varten, löytyi ongelmia RTC-viestien lähetyksessä. Mitatusta kehysympäristöstä lähetettyjen viestin allokatiossa oli virhe, joka selvisi hajontakaaviosta, kun prosessoitujen viestien välille oli ilmestynyt viivettä.

Työn tuloksena ymmärrys mahdollisuuksista datan kanssa työskennellessä kasvoi; on olemassa työkaluja, joilla voidaan mitata laitteistojen suorituskykyä eri tasoilla, ja kehysympäristöjä, joilla saatua mittaustuloksia pystytään visualisoimaan.

Työ myös antaa tarkemman kuvan itse mittauksen kohteesta. Työssä kuitenkin olisi voinut käyttötapauksen kohdalla ottaa huomioon datamäärät ja tiedostotyyppiin, sillä datan lähetys aiheutti huomattavaa viivettä.

Asiasanat: visualisointi, vertailuanalyysi, suorituskykyanalyysi, radiotekniikka, matkaviestinverkot

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology, Option of Software Development

Author: Joonas Juuso
Title of thesis: Visualizing benchmarking results in the base station RF module
Supervisor: Janne Kumpuoja
Term and year when the thesis was submitted: Autumn 2022
Number of pages: 35 + 5 appendices

Mobile networks have advanced rapidly in the past centuries, which has increased the data rate greatly. This thesis covers the topic of the need for benchmarking and visualization of the benchmarking results from the RF module in the base station.

The main point of the thesis is to gain competence in the field of benchmarking and visualization in the RF side and create visualized results of benchmarking results for Nokia.

The thesis covers the basics of wireless communications, starting from 4G and going through to fifth generation new radio. The basics of benchmarking and visualization are also covered with the tools and frameworks that help with the topics.

A use case is introduced in the later part, where we go through the workflow for the visualization of one specific hardware benchmarking test results. The benchmarking results are from tracing the EM events, getting the execution times and event delays, to determine any faults with our implementation when sending RTC messages. The aim of the use case was to create a dashboard to visualize these results via Splunk and our test automation.

The results were as expected, as the data being handled is large, the trends and possible faults may not be visible by eye. When visualizing the EM trace results, we found that there was a considerable delay in the RTC message processing.

All in all, the thesis covers the fundamental benefit of using visualization in our work environment. Without it, we might have not found out the issues that were within our benchmarking results, and by automating the sending of results, we can keep on fixing any issues in real time.

Keywords: benchmarking, visualization, mobile communication networks, radio technology

SISÄLLYS

1	JOHDANTO	11
2	LANGATTOMAT RADIOVERKKOTEKNOLOGIAT	12
2.1	4G	12
2.1.2	LTE:n korkean tason arkkitehtuuri	12
2.1.3	Tärkeimmät 4G LTE:n suorituskyvyn mahdollistavat tekniset ominaisuudet radioyhteydessä	13
2.2	5G New Radio	14
2.2.1	LTE:n ja NR:n (Release 15) tärkeimmät eroavaisuudet	14
2.2.2	Keilanmuodostus ja MIMO	15
2.2.3	CPRI vs eCPRI	17
2.2.4	Fyysisen tason parannukset	18
3	SUORITUSKYKYMITTAUS	20
3.1	Suorituskykymittaus RF L1Low radioyksikössä	21
3.2	Työkalut suorituskyvyn mittaamiseen	21
3.2.1	EM tapahtumien jäljitystyökalu	21
3.2.2	Perf	23
3.2.3	Lauterbach	24
4	DATAN VISUALISOINTI	26
4.1	Erilaiset visualisointitavat	26
4.2	Visualisointialustat	30
5	EM TAPAHTUMIEN SUORITUSKYKYMITTAUS – KÄYTTÖTAPAUS	32
5.1	Käyttötapauksen toteutus	33
5.2	Käyttötapauksen tulokset	38
6	YHTEENVETO	41
	LÄHTEET	43

LIITTEET 47

LYHENTEET JA TERMIT

3GPP	3rd Generation Partnership Project. Usean standardointijärjestön yhteistyöorganisaatio.
3G	Kolmannen sukupolven radioverkkoteknologia.
4G	Neljännän sukupolven radioverkkoteknologia.
5G	Viidennen sukupolven radioverkkoteknologia.
5G NR	5G New Radio. 5G-standardi.
AI	Artificial intelligence. Tekoäly.
BBU	Baseband unit. Kantataajuusyksikkö. Prosessoi kantataajuuden ja vastaa kommunikaatiosta fyysisellä tasolla
Bin	Kuvaa datajoukkoa.
CA	Carrier aggregation. Kantoaaltojen yhdistäminen.
CI	Continuous Integration. Jatkuva integraatio. Esimerkiksi ohjelmistoon tehtävien muutosten integrointi automatisointi.
CPRI	Common Public Radio Interface. Protokolla tukiaseman ja antennin välisissä yhteyksissä 4G:ssä ja aikaisessa 5G:ssä
CPU	Central processing unit. Keskusyksikkö.
CSI	Channel state information. Kanavatilainformaatio.
CSV	Comma-separated values. Data on tiedostossa pilkulla erotettuna.
DFTS-OFDM	Discrete Fourier transform – spread-OFDM. Siirtotekniikka, jolla voidaan esimerkiksi suorittaa matalakompleksista tasausta taajuusalueella.
DL	Downlink. Siirtoyhteys verkosta laitteeseen

ECPRI	Enhanced common public radio interface. NR-mobiiliverkkoihin suunniteltu uusi datapakettipohjainen protokolla tukiaseman ja antennin välille.
EM	Event Machine. Kehäympäristö kuormantasaukselle.
eMBB	Enhanced Mobile Broadband. Paranneltu mobiililaajakaista.
eNB	Evolved Node B. LTE:n tukiasemakomponentti.
EO	Event object. Suorituksen kohde EM:ssä.
EPC	Evolved Packet Core. Kommunikaatio pakettiverkkojen välillä.
E-UTRAN	Evolved UMTS Terrestrial Radio Access Network. LTE-tekniikan radioliityntäverkko.
FDD	Frequency-division duplex. Taajuusjakoinen dupleksi. Eritajuisen kantaallon käyttö lähetys- ja vastaanottokanavilla.
gNB	Next Generation NodeB. 5G:n tukiasemakomponentti.
HEC	HTTP Event Collector. Mahdollistaa datan lähetyksen Splunk kehäympäristöön HTTP ja HTTPS protokollien kautta.
HTTP	Hypertext Transfer Protocol. Tiedonsiirtoprotokolla, jota selaimet ja palvelimet käyttävät.
IA	Initial Access. Alustava yhteys käyttäjän ja verkon välillä.
IIoT	Industrial Internet of Things. Teollisuudessa käytettävät IoT laitteet.
JTAG	Joint Test Action Group.
Keilanmuodostus	Beamforming. Usean antennin järjestelmässä signaalin vaiheen ja mahdollisesti amplitudin muutos jokaisella antennielementillä, joka mahdollistaa signaalin kohdistamisen.
L1	Layer 1. Fyysinen taso OSI-mallissa.

LTE	Long-term evolution. Käytetään 4G:n yhteydessä kuvattaessa tietynlaista 4G:n yhteysmuotoa.
Matkapuhelinverkko	Tarkoittaa langatonta verkkoa, johon on yhdistynyt monta tilaajaa tai käyttäjää, esimerkiksi matkapuhelimia tai äylaitteita. Laitteet yhdistävät tukiasemaan yhteyden lähettäessään tai vastaanottaessaan signaalin.
MIMO	Multiple-input multiple-output. Usean antennin käyttö lähettimessä ja vastaanottimessa.
ML	Machine learning. Koneoppiminen.
MLTK	Machine Learning Toolkit. Splunkin tarjoama koneoppimiskirjasto.
OFDM	Orthogonal frequency-division multiplexing. Tiedonsiirto yhtä aikaa useilla häiritsemättömillä taajuuskanavilla.
RF	Radio frequency. Radiotaajuus.
RTC	Real-Time Control. ECPRI:n reaaliaikaista hallintatason tietoa.
SCT	System Component Test. Järjestelmäkomponenttitestit.
SINR	Signal to Interference & Noise Ratio. Laadun mittaustyyppi langattomissa verkoissa.
SPL	Search Processing Language. Splunk kehysympäristön hakukieli.
TDD	Time-division duplex. Aikajakoinen dupleksi. Samassa siirtokanavassa eri signaalien siirto ajallisesti lomitettuna.
Tukiasema	Käytetään langattomassa verkkoliikenteessä laitteen yhdistämisessä radioteitse kiinteään verkkoon
UE	User Equipment. Matkaviestinlaite. Telepäätelelaite matkaviestintään.
UL	Uplink. Siirtoyhteys laitteesta verkkoon.
UMTS	Universal Mobile Telecommunication System. Kolmannen sukupolven matkaviestinjärjestelmä.

URLLC	Ultra Reliable Low Latency Communication. Lähes viiveetön yhteys.
V2X	Vehicle-to-everything. Verkossa käyttäjinä toimii esimerkiksi ajoneuvot.

1 JOHDANTO

Viidennen sukupolven matkapuhelinverkkojen myötä tapahtuu paljon muutoksia laitteisto- ja ohjelmistopuolella. Laitteistojen ja ohjelmistojen arkkitehtuuri muuttuu entistäkin kompleksisemmaksi, jotta saadaan lähetettyä entistäkin enemmän ja nopeammin dataa verkon yli. Esimerkiksi radion fyysisellä tasolla (L1) enemmän toiminnallisuutta siirretään radion antenniosaan.

Opinnäytetyön tarkoitus on perehtyä visualisoinnin hyötyihin esiteltäessä suorituskykymittaustuloksia mobiiliverkon radio-osasta. Suorituskykymittaus on erittäin tärkeää, sillä se auttaa selvittämään mahdolliset suorituskykyongelmat sekä ohjelmistossa että laitteistossa. Suorituskyvyn mittaamiseen on kuitenkin olemassa monia hyviä työkaluja, joita esitellään myöhemmissä luvuissa.

Työssä selvitetään visualisoinnin tuomat mahdollisuudet. Visualisoimalla tulokset pystytään selvittämään isosta määrästä dataa esimerkiksi trendejä tai yksittäisiä piikkejä, joiden avulla voidaan lähteä avaamaan tiettyä ongelmakohtaa tarkemmin. Tämän lisäksi työssä käydään läpi yrityksiltä erilaisia pinoratkaisuja, jotka tuovat datan visualisointiin helpotusta sekä nopeutta.

Opinnäytetyössä esitellään yhden käyttötapauksen suorituskykymittauksia ja niiden visualisointia. Käyttötapaus perustuu opinnäytetyön toimeksiantajan, Nokian, tarpeeseen saada tietoa esitettyä radio-osalla ajettujen testien aikana prosessoiduista tapahtumista. Tapahtumien data tulee saada automatisoidusti visualisoitua valitussa kehysympäristössä. Tärkeää on saada tietää etenkin tapahtumien ajoaika, ydinallokaatio sekä tapahtumien välinen viive.

Opinnäytetyö ja käyttötapaus tulevat suoraan käyttöön Nokialle. Automatisoimalla suorituskykymittaukset ja visualisoimalla tulokset, parannamme tuotteiden laatua, sillä ongelmat huomataan, ennen kuin laitteet pääsevät kuluttajien haltuun.

2 LANGATTOMAT RADIOVERKKOTEKNOLOGIAT

Langattomat radioteknologiat ovat olleet jo pitkään käytössä, jo yli sata vuotta sitten pystyttiin lähettämään radioaaltoja Atlantin yli. Tässä opinnäytetyön teoriaosassa kuitenkin keskitytään matkapuhelinverkkoon sekä sen edistykseen viimeisen viidentoista vuoden aikana ja siihen, miten kehitys on vaikuttanut datan lähetyksnopeuksiin sekä muutoksiin verkkojen arkkitehtuurissa. (1, s. 2.)

2.1 4G

Vuonna 2004 3GPP aloitti tutkimuksen UMTS:n pitkäaikaiseen evoluutioon, jonka tarkoitus oli kehittää radiojärjestelmää vastaamaan tulevaisuudessa käyttäjien korkeampiin datanlähetyksmääriin sekä tarjoamaan yhteyden entistäkin pienemmällä latenssilla. Tätä kehitystä vauhdittaa älypuhelinien kasvu suosiossa ja sitä myötä esiin nousevien applikaatioiden räjähdysmäinen kasvu. 4G LTE poistaa 3G:n latenssin, joka on noin 100 millisekuntia. Tällainen viive datapakettien lähetyksessä verkkoelementtien ja ilman läpi vaikeutti esimerkiksi moninpelaamista eli pelaamista reaaliaikaisesti muiden käyttäjien kanssa. (1, s. 9, 11 & 12.)

2.1.1 LTE:n korkean tason arkkitehtuuri

LTE sisältää kolme pääkomponenttia: matkaviestinlaite (UE), LTE-tekniikan radioliityntäverkko nimeltä *Evolved UMTS Terrestrial Radio Access Network* (E-UTRAN) ja *Evolved Packet Core* (EPC). Matkaviestinlaitteella tarkoitetaan lyhyesti laitetta, johon sisältyy SIM-kortti. (2, s. 21.)

E-UTRAN prosessoi kommunikation viestintälaitteen ja EPC:n välillä, ja siihen yksi komponentti nimeltä *evolved Node B* (eNB). Jokainen eNB on tukiasema, joka hallinnoi viestintälaitteita yhdellä tai useammalla solulla. Yksi viestintälaitte voi kuitenkin olla yhteyksissä samaan aikaan vain yhteen tukiasemaan ja soluun. Solulla tarkoitetaan lähetin-vastaanottimen peittoaluetta. (2, s. 24.)

EPC kommunikoi pakettidataverkkojen, kuten Internetin tai yksityisten yritysverkkojen kanssa (2, s. 21).

2.1.2 Tärkeimmät 4G LTE:n suorituskyvyn mahdollistavat tekniset ominaisuudet radioyhteydessä

4G LTE on aiempia sukupolvia kyvykkäämpi radiojärjestelmä, joka johtuu useista eri kehitysaskelista, kuten muutoksista datansiirtojärjestelmässä, kanavajaon käyttöönotosta, siirto useaa antennia hyödyntäen, joustavuus taajuuksissa ja kaistanleveydessä.

OFDM:n käyttöönotto datansiirtojärjestelmässä mahdollistaa erittäin vapaan kanavan taajuuden valintatarkkuuden. Modulaation käyttöönottoon on vaikuttanut myös muut hyödynnetyt tekniikat 4G LTE:ssä, kuten suuren kaistanleveyden käyttö sekä tuki datansiirrolle useammalla antennilla. Tätä hyödynnetään myös siirtoyhteydessä laitteesta verkkoon (UL) DFTS-OFDM-tekniikan avulla. (2, s. 28.)

LTE-siirtotekniikan ytimessä on kanavan jaettu käyttö useamman käyttäjän kanssa, joka on mahdollista hyödyntämällä ”järjestelijää” (scheduler). Järjestelijä päättää, kenelle sen kannattaa sijoittaa resursseja siirtovaiheessa sekä millaisella nopeudella. OFDM:n käyttö DL- ja UL siirtotavoissa tarjoaa järjestelijälle käyttöön aika- ja taajuusalueen hyödyntämisen, jolloin voidaan valita käyttäjä parhaiden kanavaolosuhteiden perusteella. Kanavan jako useammalle käyttäjälle määrittää hyvin pitkälti radiojärjestelmän suorituskyvyn etenkin kovan rasituksen alla. (2, s. 28.)

Useamman antennin hyödyntäminen datansiirrossa on yksi tärkeimmistä edistysaskelista radiojärjestelmien evoluutiossa, sillä se mahdollistaa LTE:n koviin suorituskykytavoitteisiin pääsemisen. Usean antennin järjestelmää hyödynnetään tilatason multipleksoinnissa (spatial multiplexing), jolla tarkoitetaan siirron bittijonon puolittamista moniksi rinnakkaisiksi lähetyksiksi. Nämä jonot lähetetään samanaikaisesti eri antennista samalla taajuusalueella. (3, s. 28.) Tilatason multipleksointia myös hyödynnetään MIMO (multiple-input multiple-output) -järjestelmissä, jolla tarkoitetaan useamman antennin käyttöä lähettimessä sekä vastaanottimessa. Useamman antennin käyttö mahdollistaa myös keilanmuodostuksen, jolla voidaan parantaa SINR:ta sekä lopulta koko järjestelmän signaalivoimakkuutta. LTE Release 9 toi mukanaan myös kaksitasoisen keilanmuodostuksen. (4, s. 96–100 ja 103.)

LTE-radiotekniikassa pystytään taajuusjoustavuuden takia käyttämään TDD- ja FDD-asetelmaa, eli dataa voidaan siirtää erotetuilla taajuusalueilla sekä eri ajassa. Tätä on 3GPP-standardin mukaan radioyhteystekniikassa hyödynnetty jo ennen LTE:tä, mutta eri radiotekniikoita hyödyntäen, joka esimerkiksi monimutkaistaa kaksisuuntaisten terminaalien kehitystä sekä toteutusta. Joustavuus kaistanleveydessä, eli mahdollisuus käyttää eri datansiirtokaistoja sekä

UL- että DL suunnissa, on LTE:n toiminnan kannalta kriittistä. LTE:n toimintaan saatavilla olevien radiotaajuusalueiden määrä voi vaihdella huomattavasti taajuuskaistojen välillä sekä operaattorin tietystä tilanteesta johtuen. LTE:n spesifikaatio sallii siirtokaistan noin 1 MHz:stä suunnilleen 20 MHz:iin asti. (4, s. 100–101.)

Aiemmin mainittujen ominaisuuksien lisäksi LTE Release 10, toisella nimellä tunnettu LTE Advanced, tuo kantoaaltojen yhdistämisen (CA) radiojärjestelmään. Kantoaaltoja yhdistämällä siirtokaistanleveys pystytään kasvattamaan jopa 100 megahertsiin asti. (4, s. 104.)

2.2 5G New Radio

Tarve korkeammille datanlähetysmäärille, pienemmälle latenssille ja luotettavuudelle johti keskusteluun viidennen sukupolven mobiiliverkoista jo vuonna 2012 (5, luku 1.2).

Vaikkakin LTE onkin erittäin kykenevä verkkoteknologia, jotkin vaatimukset eivät olleet mahdollisia LTE:ssä ja sen evoluutiossa. Tämän lisäksi LTE:n tekniikkaa on aloitettu kehittämään yli 10 vuotta sitten, joka on yksi mahdollistajista 5G NR:n rakentamisessa, kun pystytään ottamaan käyttöön kehittyneempiä teknologisia ratkaisuja. Kehittyneemmät teknologiset ratkaisut parantavat muun muassa lähetysnopeutta ja latenssia. (Taulukko 1.) (5, luku 1.2.2.)

TAULUKKO 1. LTE ja 5G vertailussa (5, luku 5.1; 6, luku 1.5).

Radiotekniikka	LTE	5G
Spektri	< 5 GHz	< 52.6 GHz
Siirtonopeus	500 Mbit/s (UL), 1000 Mbit/s (DL)	10 Gbit/s (UL), 20 Gbit/s (DL)
Latenssi	5 ms	0,5 ms

2.2.1 LTE:n ja NR:n (Release 15) tärkeimmät eroavaisuudet

NR käyttää paljon korkeampia taajuusalueita tukeakseen isoja siirtokaistoja ja siihen liittyvää korkeaa datansiirtomäärää. Näillä niin kutsutuilla millimetrialloilla mahdollistetaan korkea datansiirtokapasiteetti ja äärimmäiset siirtonopeudet. (5, luku 5.)

5G NR:n radiojärjestelmä on suunniteltu olemaan mahdollisimman tehokas; poistamalla aina päällä olevat signaalit, kuten esimerkiksi referenssisignaalit kanavan arviointia varten, voidaan

parantaa koko verkon energiatehokkuutta sekä vähentää signaalien interferenssiä. NR:n suunnittelussa on otettu myös huomioon mahdolliset tulevaisuuden teknologiat ja käyttökohteet radiolle rakentamalla radiojärjestelmästä eteenpäin yhteensopivan (forward compatibility). (5, luku 5.)

Myös LTE:stä hieman tuttu keilanmuodostus on 5G NR järjestelmissä yhtenä kulmakivenä. Keilanmuodostusta hyödynnetään datansiirrossa, joka tiettyyn rajaan asti on mahdollista LTE:llä, mutta myös hallintatason (control plane) aliohjelmien kanssa, kuten alustavan yhteyden muodostamisen (IA) käyttäjälaitteen sekä tukiaseman välille. (5, luku 5; 7, s. 1.)

NR kuitenkin toimii LTE:tä mielessä pitäen, sillä tiedonsiirtotekniikat on suunniteltu toimimaan toistensa kanssa. Tähän sisältyy muutakin kuin sujuva siirto tekniikoiden välillä ja yhtäaikainen toimiminen. NR tarjoaa yhtenä ominaisuutena kaksoisyhdistettävyyden (dual-connectivity) LTE:n kanssa, tarkoittaen siis laitteiden samanaikaista yhteyttä sekä NR että LTE verkon kanssa. Taajuusalueen jakaminen molempien tekniikoiden välillä on myös mahdollista ottamalla NR käyttöön samalla taajuusalueella kuin LTE, jolloin kokonaisspektrikapasiteetti voidaan dynaamisesti jakaa molempien radiotekniikoiden välillä. (5, luku 5.9.)

2.2.2 Keilanmuodostus ja MIMO

Keilanmuodostus on yksi 5G NR:n mahdollistavista tekijöistä, NR radiojärjestelmä on rakennettu tukemaan monta liikuteltavaa antennielementtiä sekä datan lähetyksessä että datan vastaanotossa. Keilanmuodostuksella tarkoitetaan usean antennin järjestelmässä signaalin vaiheen ja mahdollisesti amplitudin muutosta jokaisella antennielementillä, jolloin lähettimen puolella lähetettävä tieto pystytään kohdistamaan tiettyyn suuntaan, tässä tapauksessa tiettyyn paikkaan lähettimen kattavuusalueella (5, luvut 11.1 ja 5.9).



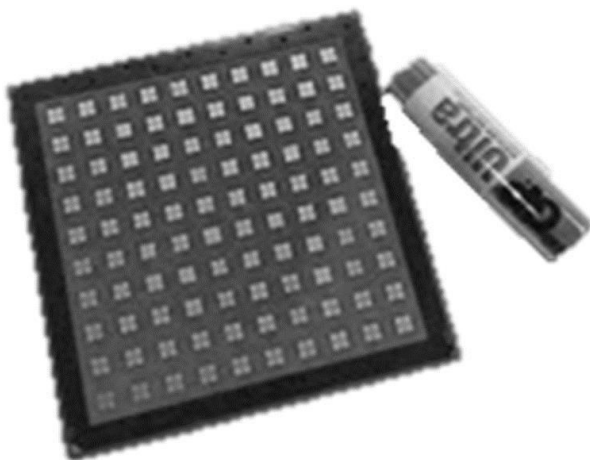
KUVA 1. NR keilanmuodostus (5, kuva 5.5).

NR kanavat sekä signaalit on suunniteltu myös keilanmuodostusta mielessä pitäen. Kanavatilainformaatio (CSI) usean antennin käytölle saadaan palautteena CSI raporteista referenssisignaalien välityksen perusteella UL:n yhteydessä. Kanavatilainformaation voidaan myös hankkia UL mittauksilla hyödyntämällä kanavan vastavuoroisuutta (reciprocity). (5, luku 5.9.)

Keilanmuodostusta voidaan hyödyntää eri tavoin, riippuen käytetäänkö korkeaa tai matalaa taajuutta. Korkeammalla taajuudella keilanmuodostusta voidaan hyödyntää pidentämään signaalin kantavuutta ja alemmilla taajuuksilla keilanmuodostus mahdollistaa täysulotteisen (full dimensional), niin kutsutun massiivisen, MIMO:n. Alemmilla taajuuksilla pystytään myös välttämään signaalihäiriöt tilatason erotuksella (spatial separation). (5, luku 5.9.)

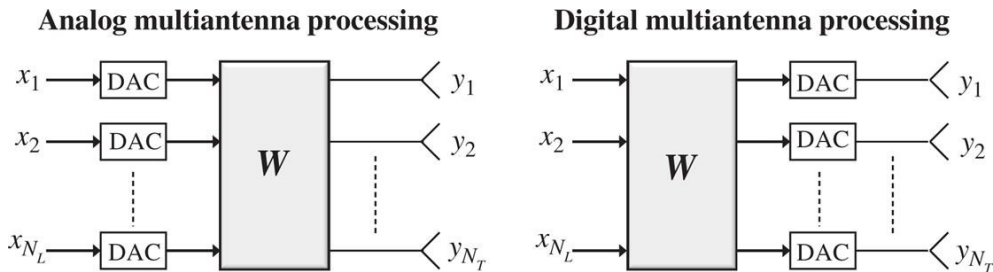
Massiivisen MIMO:n käyttötarkoitus perustuu siihen, että korkeammilla taajuuksilla, niin kutsutuilla millimetrialloilla, signaali vaimentuu rajusti. Etenemisvaimennuksen kompensoimiseksi keilanmuodostusta hyödynnetään gNB:n ja UE:n puolella keilanhallintamenetelmillä usean antennin datansiirrossa. (8, s. 100.)

Mobiiliverkkojärjestelmissä, jossa laitteet ovat eri suunnissa ja liikkuvat satunnaisesti, ei voida käyttää kiinteää korkeasti suunnattua antennia, vaan tarvitaan mahdollisuus suunnata keila aina käyttäjän suuntaan. Tämä saavutetaan käyttämällä monesta antennielementistä rakentuvaa antennipaneelia, jossa väli antennielementtien välillä on suhteutettuna signaalin aallonpituuteen (Kuva 2 antennipaneeli tähtää 28 GHz taajuuskaistalle). Lähettimen keilaa voidaan suunnata säätämällä erikseen jokaisen antennielementin signaalin vaihetta. (5, luku 11.1.)



KUVA 2. 64 kaksoispolarisoidun antennielementin antennipaneeli. Paristo kuvassa havainnollistaa antennipaneelin kokoa. (5, kuva 11.1.)

Myös sillä on väliä, käytetäänkö keilanmuodostuksen prosessoinnissa digitaalista- vai analogista prosessointia.



KUVA 3. Digitaalisen ja analogisen prosessoinnin ero (5, kuva 11.3).

Analoginen prosessointi on digitaalista prosessointia yksinkertaisempaa, sillä digitaalinen prosessointi vaatii yhden digitaali-analogiamuuntimen aina yhtä antennielementtiä kohden, jolloin järjestelmän kompleksisuus sekä virran kulutus kasvaa. Monen lähekkäisen antennielementin käytössä korkeilla taajuualueilla yleisesti käytetään analogista prosessointia, jolloin moniantennilähetyksissä datansiirto on rajoitettu antennikohtaisiin vaihesiirtoihin keilanmuodostuksessa. (5, luku 11.1; 6, luku 6.2.)

Digitaalinen prosessointi kuitenkin mahdollistaa entistäkin kehittyneemmän datansiirron. Sen sijaan että käytettäisiin yksinkertaista analogista vaihesiirtoa, voidaan symbolit prosessoida digitaalisesti hyödyntämällä antennikohtaista amplitudin skaalauskerrointa ja vaihesiirtoa. Tämän monimutkaisemman prosessointitavan hyötynä on joustavuus sekä mahdollisuus vastaanottaa useita kerroksia ja useita keilamuodostettuja signaaleja samanaikaisesti. Hyötynä myös on kompleksisempi keilanmuodostus, jossa antenniryhmässä, johon sisältyy M määrä antenneja, voidaan ohjata niiden nolla-arvoja ($M - 2$). Tällöin voidaan välttää häiriöt DL datansiirrossa muissa vastaanottimissa sekä UL datansiirrossa muilta vastaanottimilta. (5, luku 11.1; 6, luku 6.2.)

2.2.2 CPRI ja eCPRI

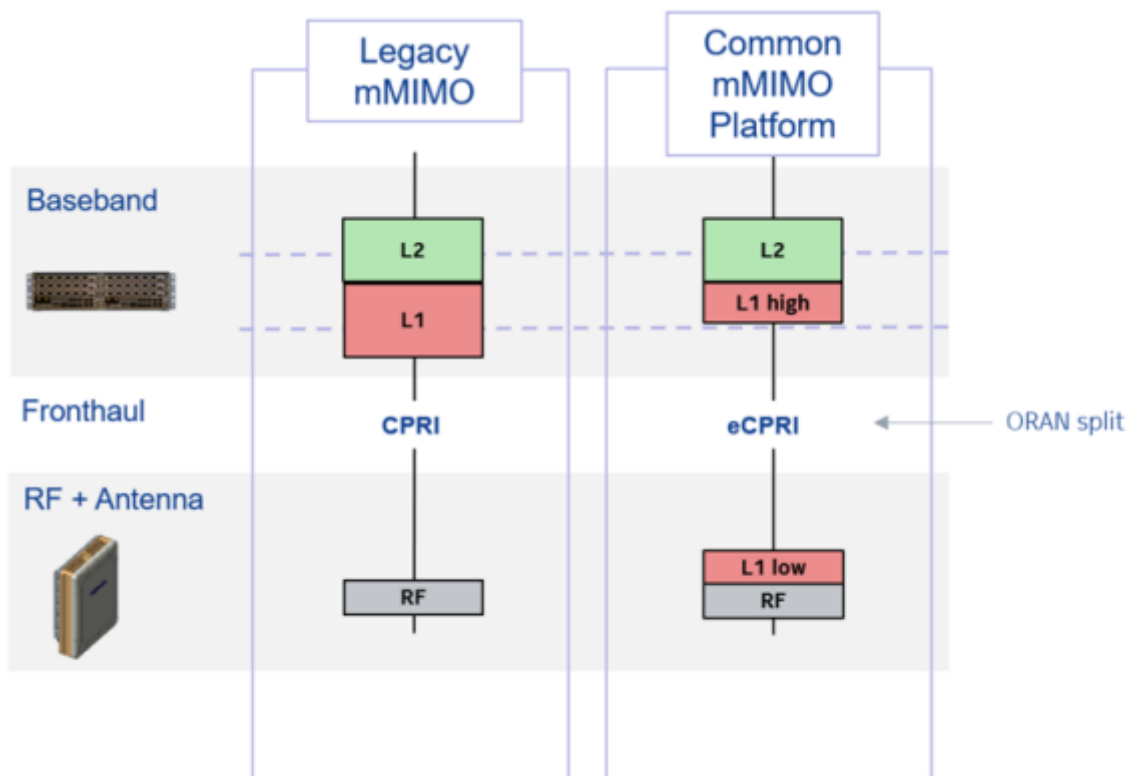
Common Public Radio Interface (CPRI) on käytetyin protokolla kantataajuusyksikön (BBU) ja antennin välisissä yhteyksissä 4G:ssä ja aikaisessa 5G:ssä. CPRI käyttää koaksiaalisen kaapelin sijaan optista kuitua datan siirrossa. (9, s. 21.)

CPRI:n toiminta perustuu I ja Q (IQ) datan lähetykseen, jossa I vastaa datan reaali-osaa ja Q datan imaginääristä osaa. Esimerkiksi RF signaalin vastaanotossa radio-osa digitalisoi datan ja

käärii sen CPRI protokollaan (IQ käyttäjädataan), joka siirtyy optisen kuidun läpi kantataajuusyksikköön. (9, s. 21 ja 25.)

Vastataksaan alati kasvaviin datanlähetyksmääriin samalla säilyttäen erittäin pienen vasteajan, NR mobiiliverkkoihin on suunniteltu uusi datapakettipohjainen protokolla kantataajuusyksikön ja antennin väliin nimeltä eCPRI (Enhanced Common Public Radio Interface) (9, s. 28).

L1 taso halkaistaan eCPRI:n myötä L1High- ja L1Low -tasoihin. L1 tason halkaiseminen tuo enemmän funktionaalisuutta keskusyksiköstä RF-moduuliin. L1 tason jako kuvassa 6. (10, s. 11.)



KUVA 4. L1 tason halkaisu (10, s. 12 kuva 1).

RF-moduulin puolelle tuodaan enemmän vastuuta L1Low'n myötä: ei reaaliaikainen (non-realtime) alustus ja konfiguraatio, johon kuuluu hallintatason viestit, sekä reaaliaikainen ohjaus, johon kuuluu ohjaus- ja käyttäjätason viestit (10, s. 11).

2.2.4 Fyysisen tason parannukset

LTE:stä tutun parannellun mobiililaajakaistan (eMBB) toiminnallisuuden lisäksi NR langaton liityntäverkko (wireless access network) tukee uusia palveluita, kuten teollinen IoT (IIoT), Ultra Reliable Low Latency Communication (URLLC) ja Vehicle-to-everything (V2X) (8, s. 99).

Teollisella IoT:lla tarkoitetaan laitteiden, sensorien ja applikaatioiden ekosysteemiä, jossa ne yhdessä esimerkiksi kerää, monitoroi ja analysoi dataa teollisista toiminnoista (11). URLLC tarkoittaa erittäin viiveetöntä yhteyttä, joka on 3GPP:n mukaan määritelty tarkoittavan 99,999 % luotettavaa yhteyttä, jossa käyttäjätason latenssi on 1 millisekunti yhdelle 32 tavun kokoiselle datapakettille. Tämän lisäksi keskimääräisen latenssin UL- ja DL siirrolle tulee olla 0.5 millisekuntia ilman määriteltyä luotettavuusprosenttia. (12, s. 3.) Matkapuhelinverkoissa hyödynnettävällä V2X (C-V2X) yhteydellä tarkoitetaan yhteyttä, jossa käyttäjinä toimii esimerkiksi ajoneuvot, kaukokytkinyksiköt (RSU) sekä jalankulkijat (13, s. 1).

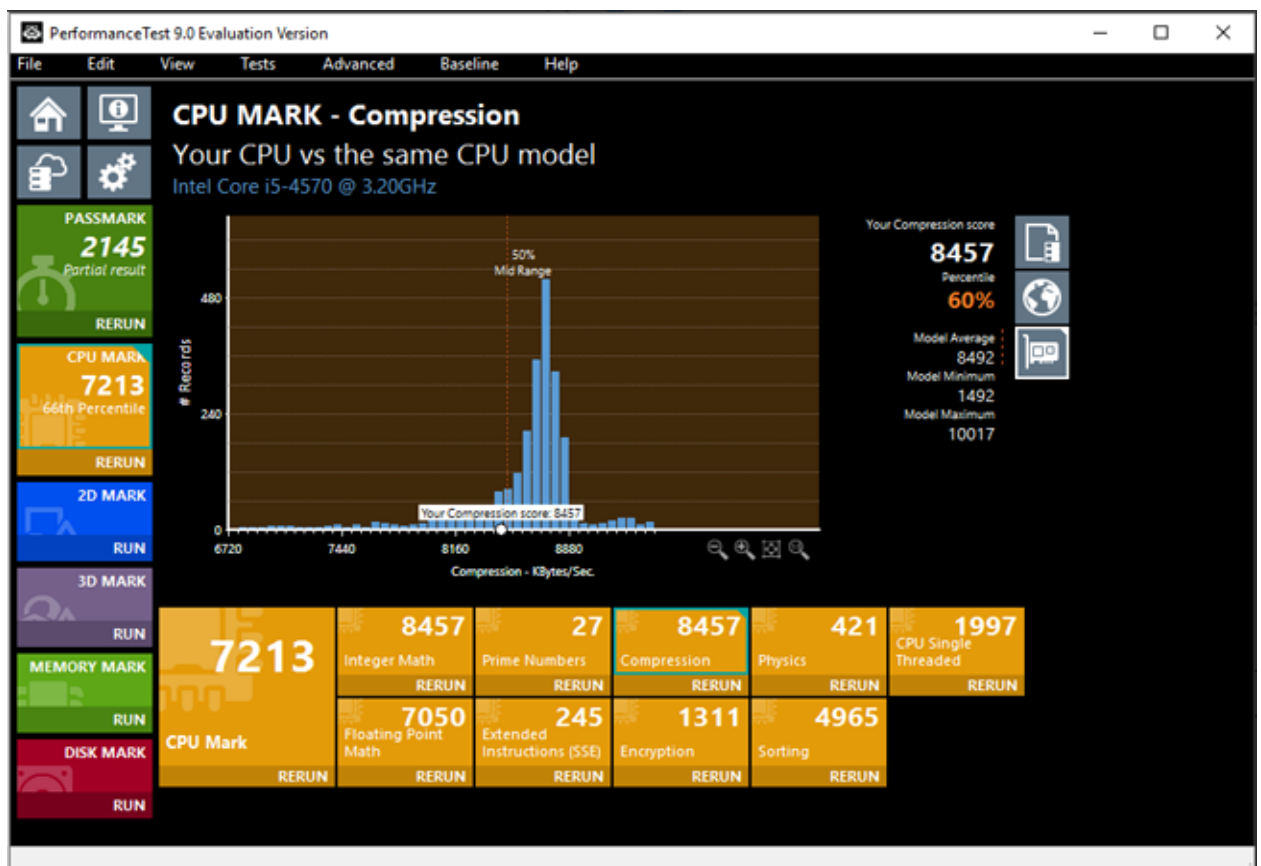
Tukeakseen monia eri NR:n käyttöönottopoja, joihin kuuluu eri kaistanleveydet monista megahertseistä useisiin gigahertseihin sekä kantoaallon taajuus alle gigahertsistä millimetrialloille asti, skaalautuvaa numerointia hyödynnetään apukantoaallon kannanleveydessä (subcarrier spacing) ja syklisen valmisteluelementin (cyclic prefix) pituudessa. Numerointi voidaan valita käyttötarkoituksen perusteella järjestelmän suorituskyvyn parantamiseen, esimerkiksi lähetin-vastaanottimen peittoalueen mukaan, samalla korjaten mahdollisia taajuushäiriöitä, kuten millimetrialloilla haitallisemmin esiintyvää vaihekohinaa. (8, s. 99.)

3 SUORITUSKYKYMITTAUS

Suorituskykymittaus (benchmarking) on prosessi, jonka tavoitteena on vertailla esimerkiksi tuotteita, järjestelmiä, palveluita tai organisaatioita. Tällöin nähdään eroavaisuudet sekä yhteiset piirteet vertailun kohteena olevien tutkimuskohteiden välillä, joilla voidaan selvittää mahdolliset kehitysaskleet ja optimointipotentiaali. (14, s. 1.)

IT-alan ihmiset kuitenkin yleisesti määrittelevät suorituskykymittauksen olevan työkalu, johon on yhdistetty metodologia, jossa tarkkaillaan ja vertaillaan järjestelmiä tai komponentteja erilaisilla kriteereillä, kuten suorituskyvyllä, luotettavuudella tai turvallisuudella. (14, s. 1.)

Hyvä esimerkki suorituskykymittauksista on PassMarkin PerformanceTest suorituskykyohjelmisto. Ohjelmistolla voidaan vertailla tietokoneen suorituskykyä verrattuna toisiin samanlaisiin tietokoneisiin, hyödyntämällä PassMarkin tietokantaa, jossa on miljoonista tietokoneista saatuja perusarvoja. PassMark tarjoaa ohjelmistossaan testisarjoja keskusyksikön, näyttöohjaimen, kovalevyn ja muistin mittaukseen. (15.)



KUVA 5. Kuvakaappaus PerformanceTest keskusyksikön suorituskykymittauksesta verkkosivulta (15).

Tässä opinnäytetyössä käytetään suorituskykymittausta terminä tuotteiden ja/tai tulosten vertailussa liittyen niiden suorituskykyyn ja luotettavuuteen.

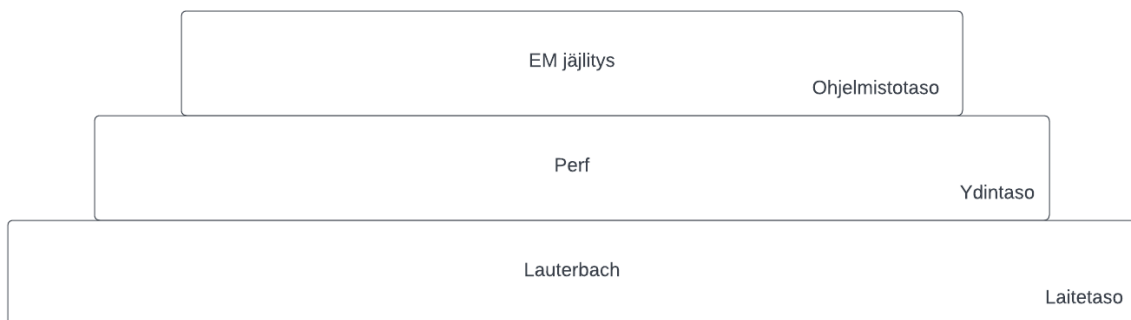
3.1 Suorituskykymittaus RF L1Low radioyksikössä

Opinnäytetyön pääpainona on suorittaa suorituskykymittauksia radioyksikössä ja esittää tulokset visualisoimalla, josta puhutaan tulevissa kappaleissa enemmän. Kysymyksenä onkin, minkälaisia suorituskykymittauksia voidaan suorittaa radioyksikössä.

Kaikista yksinkertaisin asia, jota voidaan mitata, on laitteiston suorituskyky. Käytännössä tämä tarkoittaa laitteiston kuormituksen seuraamista, esimerkiksi seuraamalla laitteiston lämpötilaa, muistinkäyttöä ja virrankulutusta. Kuormituksen kirjaaminen lokitiedostoihin erilaisten järjestelmäkomponenttitestien (SCT) aikana auttaa löytämään esimerkiksi pullonkauloja käytetystä ohjelmistosta tai laitteistosta.

3.2 Työkalut suorituskyvyn mittaamiseen

Radioyksikössä voidaan mitata suorituskykyä myös erilaisia työkaluja hyödyntäen, kuten EM tapahtumien jäljitystyökalu, Perf ja Lauterbach.



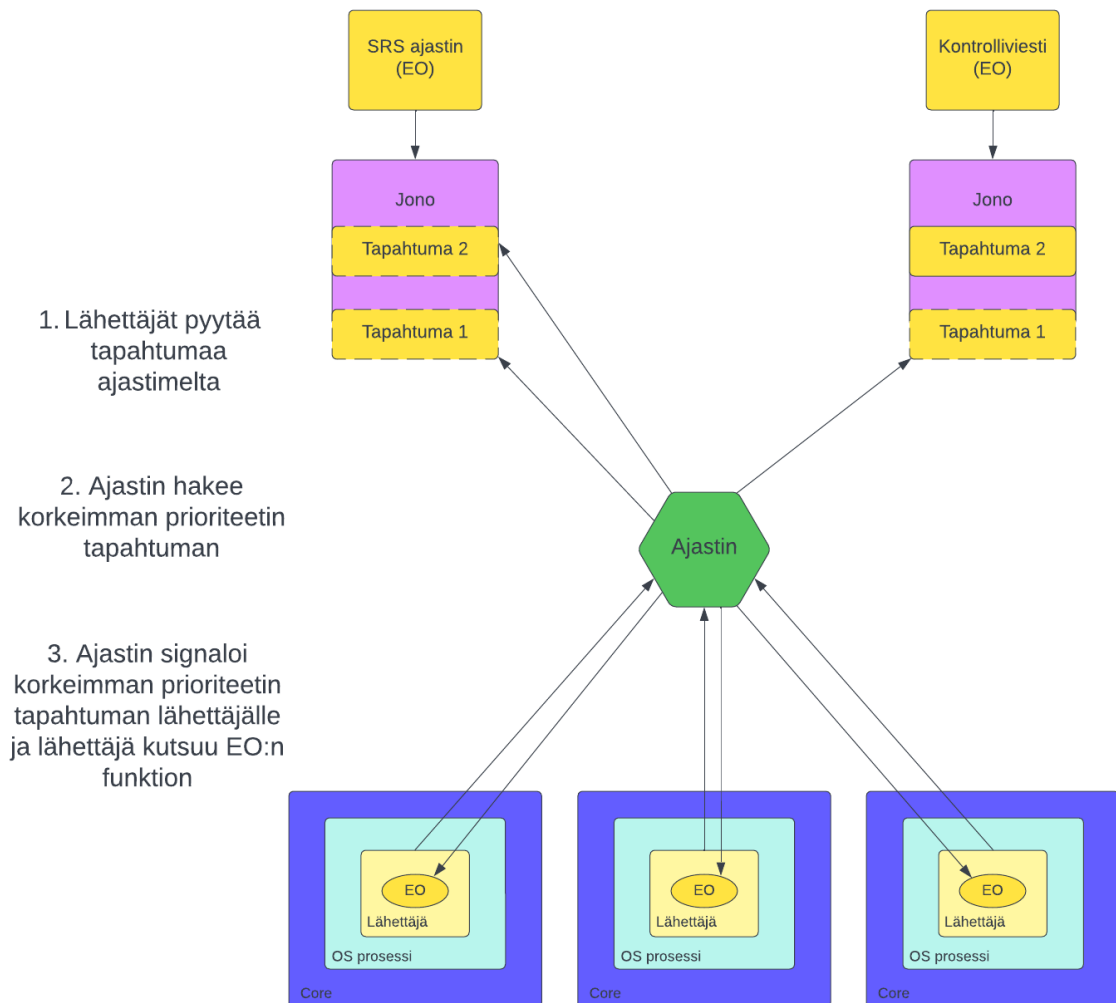
KUVA 6. Tasokuvaaja työkaluista

3.2.1 EM tapahtumien jäljitystyökalu

Event Machine (EM) on kehysympäristö, joka tarjoaa yksinkertaisen konseptin skaalata sekä dynaamisesti suorittaa kuormantasausta moniydin- tai monisäieapplikaatioissa erittäin vähäisellä rasitteella järjestelmälle. Tätä konseptia voidaan hyödyntää monissa eri käyttötarkoituksissa, mutta se on alkujaan suunniteltu käytettäväksi verkkoliikenteessä tiedonsiirtotasolla. (16.)

EM-konseptin pääelementit muodostaa tapahtumat, jonot sekä suorituksen kohteet (EO) ajastimen ja lähettäjän kanssa. Tapahtuman muodostaa applikaatiokohtainen data, kuten viesti tai datapaketti, ja se lähetetään tapahtumakohtaiseen asynkroniseen jonoon. Lähettäjä toimii silmukassa aina yhdellä säikeellä jokaisella ytimellä tai säikeellä, jos kyseessä on monisäikeinen ydin. Lähettäjä pyytää rajapinnan kautta pyytää ajastimelta tapahtumaa, jolloin ajastin tarjoaa korkeimman prioriteetin tapahtuman prosessoitavaksi. Tapahtuman perusteella lähettäjä etsii tapahtuman jonon ja lopulta kutsuu suorituksen kohteeksi rekisteröidyn funktion toimittakseen tapahtuman prosessoitavaksi. Tapahtuman käsiteltyä ja funktion suoritettua, yllämainittu tapahtumarakenne alkaa alusta. (16.)

On myös erittäin tärkeä muistaa, että aiemman kappaleen tilanne ajetaan rinnakkain kaikilla ytimillä EM-instanssissa, sillä EM-konsepti on suunniteltu mahdollisimman tehokkaaksi (16).



KUVA 7. Tapahtuman prosessointi EM-kehysympäristössä.

EM-jäljitystyökalu – nimensä mukaan – jäljittää näiden tapahtumien suoritusajoja sekä esimerkiksi näiden tapahtumien lähetysaikavälejä ja ytimien prosessointijärjestystä. Tapahtumien lähetysaikavälien tutkiminen voi olla todella tärkeääkin, sillä tapahtumien prosessoiminen tulisi olla mahdollisimman viiveetöntä.

TAULUKKO 2. Esimerkkidata jäljityksestä.

receive time (ns)	execution time (ns)	event ID	core ID	EO name	queue name
236684	20397932	0x0000		0 EM jäljitys	EM jäljitys
237782	3585	0x0001		1 Ajastin	Ajastin

Taulukossa 2 esitetty data on hyvin pieni pätkä jäljitysdatasta, ajetusta testistä riippuen dataa voi kertyä kymmeniä tuhansia rivejä. Taulukossa on saadun tapahtuman aika nanosekunteina, tapahtuman suorittamiseen kulunut aika nanosekunteina, tapahtuman ID, suorituksen kohteen nimi sekä jonon nimi.

3.2.2 Perf

Perf on työkalu, jolla pystytään suorittamaan kevyttä suorituskykyprofilointia. Työkalu on sisällytetty valmiiksi Linux-järjestelmän ytimeen, sillä Perf sai alkunsa työkaluna, jolla käytettiin suorituskykylaskureita Linux alijärjestelmillä. (17.)

Perfillä pystytään seuraamaan CPU:n suorituskykylaskureita ja jäljityspisteitä. Suorituskykylaskureita ovat CPU:n laitteiston rekisterit, jotka seuraa esimerkiksi suoritettuja ohjeita ja välimuistin käytössä aiheutuneita ”huteja”, joilla tarkoitetaan siis sitä, että välimuistista haettaessa tiettyä dataa, tätä dataa ei ole saatavilla tai sitä ei löydy välimuistista. Laskurit ovat profilointikäyttötapauksissa esimerkiksi ”kuumien”, eli kuormittavimpien käyttöalueiden tunnistamiseen. Suorituskykylaskureiden hieno puoli Perfissä on se, että voidaan tarjota tehtävä-, prosessori- ja kuormakohtaisesti laskurit. Jäljityspisteet on instrumentointipisteitä (instrumentation points), jotka ovat laitettu loogisiin pisteisiin koodissa, kuten järjestelmäkutsuihin tai TCP/IP tapahtumiin. Perf-komentoa käyttäen voidaan kerätä tietoa kuten aikaleimoja ja pinomuistin jäljitystä (stack trace), eikä mittauksen käyttämättömyyskään haittaa, sillä jäljityspisteet ovat minimaalinen rasite järjestelmälle. (17.)

Perf käyttää tapahtumakohtaista otantaa, jossa oletusotantana on prosessorin kellojakso eli sykli. Jos esimerkiksi näytetähdiksi määritetään 1000, niin Perf tallentaa näytteen aina 1000 kellojakson jälkeen. (18, s. 40.)

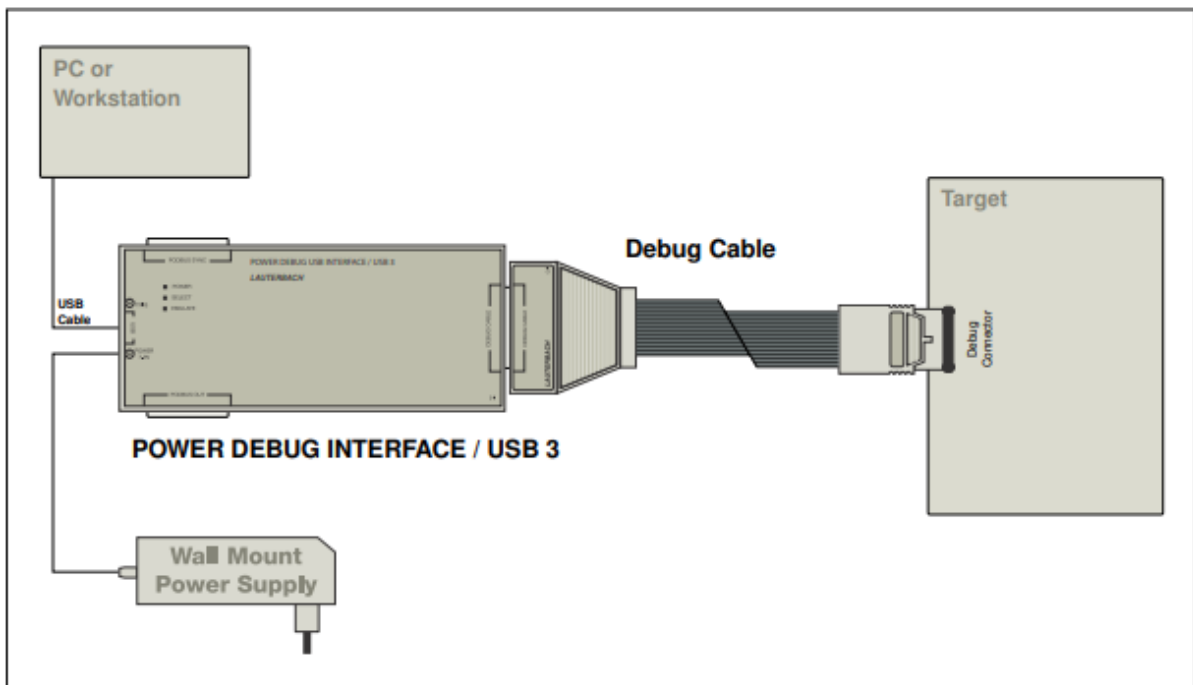
Perfin rakentama profiili perustuu tilastolliseen todennäköisyyteen, joten kerättyjen näytteiden määrä tulee olla tarpeeksi korkea, jotta pystytään tekemään luotettava profiili. Näytteiden määrä kuitenkin riippuu hyvin pitkälti mitatusta tapahtumasta; jos mitataan syklejä, niin näytetahti voidaan määrittää korkeaksi, koska kokonaisnäytemäärä syklien kanssa on korkea. (18, s. 40.)

3.2.3 Lauterbach

Lauterbach tarjoaa virheenkorjaustyökaluja prosessorisirulle, joilla pystytään analysoimaan multiydinprosessoreja laitteisto- tai ohjelmistopohjaisella virheenkorjauksella (19).

Lauterbachin TRACE32-virheenkorjain mahdollistaa sulautetun järjestelmän ja ohjelmiston testauksen käyttämällä sirun (on-chip) virheenkorjauskäyttöliittymää, joka yleisimmin on JTAG. JTAG (Joint Test Action Group) on käyttöliittymä, jota käytetään virheenkorjaukseen sekä ohjelmointiin esimerkiksi mikrokontrollereilla. (20, s. 7; 21.)

Virheenkorjaimen tyypillisiin ominaisuuksiin – prosessorista riippuen – kuuluu esimerkiksi rekisterin ja muistin lukeminen sekä kirjoittaminen ja ohjelman ajaminen sekä sen pysäyttäminen. (20, s. 7.)



KUVA 8. TRACE32-laitteisto, johon sisältyy aina universaali virheenkorjauslaitteisto (POWER DEBUG INTERFACE) sekä prosessoriarkkitehtuurikohtainen kaapeli (Debug cable) (20, kuvakaappaus s. 8).

Lauterbach tarjoaa myös TRACE32-laitteistoon lisäosia, kuten PowerTrace 3, jolla voidaan tallentaa jopa 8 gigatavun edestä jäljitysdataa sirulta rinnakkaiselta- tai sarjajäljitysportilta (20, PowerTrace 3).

TRACE32 toimii omassa graafisessa käyttöliittymässä (PowerView GUI), jossa pystytään laskea prosentuaalinen suoritusaika funktioille, prosesseille tai muuttujille. Tällaisia ovat esimerkiksi korkean tason funktiot, kuten C++ -kielen funktiot, muuttujat, joihin on määritetty jokin tietty sisältö, ja käyttöjärjestelmän tehtävät. (22.)

4 DATAN VISUALISOINTI

Opinnäytetyön yksi keskeisimmistä asioista on avata sitä, miksi datan visualisointi suorituskykymittauksissa on niin tärkeää. Hyvä visualisointi tekee muutakin kuin vastaa vain kysymyksiin; se auttaa huomaamaan, että on muitakin kysymyksiä joihin tarvitsemme vastauksen (23, luku 1).

4.1 Erilaiset visualisointitavat

Dataa voidaan visualisoida monin tavoin, pitkälti riippuen mikä käyttötarkoitus datalla on, eli mitä halutaan visualisoinnilla saavuttaa, ja minkälaista dataa on saatavilla.

Käytetään esimerkkinä laitteiston muistinkäytön mittaustuloksia sekunnin intervallilla (taulukko 3).

TAULUKKO 3. Esimerkkitaulukko muistinkulutuksesta testeittäin.

Mittaukset	Testi 1	Testi 2	Testi 3	Testi 4
1 s	7564 kb	8992 kb	8423 kb	8521 kb
2 s	7693 kb	8954 kb	7946 kb	8911 kb
3 s	7986 kb	9078 kb	7833 kb	8531 kb
4 s	8122 kb	8577 kb	8475 kb	8384 kb
5 s	8454 kb	8609 kb	8566 kb	
6 s	8128 kb	8234 kb	8381 kb	
7 s		8666 kb	8900 kb	
8 s		8843 kb		

Taulukossa 3 esitetään usean testitapauksen tulokset sekä mittaukset. Testitapauksien kestossa on eroja, jonka takia joissakin tapauksissa on enemmän mittaustuloksia kuin toisissa. Testien väliset muistinkäytön erot selittyvät sillä, että testattavassa tapauksessa laitteistoa saatetaan ajaa erilaisella konfiguraatiolla.

Jos kysymyksenä on että missä testissä keskimääräisesti laitteisto käyttää eniten muistia, niin taulukosta sitä ei välttämättä helposti huomaa. Tästä tätä esimerkkiä varten rakennetusta datasta

voidaan huomata nopeasti trendit, mutta jos mittaisimme vaikka muistinkulutusta nopeammilla iteraatioilla, kuten sekunnin sadasosan välein, ja testejä olisi monta kymmentä, niin datan lukeminen vaikeutuu huomattavasti. Tällöin dataa olisi kymmeniä tai satoja rivejä.

Värikoodaamalla taulukon pystytään hieman helpottamaan datan lukemista:

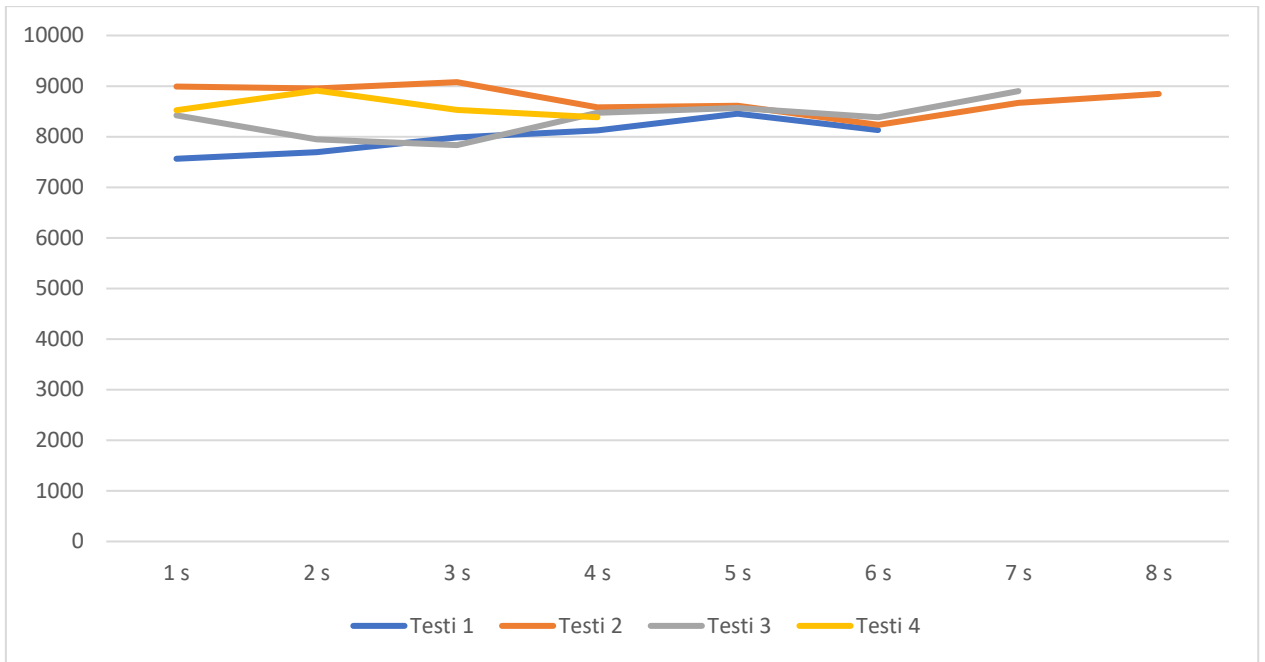
TAULUKKO 4. Värikoodattu esimerkkitaulukko Excelissä.

Mittaukset	Testi 1	Testi 2	Testi 3	Testi 4
1 s	7564	8992	8423	8521
2 s	7693	8954	7946	8911
3 s	7986	9078	7833	8531
4 s	8122	8577	8475	8384
5 s	8454	8609	8566	
6 s	8128	8234	8381	
7 s		8666	8900	
8 s		8843		

Tätä taulukon 4 tyyliä kutsutaan korostustaulukoksi, jossa on yhdistetty perinteinen tekstitaulukko sekä lämpökartta (23, luku 1). Värittämällä arvot pienimmästä suurimpaan, on paljon helpompi nähdä kysymykseemme vastaus, eli laitteisto käyttää eniten muistia testissä 2.

Taulukosta 4 myös tulee esiin trendejä, jotka tässä tapauksessa ovat siis vastauksia kysymyksiin, joita meillä ei vielä ole. Esimerkiksi huomaamme että ensimmäistä testiä ajettaessa muistinkulutus on vielä pieni verrattavissa muihin testeihin, mutta jo toisen testin alussa laitteisto käyttää paljon enemmän muistia. Tämä toistuu jokaisen testin alussa, joten voidaan tehdä johtopäätelmä, että laitteiston ensimmäisen testin voi vielä tapahtua jotain järjestelmän alustusta.

Hyvän visualisoinnin tarkoitus on myös tarkoitus herättää uusia kysymyksiä (23, luku 1).

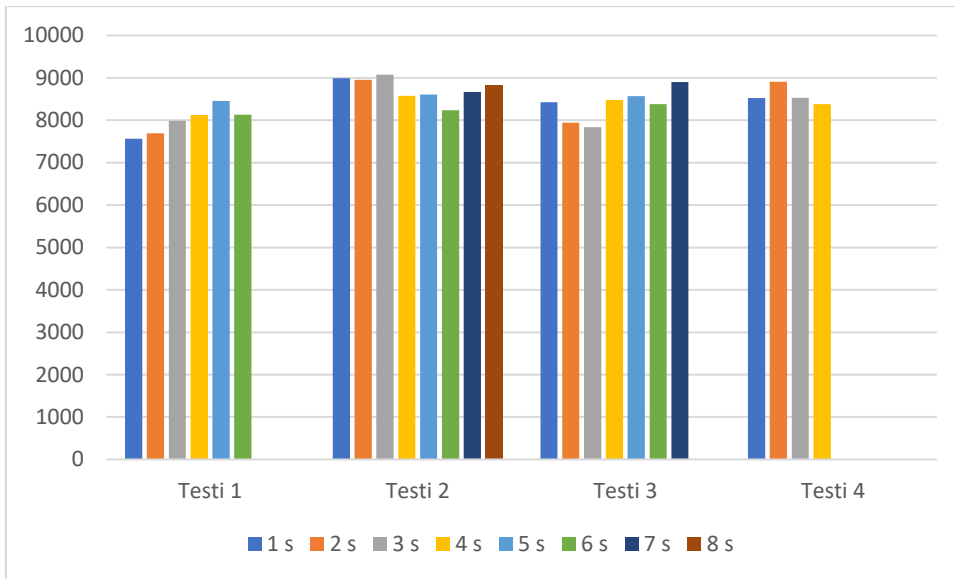


KUVA 9. Esimerkkitaulukko visualisoitu kuvaajana Excelissä.

Kuvan 9 kuvaajaa kutsutaan viivadiagrammiksi ja se on yksi yleisimmistä visualisointitavoista. Viivadiagrammi on myös yksi parhaimmista tavoista visualisoida trendejä ajan myötä. (23, luku 4.) Kuvaajasta helposti siis herää erilaisia kysymyksiä dataan liittyen, kuten aiemmin mainittu mahdollinen laitteiston alustus vielä ensimmäisen testin alussa, sillä käytetyn muistin määrä on selvästi noususuhtainen.

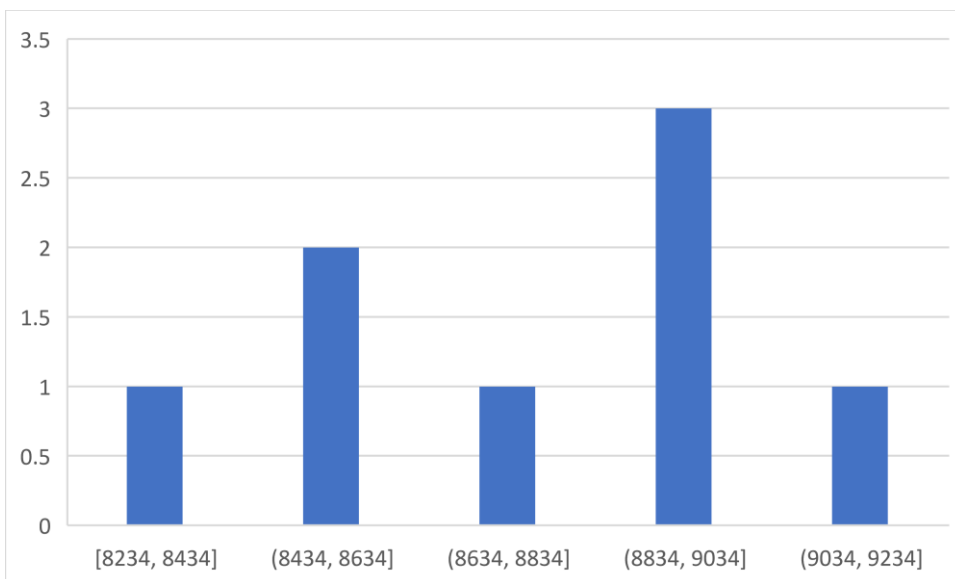
Käytetyimpiin visualisointityyleihin kuuluu myös pylväsdiagrammit ja histogrammit. Nämä kaksi tyyliä ovat hyvin samanlaisia, mutta teknisesti eroavina ominaisuuksina on histogrammin graafinen esitystapa, jossa data on jaoteltu joukkoihin (bins) ja se, että histogrammissa pylväät usein koskettavat toisiaan. Pylväsdiagrammit taasen usein esittävät dataa kategorisesti vertailtuna. (23, luvut 1 ja 2.)

Arvojen jakauma on monien ilmiöiden kanssa keskeinen ominaisuus. Histogrammi on yksi yleisimpiä jakaumakuviota, sillä siinä muuttujan arvoalue jaetaan luokkiin ja lasketaan, montako havaintoa kuhunkin luokkaan sijoittuu. (24.)



KUVA 10. Esimerkkitaulukko visualisoitu pylväsdiagrammina Excelissä.

Pylväsdiagrammit ovat hyvin monipuolisia, sillä data pystytään esittämään pysty- ja vaakatasossa. Visualisointityylissä pystytään myös hyvin suorittamaan vertailua pylväiden koon perusteella, olettaen että pylväillä on yhteinen lähtötilanne. (23, luvut 2 ja 4.)



KUVA 11. Esimerkkitaulukon toinen testi visualisoitu histogrammina Excelissä.

Kuvasta 11 huomataan, että keskimäärin toisessa testissä vaihtelu on pientä, suuria vaihteluita ei testin aikana tapahdu. Joukkoihin erotellusta datasta nähdään, kuinka eniten mittaustuloksia saatiin 8800–9000 kilotavun kohdalta.

Aiemmin mainittujen tyylien lisäksi on olemassa monia muitakin visualisointityylejä, joilla on oma tarkoituksensa. Tällaisia ovat esimerkiksi hajontakuviot, pinta-alakaavio ja ympyräkaavio. (23, luku 4.)

4.2 Visualisointialustat

Datan visualisointi usein vaatii kuitenkin työkalujen käyttöä, joten tässä kappaleessa tutustumme tunnettuihin visualisointityökaluihin.

Excel on mahdollisesti markkinoiden tunnetuin visualisointityökalu. Sen isoin myyntivaltti on monipuolisuus, sillä Excel on tunnettu laskennallisista ominaisuuksistaan, mutta se pystyy myös paljon muuhunkin. (25, luku 1.)

Excelillä pystytään suorittamaan kompleksisia laskutoimituksia käyttäen sisäänrakennettuja funktioita, kuten keskiarvon sekä summan laskemista. Tämän lisäksi pystytään laskutoimitukset automatisoimaan hyödyntämällä työkalun makrotoimintoa, jonka avulla voidaan rakentaa olemassa olevien funktioiden lisäksi kustomoituja funktioita. (25, luku 1.)

Excel myös mahdollistaa monien eri kuvaajien ja diagrammien käytön. Kaikki aiemman kappaleen kuvaajat ja taulukot ovat rakennettu Excelillä, ja erilaisia visualisointimahdollisuuksia näiden lisäksi on monia. Monien kuvaajien tai suorituskykyindikaattorien esitys yhdessä tiiviissä käyttöliittymässä, niin sanottujen kojelautojen, rakentaminen on mahdollista myös Excelissä. (25, luku 1.)

Datan visualisointia varten on olemassa kehysympäristöjä, jotka on suunniteltu vastaamaan nykYTEKNOLOGIASSA kasvavaan "big data" ilmiöön, jossa datamäärät voivat kasvaa satoihin tai tuhansiin gigatavuihin. Tällöin kehysympäristöjen käyttö data-analytiikassa on hyvinkin tärkeää, sillä se mahdollistaa esimerkiksi nopean datan yhdistämisen ja -haun sekä visualisoinnin. Kehysympäristön myötä myös pystytään rakentamaan monimutkaisia kojelautoja (dashboards), joissa pystytään nähdä data suoraan testien jälkeen automatisoimalla datan lähetys.

Elastic ja Grafana ovat yksiä tunnetuimpia datan aggregaatioon ja visualisointiin perustuvia kehysympäristöjä. Elastic tarjoaa niin kutsutun ELK pinon, johon kuuluu kolme avoimen lähdekoodin projektia: Elasticsearch, Logstash ja Kibana. Elasticsearch toimii data- sekä analytiikkamoottorina ja Logstash on prosessointiputki, joka voi ottaa dataa vastaan useilta lähteiltä samanaikaisesti. Logstash mahdollistaa datan manipuloinnin, jonka jälkeen se lähettää

sen esimerkiksi Elasticsearchiin. Kibana toimii ELK pinossa itse visualisointielementtinä, eli se mahdollistaa kuvaajien ja diagrammien rakentamisen datasta. (26.)

Grafana tarjoaa monia mahdollisuuksia datan analysointiin. Se mahdollistaa datan haun melkein mistä vain Grafanan liitännäisten kautta, sen visualisoinnin sekä kojelautojen rakentamisen. Kehäympäristössä voidaan suorittaa myös datamanipulaatiota, jolla voidaan muuntaa dataa ja tehdä laskutoimituksia eri hakujen ja lähteiden välillä, ja asettaa hälytyksiä jos esimerkiksi hausta tulostetut arvot ylittävät jonkin rajan. (27.)

5 EM TAPAHTUMIEN SUORITUSKYKYMITTAUS – KÄYTTÖTAPAUS

Opinnäytetyön käyttötapauksessa on tarkoitus selvittää EM-tapahtumien suoritusajkoja sekä tapahtumien aikaintervalli, eli aika tapahtumien välillä. Suunnitelmana on ajaa kaikki testit tietyllä RF-moduulilla, jossa prosessoidaan RTC-viestejä EM-jäljitysokalu aktivoituna. Tällöin saadaan testien aikana kutsuttujen tapahtumien lokit, jotka on tarkoitus lähettää käytettyyn data-analytiikkakehysympäristöön, jossa rakennetaan kojelauta datan visualisoimiseksi. Työn pääpainona tulee olemaan yllä mainittujen RTC-viestien suorituskykymittaus EM-tapahtumien kautta.

Käyttötapauksessa tullaan suodattamaan dataa, jotta kehysympäristöön lähetettävä tiedostokoko pidetään maltillisena ja vältetään turhan datan siirtämistä. Suodatettavan tiedoston muoto tulee olemaan CSV (comma-separated values), sillä alustavan tutkimuksen mukaan tällainen tiedosto on helposti käsiteltävissä työssä käytettävässä kehysympäristössä.

Motivaationa työlle on selvittää, miten eri testeissä lähetetyt EM-tapahtumat todellisuudessa käyttäytyvät eri laitteistoilla. Tällöin voidaan myös selvittää, onko ohjelmakoodissa mahdollisesti tapahtunut jokin vika, jos vaikka lähetysaikavälit, suoritusajat tai ytimien allokaatiot tapahtumille muuttuvat rajusti.

Työn teko tapahtuu suoraan opinnäytetyön toimeksiantajalle Nokialle, jonka tulokset tulevat käyttöön Nokian käyttämään visualisointialustaan. Käyttötapaukselta varten kuitenkin käytämme visualisointiin realistista ja suuntaa antavaa esimerkkidataa.

EM-jäljitys on lisätty Docker-virtuaalityöympäristössä olemassa oleviin testeihin tiimimme käyttämässä Git-versionhallinnan tiedostoarkistossa. Datan suodatukseen vaadittu komentosarja kuitenkin itse kirjoitetaan käyttämässä työympäristössämme ja lisätään Robot Framework -testiautomaatioon.

Tulosten visualisointi tapahtuu Splunkissa. Kappaleessa 5.2 mainitun Elasticin tavoin Splunk on kehysympäristö, jossa voidaan suorittaa laajaa data-analyysia visualisointia hyödyntäen kuten kojelautoja rakentamalla. (28.)

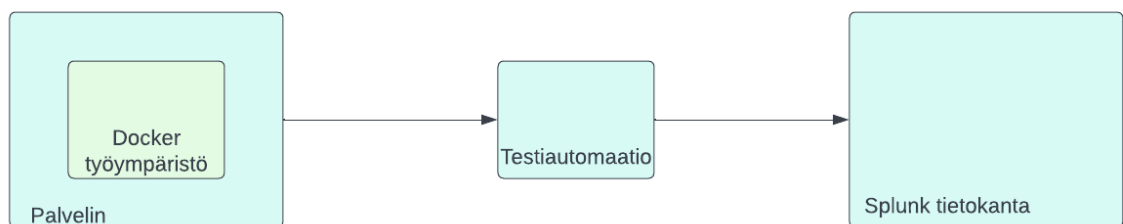
Kehysympäristö tarjoaa puitteet datan varastoinniseksi omiin indekseihin, joilla data saadaan eroteltua. Tällä ratkaisulla pystytään hakemaan vain tietyn tyyppistä dataa, eli testien aikana

saadut laitteiston lämpötilan seurantalokit ja testien tulokset ei ole samassa tietojoukossa, joka myös suoraan nopeuttaa tiedon hakemista Splunkin rajapinnasta. (29.)

Splunkissa pystytään koostamaan dataa Splunkin SPL:llä (Search Processing Language). SPL kattaa kaikki hakukomennot sekä niiden funktiot, ja sen vaikutusalueeseen kuuluu muun muassa datan hakeminen, suodattaminen ja manipulointi. (30.)

Yhtenä Splunkin tärkeimpänä ominaisuutena on mahdollisuus hyödyntää koneoppimista (ML) ja tekoälyä (AI). Tekoälyn ja koneoppimisen avulla pystytään ennustamaan tuloksia aiemman datan perusteella. Splunkin MLTK (Machine Learning Toolkit) tarjoaa valmiiksi rakennettuja algoritmeja helposti käytettäviksi graafisen käyttöliittymän kautta. (31; 32.)

Grafanan lailla myös Splunkkiin voidaan asettaa hälytyksiä. Hälytykset voidaan määrittää hyvin kattavasti, kuten haun tarkistus reaaliajassa tai tietyin väliajoin, ja toiminta voidaan määrittää esimerkiksi ilmoittamaan sähköpostitse tai ajamaan komentosarja hälytyksen lauettua. (32.)



KUVA 12. Käyttötapauksen rakenne. Sininen väri kuvaa palvelimia.

5.1 Käyttötapauksen toteutus

Työn toteutus lähtee liikkeelle työympäristöstä. Aluksi tarvitaan valmiit testitoteutukset, joissa käytetään RTC-viestejä sekä EM-jäljityksen käyttöönotto tälle tietylle tuotteelle. Helpotusta tuovat kuitenkin valmiit testitoteutukset, jolloin kyseisen laitteiston testejä ei tarvitse lähteä muokkaamaan tai lisäämään itse jäljitystä. Testien implementaatio voi olla hyvinkin erilainen verrattuna muiden laitteistojen testeihin, joten valmis rakenne jäljitystä varten helpottaa työn tekoa.

Seuraava askel on komentosarjan kirjoittaminen datan suodattamista varten. Analysoitavat tiedostot voivat olla kokoluokaltaan yli yhden megatavun, ja kun ottaa huomioon tiedostojen

määrän, koko voi kasvaa hyvinkin suureksi. Tämän lisäksi myös datan lähetyksessä Splunkiin tulee ottaa huomioon tiedostokoot, sillä Splunkissa voi olla rajoitettu datanlähetysmäärä.

Suodattamiseen tarkoitettu `bash` -komentosarja rakentuu turhan datan poistamiseen. Tärkein tapahtuma saada visualisoitua on RTC-viestit, joten komentosarja perustuu datan poistamiseen ennen ensimmäistä RTC-viestiä sekä viimeisen RTC-viestin jälkeen. Kaikki jäljitetyt tapahtumat kuitenkin tämän välillä säilytetään, jotta pystymme myös seuraamaan muiden tapahtumien käyttäytymistä viestien aikana. Kirjoitettu komentosarja on seuraava:

```
read -t 1 first_rtc_object last_rtc_object <<< $(grep -n rtc_object logikansio/testinimi_em_jaljitys.txt | awk -F: 'NR==1 {printf "%d ", $1}; END{print $1}'); awk -v f=$first_rtc_object -v l=$last_rtc_object 'NR>=f && NR<=l' logikansio/testinimi_em_jaljitys.txt | sed -E 's/ \+/,g;s,$//' | head -n 2000 > logikansio/testinimi_suodatettu_rtc_jaljitys.csv
```

Avataan komentosarja auki pala kerrallaan:

```
read -t 1 first_rtc_object last_rtc_object <<< $(grep -n rtc_object logikansio/testinimi_em_jaljitys.txt | awk -F: 'NR==1 {printf "%d ", $1}; END{print $1}');
```

Yllä oleva komento lukee `first_rtc_object`-muuttujaan ensimmäisen rivinumeron ja `last_rtc_object`-muuttujaan viimeisen rivinumeron `em_jaljitys` -tiedostosta etsimällä rivejä, joissa mainitaan `rtc_object` ja tulostamalla kyseisen rivin rivinumeron `grep -n`-komennolla.

```
awk -v f=$first_rtc_object -v l=$last_rtc_object 'NR>=f && NR<=l' logikansio/testinimi_em_jaljitys.txt | sed -E 's/ \+/,g;s,$//' | head -n 2000 > logikansio/testinimi_suodatettu_rtc_jaljitys.csv
```

Määritetään `awk`-komentoon aiemmin luetut rivinumeromuuttujat ja tulostetaan tiedostossa esiintyvä teksti näiden rivien välillä käyttämällä `'NR>=f && NR<=l'`, jossa NR kuvaa tiedoston rivinumeroa. Tulostetaan siis rivit, joiden rivinumerot ovat suurempia kuin `first_rtc_object` ja pienempiä kuin `last_rtc_object`.

`Sed`-komennon avulla poistetaan tekstistä välit ja muutetaan ne pilkuksi käyttämällä säännöllistä lauseketta (regular expression), joka on operaatioista muodostettu lauseke merkkijonojoukkojen tai säännöllisen relaation kuvaamiseen. Tätä hyödyntämällä tiedosto saadaan CSV-muotoon. (33).

Lopuksi määrittellään vielä maksimirivirajaksi 2000 riviä, jotta vältetään mahdollisia tiedostokoon ylityksiä `head -n 2000` -komennolla, ja tulostetaan suoraan rivit uuteen `rtc_jaljitys.csv` -tiedostoon.

Sitten kun komento tiedostojen suodattamista varten on valmis, seuraava askel on komennon integrointi testiautomaatioon. Robot Framework tarjoaa mahdollisuuden rakentaa omia korkeamman tason avainsanoja (keywords). Tällöin voidaan ajaa yhdellä avainsanalla useampi avainsana yhden avainsanan alla. Avainsanat ovat Robot Frameworkin määritelmä funktiosta; Robot Framework esimerkiksi tarjoaa valmiiksi rakennettuja avainsanoja kehysympäristönsä kuten kehysympäristössä määriteltyjen muuttujien konversio tekstiksi *Conversion To String* -avainsanalla. (34, luku 2.7; 35.)

Oma avainsana rakennettiin, jossa aiemmin kirjoitettu komentosarja määritetään muuttujaksi ja ajetaan komentorivissä onnistuneesti:

```
 ${cmd}= Catenate
```

```
 ... read -t 1 first_rtc_object last_rtc_object <<< $(grep -n rtc_object ${log_dir}/${test_name}_em_jäljitys.txt
```

```
 ... | awk -F: 'NR==1 {printf "%d ", $1}; END{print $1}'; awk -v f=$first_rtc_object -v
```

```
 ... l=$last_rtc_object 'NR>=f && NR<=l' ${log_dir}/${test_name}_em_jäljitys.txt | sed -E 's/\+/,/g;s/,/$/'
```

```
 ... | head -n 2000 > ${log_dir}/${test_name}_suodatettu_rtc_jäljitys.csv
```

```
 Run Command  ${cmd}
```

Catenate-komentoa käytetään yhdistämään tekstit, jotta komennon luku olisi helpompaa eikä kaikki teksti olisi yhdellä rivillä. Komennossa näkyvät sanat `${log_dir}` ja `${test_name}` tarkoittavat muuttujia, jotka syötetään avainsanaan argumentteina. Tällöin voidaan automatisoida tulostus joka testitapauksen jälkeen ajamalla avainsana EM-jäljityksen jälkeen testin lopussa.

Run Command on itse rakennettu avainsana, joka ajaa komennon komentorivillä ja tarkistaa, että komento suoritettiin onnistuneesti. Tämän olisi voinut ajaa myös käyttämällä Robot Frameworkin omaa Process -kirjastoa, joka käyttää Pythonin *subprocess* moduulia sekä sen *Popen* luokkaa. Kirjastosta löytyvät avainsanat Run Process ja Start Process, joiden ero on se, että Run Process odottaa ajon loppuun asti, kun Start Process aloittaa komennon ajon ja laittaa sen taustalle. Prosessien ajon tuloksen pystyy tarkistamaan prosessin tulosobjektista, joka on 0, jos ajossa ei tapahtunut virhettä. (36.)

Ajamalla avainsana joka testitapauksen jälkeen, pystytään automatisoida suodatus kyseisille jäljitystiedostoille. Seuraavana askeleena onkin saada data lähetettyä jatkuvan integraation (CI) putken kautta Splunkiin.

Hyödynnetään olemassa olevaa funktiota testilinjastossa käytettävässä skriptissä, johon lisätään datan lähetykseen tarkoitettu funktio, jonka parametreiksi asetetaan etsittävä merkkijono korvausmerkin kanssa, maksimitiedostokoko sekä indeksi. Korvausmerkki mahdollistaa sen, että kaikki tiedostonimet, joissa annettu merkkijono esiintyy, prosessoidaan ja lähetetään Splunkiin. Esimerkiksi aiemmin mainitussa komentorivissä suodatettu data on varastoitu testikohtaisesti suodatettu_rtc_jäljitys.csv -tiedostoon. Täten hyödyntämällä korvausmerkkiä testinimen kohdalla, voidaan lähettää kaikki kyseiset tiedostot Splunkin tietokantaan:

```
*_suodatettu_rtc_jäljitys.csv
```

Maksimitiedostokoon asettaminen luo turvaa, sillä tämän myötä vahingossakaan ei voida lähettää liian suuria tiedostoja, jos esimerkiksi kyseisen komentorivin toiminnallisuus joskus muuttuisi, sillä funktio tarkistaa tiedostojen koon ennen niiden lähetystä Splunkiin.

Aiemmin esittelyssä mainittu datan indeksointi mahdollistaa datan erottelun, joten EM-tuloksia varten on oma indeksi, johon varastoimme datan. Näin pystytään myös seuraamaan tarkemmin paljonko dataa lähetetään tähän indeksiin esimerkiksi kuukauden aikana ja tätä kautta tarkentamaan datanlähetyksärajoja kyseiselle indeksille.

Prosessoituaan listan valideista tiedostoista, funktio kutsuu tiedosto kerrallaan Python skriptiä, joka hoitaa tiedostojen lähetyksen HEC:ia (HTTP Event Collector) hyödyntäen. HEC mahdollistaa datan ja tapahtumien lähetyksen Splunkiin HTTP ja HTTPS protokollien kautta käyttämällä tunnus pohjaista todennusta. (37.)

Varsinaiseen datan lähetykseen rakennettu Python skripti pystyy tunnistamaan datatyypin, joka tässä tapauksessa on CSV-muodossa. Toiminnallisuus perustuu datatyyppiin, eli tässä tapauksessa lähetämme rivi kerrallaan tietoa Splunkiin, jossa jokainen rivi on oma yksittäinen tapahtumansa. Splunkissa voimme tiedostokohtaisesti katsoa tapahtumien määrän ja aloittaa datan visualisoinnin ja manipulaation.

Datan lähetyksen jälkeen alkaa visualisointi. Lähdetään avaamaan SPL:n hakusyntaksia pala kerrallaan (liite 1). Syntaksi on peräisin liitteestä 3, ja tämä kuvaaja on valikoitunut esimerkiksi, koska kuvaaja rakentuu monimutkaisimmin kaikista liitteistä:

```
index="index_for_test_data" source=test_data_for_thesis.csv
```

Haemme kyseisestä indeksistä data, jonka lähteenä on "test_data_for_thesis.csv" tiedosto.

```
| rex field=_raw
"(?ms)^(?P<receive_time>[^,]+),(?P<execution_time>[^,]+),(?P<event_id>[^,]+),(?P<core_id>[^,]+),(?P<eo_name>[^,]+),(?P<queue_name>[^,]+),(?P<payload_size>[^,]+)"
```

Saamamme data käsitellään *rex*-komennolla, jolla erottelemme datan muuttujiin säännöllistä lauseketta hyödyntäen, joka tässä esimerkissä erottelee datan muuttujiin käyttämällä pilkkua erottajana. *Field* merkkää haluamaamme lähdetä, josta data erotellaan, joka tässä tapauksessa on raakadata (*_raw*) eikä esimerkiksi jokin muuttuja, johon olisi säilöttynä tietoa.

```
| search "rtc_eo"
```

```
| sort receive_time
```

Haetaan vain tapahtumat, joihin on sisällytetty "rtc_eo", sillä se on ainoa EM-suorituskohde jota tässä kuvaajassa seurataan. Tämän lisäksi tulokset järjestetään aikaleiman (*receive_time*) mukaan.

```
| streamstats range(receive_time) window=2 as time_diff
```

Streamstats-komento mahdollistaa kumulatiiviset tulokset tapahtumille, eli tulokset tulostuvat tapahtumien haun aikana. Komennossa käytetään *window*-parametriä, joka määrittää tapahtumien lukumäärän, jolle suoritetaan aggregaattifunktio *range*. Funktio laskee aina kahden aikaleiman erotuksen ja asettaa sen *time_diff*-muuttujaan.

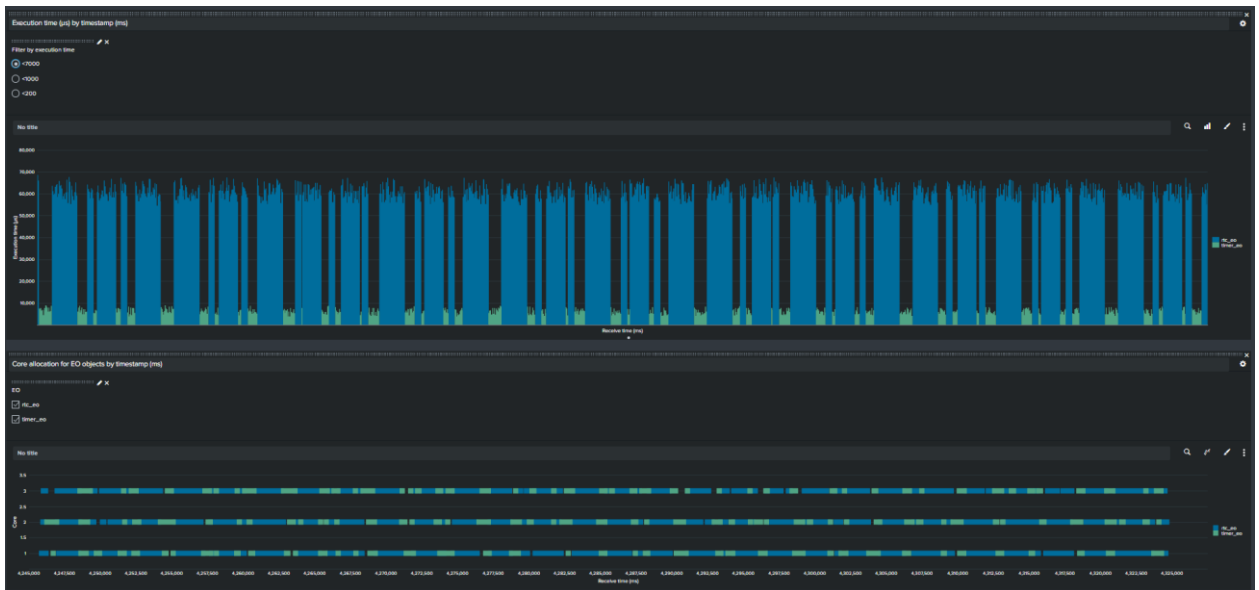
```
| search time_diff > 0
```

```
| stats count(receive_time) AS amount_of_consecutive_rtc_eos BY time_diff
```

```
| chart values(amount_of_consecutive_rtc_eos) by time_diff span=1
```

Haetaan vain data, jonka erotus on yli 0, ja lasketaan, kuinka montako kertaa aikaleiman aikana esiintyy *time_diff*-arvoja. Lopulta tulostetaan kuvaaja, jossa esiintyy nämä arvot erotuksen funktiona. Käyttämällä *span=1* pystytään arvot esittämään lineaarisesti niin, että nähdään kokonaiskuva arvojen jakaumasta, eikä vain pelkkiä erotuksia.

Lopulta rakennetaan visualisointia varten kappaleessa 5.2 mainittu kojelauta, jossa kaikki tärkeät aspektit datasta on visualisoituna (kts. liitteet 1-4).



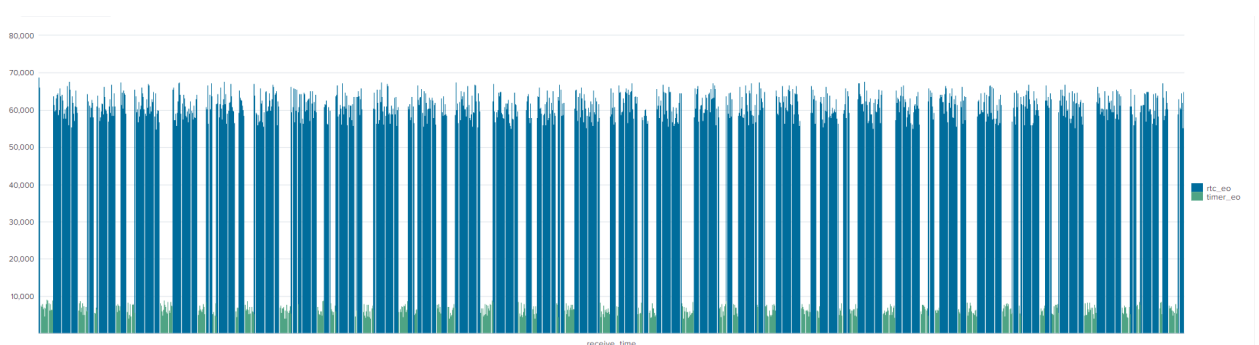
KUVA 13. Osa rakennetusta kojelaudasta.

Kojelautojen käyttö helpottaa usean näkymän automatisointia. Kojelaudat antavat mahdollisuuden käyttää ”tulopaneeleja”, joihin voimme manuaalisesti syöttää muuttujia, kuten esimerkiksi tuotteiden nimiä. Näistä muuttujista voidaan rakentaa valikoita käyttäjille, kuten radiovalikko ja ruutuvalikko (kuva 13). Käyttäjän valitsemat arvot tallentuvat kojelaudan sisäiseen muuttujaan, jota voidaan käyttää visualisoinnin hakuvaiheessa.

Tulopaneeliin ilmestyviä muuttujia voidaan myös automatisoida käyttämällä Splunkin tulopaneelissa dynaamista hakua. Hakemalla dataa aiemmin kappaleessa mainitulla tavalla indeksistä ja lähteestä, voidaan hausta prosessoida muuttujia, jotka syötetään valikkoon.

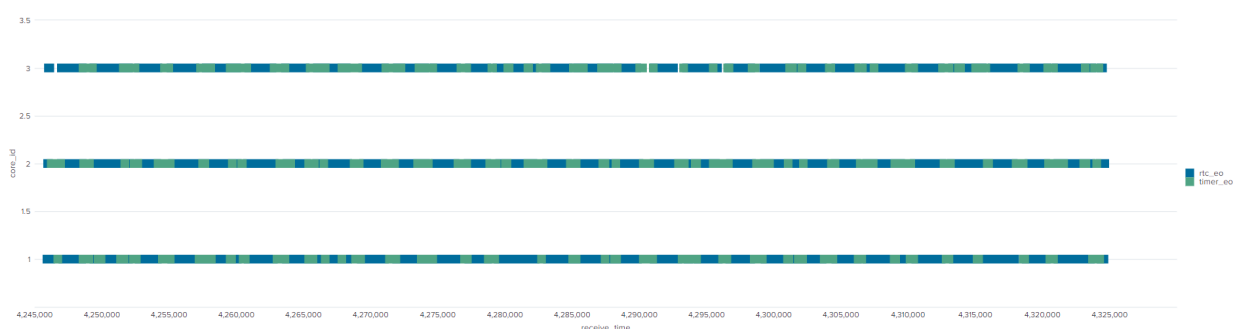
5.2 Käyttötapauksen tulokset

Viimeinen askel käyttötapauksessa on visualisoinnin tuloksien analyysi. Tärkeää on ymmärtää mitä rakennetulla kojelaudalla oikein esitetään, joten käydään läpi visualisointi kuvaaja kerrallaan:



KUVA 14. Suoritus aika ajan funktiona (tarkempi kuva liitteessä 2).

Yllä oleva kuvaaja on pylväsdiagrammi, jossa esitellään kahden EO tapahtuman (rtc_timer_eo ja srs_timer_eo) suoritusajat EM-kehysympäristössä ajan funktiona. Kuvaajasta voidaan tärkeimpinä asioina seurata EO tapahtumien suoritusajojen trendejä, eli voidaan tutkia tapahtuuko muutoksia esimerkiksi hallintaviestin lähetyksen tapahtumissa EM-kehysympäristössä, ja suoritusajojen vaihtelevuutta suorituskertojen välillä, jolloin nähdään suoritusajojen ennustettavuus.



KUVA 15. EO tapahtumien allokaatio ytimille ajan funktiona (tarkempi kuva liitteessä 3).

Kuva 15 kertoo EO tapahtumien allokaatiosta ytimille ajan funktiona. Kuvaaja on hajontakaavio, joka mahdollistaa käyttötarkoitusta varten tarkemman kuvaajan EO allokaatioista ytimillä ja myös näiden tapahtumien suoritusajoista. Tämä kuvaaja on ainakin omasta mielestä yksi tärkeimpiä kuvaajia EM-kehysympäristöön liittyen, sillä tästä pystymme näkemään tapahtumien allokaation ja aikavälin niiden välillä. Esimerkkidatasta näemme, kuinka tapahtumien allokaatio on jatkuvaa ja käytännössä viivettä tapahtumien välillä ei ole.

Kyseinen kuvaaja tapahtumien allokaatiosta ytimille auttoi myös löytämään vian RTC-viestien lähetyksessä. Esimerkkinä testitapauksessa lähetetään silmukassa 500 viestiä viidelle EM-jonolle, jossa ensimmäiset 100 viestiä jakautuvat ensimmäiseen jonoon, toiset 100 viestiä jakautuvat toiseen jonoon ja niin edelleen. Tapauksessa kuitenkin kävi niin, että jonot eivät ehtineet tyhjentyä oikeassa ajassa ja testien loppuajana oli vain yksi ydin käytössä kun muut jonot olivat kerenneet tyhjentyä. Kuvaajan löydösten perusteella tämä ongelma pystyttiin korjaamaan muuttamalla viestien lähetyksen toiminnallisuutta.

Kuvan 16 kuvaaja on visualisoinneista monimutkaisin. Seuraamme aina kahden saman suorituskohteen lähetyksaikaväliä ja visualisoimme y-akselilla yhteenlasketun lukumäärän samalla lähetyksaikavälillä lähetetyistä suorituskohteista. Kuvaajan tapauksessa ollaan keskitytty vain

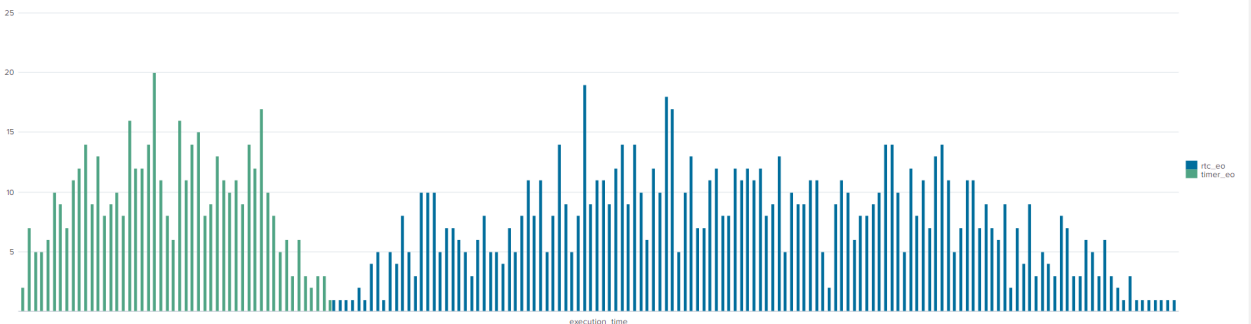
RTC-tapahtumiin, eli näemme yhteenlasketun lukumäärän RTC-tapahtumista lähetysaikavälin funktiona.



KUVA 16. Kahden peräkkäisen RTC tapahtuman lukumäärä näiden lähetysaikavälin funktiona (tarkempi kuva liitteenä 4).

Kuvaajan erottaa myös x-akselilla hyödynnetty lineaarinen datan esitystapa, jolla tarkoitetaan arvojen jakautumista x-akselilla. Ensimmäinen pylväs voi kuvata arvoja esimerkiksi aikavälillä 8, kun taas viimeinen pylväs arvoja aikavälillä 5000. Asettamalla x-akselin arvot esimerkiksi välillä 1, data visualisoidaan aina yhden arvon välein. Näin saadaan esille myös tyhjät arvot aikavälillä, eli nähdään jakautuma lähetysaikavälien välillä.

Kuva 17 on kuvaaja, joka kertoo EO-tapahtumien lukumäärän ajan funktiona.



KUVA 17. EO tapahtumien lukumäärä suoritusajan funktiona (tarkempi kuva liitteenä 5).

Kuvaajan tyyli on histogrammi, eli data on jaoteltu suoritusajan mukaan joukkoihin. Kuvaajassa esitetään EO-tapahtumien lukumäärä kyseisellä suoritusajavälillä. Näin pystymme näkemään, mikä on yleisin suoritusajaka esimerkiksi rtc_eo:lla. Kuvaajasta myös näemme hajonnan suoritusajojen lukumäärän välillä.

6 YHTEENVETO

Opinnäytetyössä tutkittiin langattomassa radioverkkotekniikassa käytetyn RF-moduulin suorituskykyä visualisoinnin kautta. Mobiiliverkkojen radio-osassa laitteistojen suorituskyvyn kehittäminen on äärimmäisen tärkeää radioverkkoteknologian kehittyessä sukupolvelta toiselle, sillä suuntana on toiminnallisuuden siirtyminen enemmän kantataajuusyksiköltä antennille, jotta radiojärjestelmistä saataisiin tehokkaampia ja datanlähetyksessä entistäkin nopeampaa.

Verkkojen välillä on tapahtunut paljon muutoksia siirtyessä sukupolvesta toiseen. 5G NR hyödyntää keilanmuodostusta, jolla signaali pystytään kohdistamaan tiettyyn suuntaan. Keilanhallintamenetelmillä voidaan kompensoida millimetriaalloilla signaalin vaimennusta, mikä on yksi mahdollistaja 5G:n suorituskyvyssä, sillä millimetriaallot tarjoavat enemmän kaistaa, joka johtaa korkeampiin datanlähetyksimääriin.

Muutoksia on tehty myös esimerkiksi kantataajuusyksikön ja antennin välillä käytettyyn protokollaan, jossa signaali siirtyy datapakettipohjaisesti. Fyysisesti muutos on iso, sillä koaksiaalikaapelin sijasta data siirtyy optisen kuidun läpi.

Pääpainona työssä kuitenkin oli suorituskykymittauksien analyysi visualisoinnin kautta. Suorituskykymittauksien avulla tarkkaillaan ja vertaillaan tuotteita tai komponentteja esimerkiksi suorituskyvyn tai luotettavuuden kannalta. Suorituskykymittauksia pystytään suorittamaan eri tasoilla, käyttämällä mittauksia varten rakennettuja työkaluja.

Visualisointi tarjoaa puitteet eri tilanteisiin. Visualisoimalla tulokset, nähdään trendit tuotteiden välillä. Eri kuvaajien avulla pystytään tarpeen mukaan visualisoimaan melkein mitä vain. Tätä vahvistaa esimerkiksi kehysympäristöjen kuten Splunkin käyttö, jonka avulla voidaan reaaliaikaisesti visualisoida hakemalla ja manipuloimalla dataa. Datan lähetyksessä pystytään myös automatisoimaan, ja voimme rakentaa monen kuvaajan näkymiä, kojelautoja, jolloin kaikki tärkeä data on visualisoituna yhdellä sivulla.

Suorituskykymittausten visualisoinnin tärkeys tulee ilmi käyttötapauksesta. Käyttötapauksena toimii EM-kehysympäristössä lähetettyjen tapahtumien prosessointi, josta tärkeimpinä asioina mittaamme RTC-viestien suoritusajankoja, lähetyksiaikavälejä ja ydinllokaatiota. Rakentamalla näille kojelaudan, pystymme testiautomaation kautta seuraamaan päivittäin tuloksia RTC-viesteihin liittyvistä testeistä ja näkemään, prosessoidaanko viestit oikein.

Käyttötapauksesta saatiin myös konkreettista hyötyä, sillä huomattiin, että RTC-viestien lähetyksessä on ongelmaa. Tapahtumien allokaatiot eivät olleet täysin optimaalisia, jotka huomattiin hajontakaaviosta, kun RTC-tapahtumien prosessoinnissa oli viivettä.

Täydellisesti ei kuitenkaan työssä onnistuttu. Tuloksia lähettäessä käyttämäämme kehysympäristöön, ei otettu huomioon mahdollisia ongelmia tiedostokoon ja datatyypin kanssa. Käytetty CSV-muoto johti pitkiin lähetyksaikoihin testilinjoiissamme, sillä olemassa oleva komentosarja lähetyksestä varten lähettää datan yhden rivin tapahtumina, jolloin lähetetään 2000 tapahtumaa yhden sijasta. Tämä johti yli puolen minuutin lähetyksaikoihin. Jos olisi käytetty JSON-muotoa tiedostossa, tiedoston olisi voinut lähettää yhtenä tapahtumana ja lähetyksaika olisi laskenut yli 93 %, eli lähetyksaika olisi ollut vain kaksi sekuntia.

Paljon myös opittiin työn aikana. Suorituskyvyn mittaaminen ja tulosten analysointi auttoi ymmärtämään laitteistojen toimintaa sekä käyttämiä kehysympäristöjä ja työkaluja. Osaamisen kasvu esimerkiksi Splunkin kanssa tulee auttamaan tulevaisuudessa, sillä käyttötapausten kaltaisia mittauksia tullaan suorittamaan tulevaisuudessa erilaisia työkaluja, kuten aiemmassa kappaleessa 3.2.2 esiteltyä Perf:iä käyttäen.

Opinnäytetyö kaiken kaikkiaan esittää sen, että visualisointi on isojen datamäärien kanssa työskennellessä avain onnistumiseen. Kuvaajat mahdollistavat syvemmän analyysin, ja testattavien laitteistojen testituloksien automatisointi myös mahdollistaa vertailun eri laitteistojen suorituskyvyssä aina ajettujen testien jälkeen.

Seuraavana askeleena voisikin olla tämän edellä mainitun skenaarion tekeminen, ottamalla EM-jäljitys käyttöön muissakin laitteistoissa ja rakentamalla kojelaudan, jolla näkisimme esimerkiksi yhden kuvaajan kautta useamman laitteiston EM-tapahtumien prosessoinnin, kun lähetetään samoja RTC-viestejä. Myöskin ongelmakohtana kerrotun tiedostotyypin muuttaminen tulee olemaan kehityskohtana.

Nähtäväksi jää, kuinka tulevaisuus ja 6G:n käyttöönotto tulee vaikuttamaan radiolaitteistoihin ja niistä saatuihin suorituskykymittaustuloksiin.

LÄHTEET

1. Tse, David & Pramod Viswanath 2005. Fundamentals of wireless communication, Cambridge University Press. Hakupäivä 26.9.2022. ProQuest Ebook Central. Vaatii käyttöoikeuden.
2. Cox, Christopher 2012. An introduction to LTE: LTE, LTE-advanced, SAE and 4G mobile communications, John Wiley & Sons. Hakupäivä 27.9.2022. O'Reilly Online Learning. Vaatii käyttöoikeuden.
3. Myllylä, Markus 2011. Detection algorithms and architectures for wireless spatial multiplexing in MIMO-OFDM systems. Oulun yliopisto. Acta Universitatis Ouluensis C Technica 380. Väitöskirja. Hakupäivä 30.09.2022. [Detection algorithms and architectures for wireless spatial multiplexing in MIMO-OFDM systems \(oulu.fi\)](https://jultika.oulu.fi/files/nbnfi-fe202102185273.pdf).
4. Dahlman, Erik, Parkvall, Stefan & Skold, Johan 2011. 4G: LTE/LTE-Advanced for Mobile Broadband, Elsevier Science & Technology. Hakupäivä 26.9.2022. ProQuest Ebook Central. Vaatii käyttöoikeuden.
5. Dahlman, Erik, Stefan, Parkvall & Skold, Johan 2020. 5G NR: The Next Generation Wireless Access Technology, Academic Press. Hakupäivä 1.10.2022. O'Reilly Online Learning. Vaatii käyttöoikeuden.
6. Cox, Christopher 2020. An Introduction To 5G: The New Radio, 5G Network and Beyond, John Wiley & Sons. O'Reilly Online Learning. Vaatii käyttöoikeuden.
7. Perera, A. Indika, Manosha, K. B. Shashika, Rajatheva, Nandana & Latva-aho, Matti 2020. An Initial Access Optimization Algorithm for millimeter Wave 5G NR Networks. Centre for Wireless Communications, University of Oulu, Finland. Hakupäivä 1.10.2022. <http://jultika.oulu.fi/files/nbnfi-fe202102185273.pdf>.
8. Sirotkin, Sasha 2020. 5G Radio Access Network Architecture: The Dark Side Of 5G, John Wiley & Sons, Incorporated. Hakupäivä 3.10.2022. ProQuest Ebook Central. Vaatii käyttöoikeuden.
9. Zenir, Maher 2020. ECPRI TIMING MEASUREMENT AND TESTING FOR 5G NEW RADIO. Oulun yliopisto Pro gradu -tutkielma. Hakupäivä 29.10.2022. <http://jultika.oulu.fi/files/nbnfioulu-202007082740.pdf>.
10. Laihiainen, Jaakko 2022. ORAN-standardin vaikutukset L1Low-radio-ohjelmistoon. Oulun ammattikorkeakoulu. Tietotekniikan tutkinto-ohjelma. Opinnäytetyö. Hakupäivä 11.10.2022.

- https://www.theseus.fi/bitstream/handle/10024/753561/Laihiainen_Jaakko.pdf?sequence=2.
11. Cisco. What Is Industrial IoT (IIoT)? Hakupäivä 4.10.2022.
<https://www.cisco.com/c/en/us/solutions/internet-of-things/what-is-industrial-iiot.html>.
 12. Popovski, Petar, Stefanovic, Cedomir, Nielsen, Jimmy J, de Carvalho, Elisabeth, Angelichinoski, Marko, Trillingsgaard, Kasper F, Bana & Alexandru-Sabin 2019. Wireless Access in Ultra-Reliable Low-Latency Communication (URLLC). IEEE transactions on communications 67 (8). Hakupäivä 4.10.2022.
<https://vbn.aau.dk/ws/files/310030894/08705373.pdf>.
 13. Garcia, Mario H. Castaneda. Molina-Galan, Alejandro, Boban, Mate, Gozalvez, Javier, Coll-Perales, Baldomero, Sahin, Taylan & Kousaridas, Apostolos 2021. A Tutorial on 5G NR V2X Communications. Julkaisussa: IEEE Communications surveys and tutorials 23 (3), 1972–2026. Hakupäivä 5.10.2022.
<https://ieeexplore.ieee.org/ielx7/9739/9520094/09345798.pdf>.
 14. Krutz, Martin 2021. The Art of Benchmarking. Springer Nature.
 15. PassMark Software. Performance test. Hakupäivä 20.11.2022.
<https://www.passmark.com/products/performance-test/index.php>.
 16. Event Machine on ODP 2022. EM-ODP Documentation. Hakupäivä 17.10.2022. [EM-ODP: Main Page \(openeventmachine.github.io\)](https://openeventmachine.github.io).
 17. Perf. Perf dokumentaatio. Hakupäivä 21.10.2022.
https://perf.wiki.kernel.org/index.php/Main_Page.
 18. Alapuranen, Sakari 2015. Performance optimizations for LTE User-Plane L2 software. Oulun yliopisto. Pro gradu -tutkielma. Hakupäivä 24.10.2022.
<http://jultika.oulu.fi/files/nbnfioulu-201505061447.pdf>.
 19. Lauterbach. TRACE32 multicore debugging. Hakupäivä 26.10.2022.
<https://www.lauterbach.com/frames.html?multicore.html>.
 20. Lauterbach. TRACE32 manuaali. Pdf-tiedosto. Hakupäivä 26.10.2022.
https://www2.lauterbach.com/pdf/training_debugger.pdf.
 21. FPGA for beginners. What is JTAG. Hakupäivä 20.11.2022.
<https://fpgabeginner.com/what-is-jtag/>.
 22. Lauterbach. TRACE32 suorituskykyanalysaattorin esittely. Hakupäivä 26.10.2022.
<https://support.lauterbach.com/kb/articles/trace32-performance-analyzer>.

23. Wexler, Steve 2021. The Big Picture: How to Use Data Visualization to Make Better Decisions—Faster. Hakupäivä 3.11.2022. O'Reilly Online Learning. Vaatii käyttöoikeuden.
24. Tilastokeskus 2001. Jakauman esittäminen. Hakupäivä 20.11.2022. https://www.stat.fi/tup/tieto aika/tilaajat/ta_11_01_kuusela.html#:~:text=Yksinkertaisin%20ja%20kenties%20perinteisin%20jakaumakuvi%20on%20histogrammi.%20Histogrammi, kuhunkin%20luokkaan%20sijoittuu%20ja%20n%20C3%A4it%20C3%A4%20vastaavasti%20piirret%20C3%A4%20pylv%20C3%A4s.
25. Alexander, Michael, Kusleika, Dick & Walkenbach, John 2019. Excel 2019 Bible, John Wiley & Sons, Incorporated. Hakupäivä 4.11.2022. ProQuest Ebook Central. Vaatii käyttöoikeuden.
26. Elastic. What is ELK stack? Hakupäivä 5.11.2022. <https://www.elastic.co/what-is/elk-stack>.
27. Grafana. What is Grafana?. Hakupäivä 6.11.2022. <https://grafana.com/oss/grafana/?plcmt=footer>.
28. Splunk. Splunk products. Hakupäivä 9.11.2022. https://www.splunk.com/en_us/products.html.
29. Splunk. The Splunk platform. Hakupäivä 9.11.2022. https://www.splunk.com/en_us/products/platform.html?301=/en_us/software/platform.htm#scalable-index.
30. Splunk. SPL documentation. Hakupäivä 9.11.2022. <https://docs.splunk.com/Splexicon:SPL>.
31. Splunk. Splunk Enterprise. Hakupäivä 9.11.2022. https://www.splunk.com/en_us/products/splunk-enterprise.html.
32. Splunk. Splunk Enterprise features. Hakupäivä 9.11.2022. https://www.splunk.com/en_us/products/splunk-enterprise-features.html.
33. Termipankki. Regular expression. Hakupäivä 12.11.2022. <https://termipankki.fi/tepa/fi/haku/regular%20expression>.
34. Robot Framework. Robot Framework user guide. Hakupäivä 12.11.2022. <https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html>.
35. Robot Framework. Convert To String. Hakupäivä 12.11.2022. <https://robotframework.org/robotframework/latest/libraries/BuiltIn.html#Convert%20To%20String>.

36. Robot Framework. Process-kirjasto. Hakupäivä 12.11.2022.
<https://robotframework.org/robotframework/latest/libraries/Process.html>.
37. Splunk. Set up and use HTTP Event Collector in Splunk Web. Hakupäivä 13.11.2022.
<https://docs.splunk.com/Documentation/Splunk/9.0.2/Data/UsetheHTTPEventCollector>.

LIITTEET

SPL syntaksi liitteen 3 visualisoimiseksi. Liite 1.

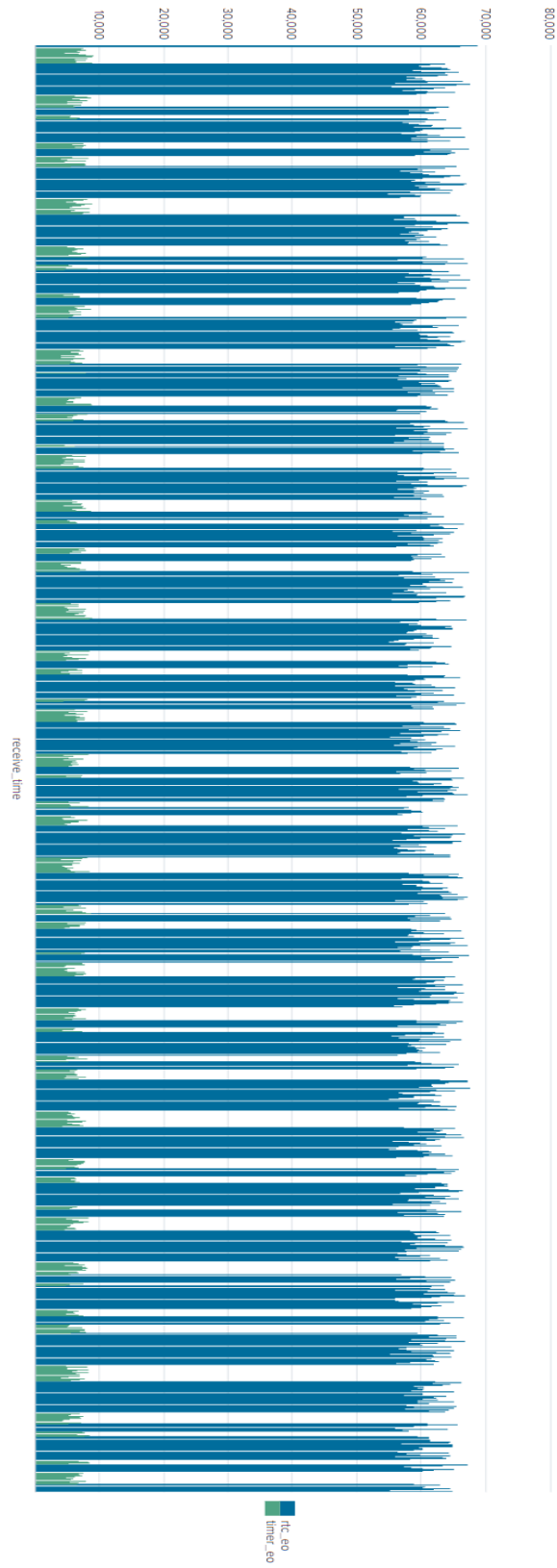
Suoritus aika ajan funktiona. Liite 2.

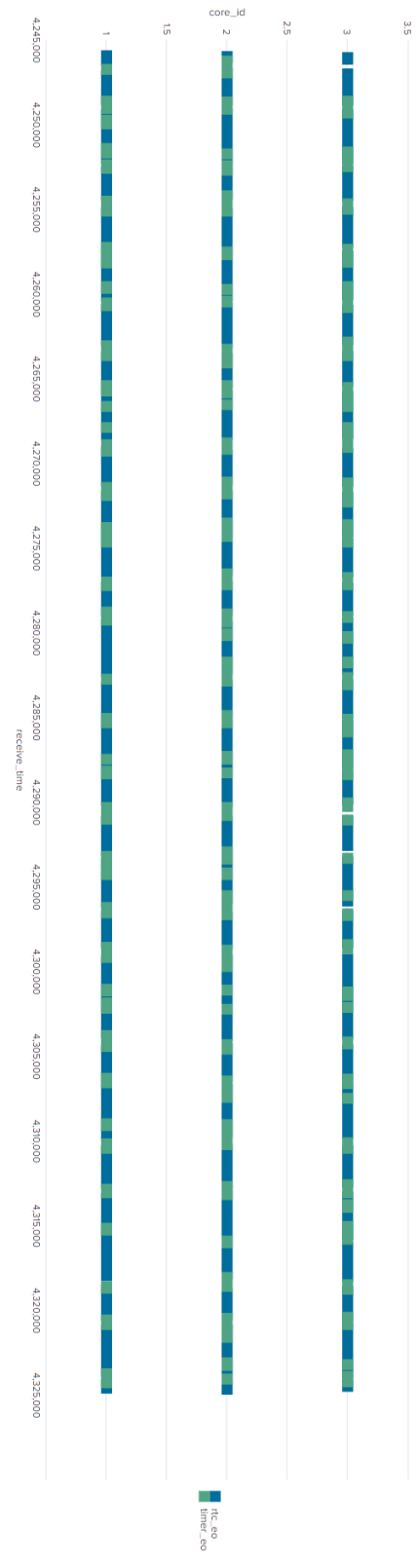
EO tapahtumien allokaatio ytimille ajan funktiona. Liite 3.

Kahden peräkkäisen RTC tapahtuman lukumäärä näiden lähetysaikavälin funktiona. Liite 4.

EO tapahtumien lukumäärä suoritusajan funktiona. Liite 5.

```
index="index_for_test_data" source=test_data_for_thesis.csv
| rex field=_raw
"(?ms)^(?P<receive_time>[^,]+),(?P<execution_time>[^,]+),(?P<event_id>[^,]+),(?P<core_id>[^,]+),(?P<eo_name>[^,]+),(?P<queue_name>[^,]+),(?P<payload_size>[^,]+)"
| search "rtc_eo"
| sort receive_time
| streamstats range(receive_time) window=2 as time_diff
| search time_diff > 0
| stats count(receive_time) AS amount_of_consecutive_rtc_eos BY time_diff
| chart values(amount_of_consecutive_rtc_eos) by time_diff span=1
```





KAHDEN PERÄKKÄISEN RTC-TAPAHTUMAN LUKUMÄÄRÄ NÄIDEN LÄHETYSAIKAVÄLIN
FUNKTIONA

LIITE 4

