Niklas Granqvist

Tietoallas ja tietovarasto massadatan hallinnassa

Tietotekniikan kandidaatintutkielma

5. toukokuuta 2022

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Niklas Granqvist

Yhteystiedot: granqnks@student.jyu.fi

Ohjaaja: Timo Tiihonen

Työn nimi: Tietoallas ja tietovarasto massadatan hallinnassa

Title in English: Data lake and data warehouse for big data management

Työ: Kandidaatintutkielma

Opintosuunta: Tietotekniikka

Sivumäärä: 19+0

Tiivistelmä: Nykypäivänä tietoaltaat ovat nousseet suureen suosioon yritysten datan käsittelyssä. Massiivisten datamäärien hallinta ja käsittely niin kutsutun massadatan aikakauden aikana on muodostanut haasteita, joihin perinteiset ratkaisut eivät ole soveltuneet. Seurauksena tälle on kehitetty tietoaltaat, jotka ratkaisevat ongelman heterogeenisen tiedon varastoinnissa ja tuovat uusia mahdollisuuksia tämän käsittelyyn, mutta poikkeavat vanhempien ratkaisujen käytettävyydestä.

Avainsanat: Kandidaatintutkielmat, Tietoallas, Tietovarasto, ETL

Abstract: In today's world the use of Data lakes have risen in popularity with enterprices. Handling and processing massive amounts of data during the Big Data era have created challenges, that traditional warehouses are not able to solve. Data lakes were created as a solution to solve problems with storing and prossesing heterogeneous data, but differ in the usage of the older solutions.

Keywords: Bachelor's Theses, Data lake, Data warehouse, ETL

Kuviot

Sisällys

1 Johdanto

Nykypäivänä liikkuvan tiedon määrä kasvaa jatkuvasti jokaisella toimialla. Tämä valtava tiedon määrä on luonut käsitteen massadata (Big Data), joka tarkoittaa hyvin monimuotoista ja laajasti saatavilla olevaa tietoa (Emmanuel ja Stanier 2016). Yritykset keräävät tietoa monista erilaisista lähteistä, kuten asiakasjärjestelmistä, sosiaalisesta mediasta ja IoT-laitteista. Tämän massiivisen ja monipuolisen tietomäärän prosessointi ja säilöminen on luonut omat ongelmansa tekniikassa, ja samalla tuottanut erilaisia ratkaisuja mahdollistamaan tiedon käytön. Tässä tutkielmassa käsitellään kirjallisuuskatsauksena tietoaltaita: etuihin massadatan käsittelyssä tietovarastoihin nähden, sekä haasteisiin, joita tietoaltaan rakenteet ja toimintamalli tuottavat vertailemalla näitä kahta keskenään.

Tutkielman toisessa luvussa käsitellään teknisestä näkökulmasta tietoaltaita, tietovarastoja, sekä näiden keskeisiä eroavaisuuksia. Olennaisia tekijöitä tässä ovat arkkitehtuuri ratkaisujen takana sekä miten tieto käsitellään, mutta tutkielmassa sivutaan myös tiedon säilöntää, sekä valmiin tiedon tuottamista loppukäyttäjälle. Luvussa tuodaan esiin molempien ratkaisujen keskeinen toimintatapa, jotta on mahdollista hahmottaa eroavaisuuden niiden kesken.

Kolmannessa luvussa keskitytään teknisestä näkökulmasta tietoaltaan etuihin tiedon käsittelyssä sekä varastoinnissa tietovarastoon nähden. Tämän jälkeen neljännessä luvussa perehdytään eroavaisuuksiin käyttäen reaalimaailman case-esimerkkiä. Tämän avulla pyritään havainnollistamaan ratkaisun merkittävyyttä, ja miksi tietoaltaat ovat kehitetty. Luvussa käsitellään myös tietoaltaiden luomia haasteita ja missä määrin tietovarasto on edelleen toimivampi ratkaisu.

2 Järjestelmien arkkitehtuurit ja toimintamallit

Tietoaltaat ja tietovarastot ovat nousseet suurten yritysten suosioon massadatan hallinnassa, sekä varastoinnissa. Koska data määrittelee nykypäivänä yrityksen toimintaa hyvin vahvasti, on tämän säilönnän ja käytön oltava mahdollisimman tehokasta, sekä kustannusystävällistä. Molemmat edellä mainituista tarjoavat hieman toisistaan eroavat ratkaisut, joista löytyy omat niin hyvät kuin huonot puolensa.

2.1 Tietovarastot

Tietovarasto on tietoallasta hieman vanhempi käsite, joka on ollut suuressa suosiossa jo pitkään. Se toimii varastona esimerkiksi päätöksenteon tukijärjestelmiä (Decission Support System, DSS), johdon tietojärjestelmiä (Executive Information System, EIS), reaaliaikaista analytiikkaa ja tiedonlouhintaa varten (Ariyachandra ja Watson 2010). Tietovarastolle löytyy useampia erilaisia arkkitehtuurillisia ratkaisuja, jotka ovat kehittyneet ajan saatossa. Näitä ovat muun muassa: yksikerroksinen -, kaksikerroksinen -, kolmikerroksinen -, itsenäinen data mart -, data mart bus -, hub-and-spoke -, sekä Enterprise Data Warehouse, jonka kehitti Bill Inmon koko yrityksen tasolla toimivaksi tietovarastoksi (Ariyachandra ja Watson 2010). Tämä arkkitehtuuri tunnetaan myös nimellä keskitetty arkkitehtuuri (Blažić, Poščić ja Jakšić 2017) ja sen ratkaisun tarkoituksena on eliminoida vanhojen tietokantojen epävakaus. Tässä tutkimuksessa keskitytään eri arkkitehtuureista erityisesti viimeisimpänä mainittuun.

Tietovaraston tarkoituksena on mahdollistaa joustava ja skaalautuva ratkaisu massadatan varastoinnille, sekä sen käytölle (Gardner 1998). Oikean arkkitehtuurillisen ratkaisun valitseminen on kuitenkin ehdottoman tärkeää tietovaraston luonnin alkuvaiheessa, sillä väärät ratkaisut voivat johtaa skaalautumisen sekä tehokkuuden puutteeseen, kuten mainitaan Ariyachandra ja Watson (2010) tutkimuksessa.

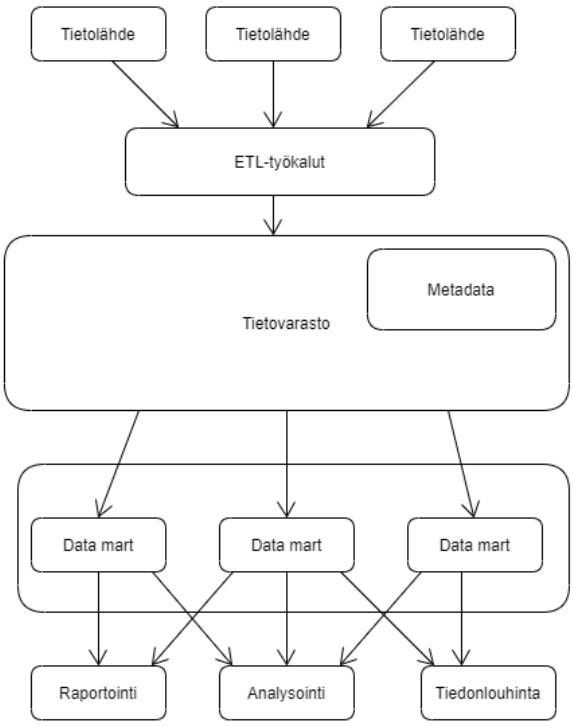
Tietovaraston yleinen toiminta rakentuu siten, että tietolähteelle suoritetaan ETL-käsittely (Extract, Transform and Load), jonka jälkeen prosessoitu ja yhdenmukaisessa muodossa oleva tieto tallennetaan tietovarastoon tarkasti strukturoituna (Ariyachandra ja Watson 2010). Tämän jälkeen prosessoitu tieto siirretään eteenpäin, minkä jälkeen tieto on käytettävis-

sä loppukäyttäjillä esimerkiksi analyysejä ja raportteja varten. Esimerkkejä tietovarastoista on Oracle database, relaatiotietokannat tai yksinkertainen pilvipohjainen ratkaisu, Amazon Redshift (Gupta ym. 2015).

ETL on yksi suurimpia osa-alueita tietovaraston luomisessa, sillä kyseessä on kompleksi, aikaa vievä prosessi, joka kattaa suuren osan implementoinnin kustannuksista ja resursseista (El-Sappagh, Hendawi ja El Bastawissy 2011). Pääperiaatteena ETL:ssä on kyse tiedon hausta tietolähteestä, sen muokkaamisesta haluttuun muotoon, jonka jälkeen tieto ladataan homogeeniseen ympäristöön (Gour ym. 2010), tässä tapauksessa tietovarastoon. Tiedon lähteet voivat olla esimerkiksi perinteisiä relaatiotietokantoja, mutta myös muita rakenteellisia tietolähteitä kuten XML-tiedostoja tai NoSQL-tietokantoja. Tärkeää tässä on, että prosessoidun datan skeema tulee vastata tarkasti tietovaraston skeemaa. On tärkeää huomioida, että tämä prosessi ei tapahdu tietovaraston sisällä. ETL on osa-alueena jatkuvasti kehitettävä, sillä ajan saatossa tietolähteitä on mahdollista lisätä, tai niiden tuottama tieto voi muuttua. Tämän johdosta ETL-prosessin kehittäminen vaatii niin data-analyytikkoja, tietokantasuunnittelijoita, kuin ohjelmistokehittäjiäkin (El-Sappagh, Hendawi ja El Bastawissy 2011).

2.1.1 Tietovaraston komponentteja

Ennen kuin paneudutaan yritystason tietovaraston toimintaan, on hyvä tunnistaa sen pohjalla oleva arkkitehtuuri. Lähimpänä loppukäyttäjää arkkitehtuurissa sijaitsee data mart. Käsitteenä se tarkoittaa osaa tietovarastosta, joka tarjoaa käyttökelpoisen tiedon käyttäjälle. Data mart saa tiedon keskitetystä tietovarastosta ja ovat suunniteltu yksilöllisesti tiettyä yksittäistä käyttötarkoitusta varten (Moody ja Kortink 2000). Hub puolestaan koostuu data marteista, joita tässä yhteydessä kutsutaan spokeiksi. Hub-and-spoke arkkitehtuuri koostuu tietolähteistä, sovitetusta tiedosta ja data marteista. Normalisoitu, atominen, tietolähteistä saatu tieto säilötään sovituskerrokseen, joka puolestaan syöttää tiedon data marteille. Tämä arkkitehtuurillinen ratkaisu keskittyy nimenomaan skaalautuvuuteen ja suurten tietomäärien hakuun.



Kuvio 1. Yksinkertaistettu esimerkkikuva yritystason tietovaraston mahdollisesta arkkitehtuurista

2.1.2 Yritystason tietovarasto

Yritystason tietovaraston (Enterprise Data Warehouse, EDW) arkkitehtuuri (Kuvio 1) on hyvin pitkälti implementoitu hub-and-spoke arkkitehtuurista (Blažić, Poščić ja Jakšić 2017). EDW pitää sisällään kaksi erilaista varastoa: operatiivisen varaston ja raportointivarastot (Rifaie ym. 2008). Operatiivinen varasto on hyvin yksityiskohtainen, vuorovaikutuskeskeinen, tarkka hakuhetkellä ja suunnattu tuotantokäyttöön. EDW-tietovarasto on enemmän analyyttinen ja pitää sisällään historioivaa tietoa, joten se on keskittynyt enemmän raportointiin. Tämä ratkaisu mahdollistaa tuotantoympäristössä historiallisen tiedon minimoinnin, jonka avulla on mahdollista optimoida tietovaraston kustannuksia.

EDW-datamallin rakennus sisältää useita eri vaiheita, jotka on suunniteltava ja toteutettava hyvin tarkasti tietovaraston tehokkaan toiminnan mahdollistamiseksi (Fang 2015). Oikeanlaisen datamallin määritteleminen on hyvin keskeisessä osassa EDW:n suunnittelua, rakennusta, sekä implementointia (Rifaie ym. 2008). Klassisesta datamallinnuksesta kehittyneiden tekniikoiden käyttöä painotetaan uuden datamallin rakentamisessa. Tämä johtuu siitä, että klassisen datamallinnuksen tekniikoissa ei huomioida operatiivisen ja analyyttisen ympäristöjen eroavaisuuksia, jotka ovat keskeisessä osassa EDW:n arkkitehtuuria. Yritys-datamalli muodostaa pohjan tietoarkkitehtuurille ja määrittelee rakenteellisen datan vaatimukset analyyttiselle tiedolle. Vaiheita datamallin rakentamista varten ovat muun muassa: lähdejärjestelmän sisäisen operatiivisen datan suodatus varastoivasta osasta, samankaltaisen datan yhdisteleminen eri tietokantataulujen välillä, johdetun tiedon lisäys, sekä aikaelementtien lisäys avainrakenteisiin (Rifaie ym. 2008).

2.2 Tietoaltaat

Nykypäivänä tietoaltaiden suosio on noussut yritysten keskuudessa täyttämään tiedonhallintaan liittyvät tarpeet ja vaatimukset (Giebler ym. 2019). Monesta eri lähteestä saatu tieto voidaan säilöä yhteen paikkaan ilman esiprosessointia tai tiedon transformoimista, joka nopeuttaa sen käyttöä. Tietoaltaat tuovat kuitenkin omat haasteensa tiedon säilömiseen.

Tietoaltaat ovat määritelmän mukaan suuria raa'an heterogeenisen datan varastoja, jotka kykenevät varastoimaan tiedon sen alkuperäisessä formaatissa (Sawadogo ja Darmont 2021). Säilöttävä tieto voi olla siis strukturoimatonta, semi-strukturoitua tai strukturoitua, toisin kuin tietovaraston kohdalla. Tietoaltaan arkkitehtuuri yhdistetään pitkälti Apache Hadoop-ekosysteemiin, jota on myöhemmin implementoitu esimerkiksi pilvipalveluiden luomissa tietoallas-arkkitehtuureissa (Sawadogo ja Darmont 2021). Toisin kuin tietovarastoissa, joissa käytetään pitkälti RDBMS:ää (Relational Database Management System) tiedon varastointiin, on tietoaltaissa käytössä yhdistelmä, joka koostuu HDFS:tä (Hadoop Distributed File System), sekä NoSQL- ja/tai RDBMS-tietokannoista. Tämä johtuu datan monimuotoisuudesta, sillä RDBMS-tietokannat kykenevät ainoastaan säilömään strukturoitua dataa, mutta ei strukturoimatonta.

Tietoaltaille mahdollisia tiedon lähteitä on lukuisia, joista saadaan hyvin laajasti erilaista dataa. Hyviä esimerkkejä ovat esimerkiksi videot, kuvat, tekstidokumentit tai sähköpostit, jotka edustavat strukturoimatonta tietoa. Semi-strukturoitua puolestaan voi olla csv-tiedostot, lokitiedot tai json-tiedostot. Strukturoitua dataa on RDBMS-sopivaa tietoa.

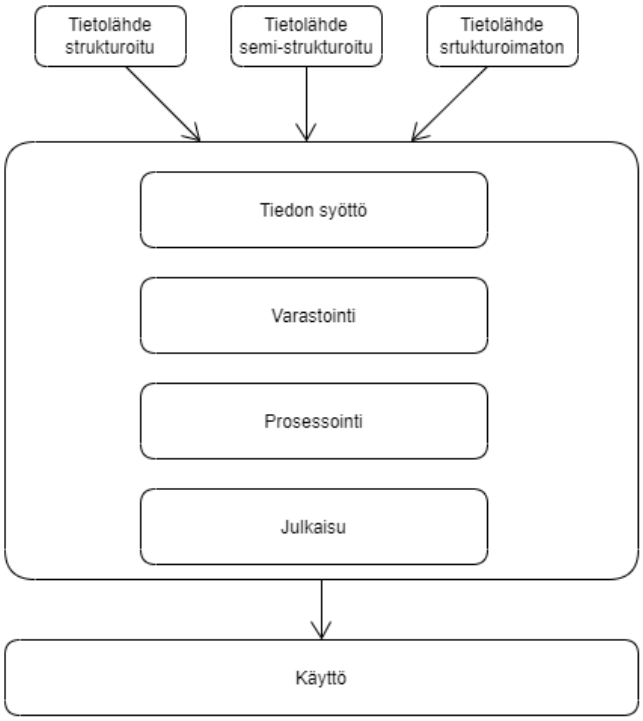
2.2.1 Prosessointi

Tietoaltaalle ei ole määritelty tarkkaa arkkitehtuuria, mutta pääsääntöisesti se voidaan jakaa omiin osa-alueisiinsa, joita ovat tiedon syöttö, varastointi ja prosessointi (Kuvio 2). Nämä kaikki vaiheet kuuluvat tietoaltaan kokonaisuuteen, jossa prosessointi on vastuussa tiedon valmistelusta käyttöä varten. Prosessointi koostuu tyypillisesti kolmesta eri vaiheesta: valmistelusta, analytiikasta, sekä tuotetun tiedon käytön mahdollistamisesta (Mathis 2017). Itse tiedon prosessointiin käytetään erilaisia työkaluja, joista suosituimpia ovat esimerkiksi Apache Spark, MapReduce, Apache Flink ja Apache Storm.

Tietoaltaiden yksi merkittävimmistä tekijöistä on nopean ja reaaliaikaisen datan prosessoinnin mahdollistaminen. Tämä on yksi suuri ero tietovarastoihin verrattuna. Tiedon säilöminen raa'assa alkuperäisessä muodossa mahdollistaa sen välittömän saatavuuden prosessointia varten (Miloslavskaya ja Tolstoy 2016). Valmiissa ratkaisussa, jossa prosessointi koostuu yksittäisiin tarpeisiin rakennetuista kevyistä prosesseista, minimoidaan viive tietolähteen ja prosessoidun tiedon välillä, toisin kuin tietovarastoissa, jossa on tehtävä kallis esiprosessointi ja transformointi ennen tiedon säilömistä. Tämä tarkoittaa samalla myös sitä, että tieto voidaan säilöä ilman skeemaa, jolloin sitä ei voida hyödyntää ilman prosessointia. Prosessointi käyttää schema-on-read käsittelyä, jossa tiedon skeema määritellään vasta prosessoinnin yhteydessä. Tämä tunnetaan myös nimellä "late bidding". Skeeman määritys voi tapahtua esimerkiksi ETL-prosessin yhteydessä. Nimensä mukaan prosessissa otetaan tietoa (Extract), prosessoidaan se (Transform), ja tuotetaan käytettäväksi (Load)(Fang 2015).

Koska säilötty tieto on monimuotoista ja voi sijaita erilaisissa tietokannoissa, ei tavanomaisten kyselyiden käyttö ole välttämättä mahdollista. Klassinen kyselykieli, kuten SQL, toimii RDBMS-tietokantojen kanssa, mutta ei varastojen, josta löytyy strukturoimatonta tai semi-strukturoitua dataa. Semi-strukturoitua tietoa varten hyödynnetään esimerkiksi SparkSQL tai

SQL++ kyselyitä (Sawadogo ja Darmont 2021).



Kuvio 2. Yksinkertaistettu esimerkkikuva tietoaltaan mahdollisesta arkkitehtuurista

Rakenteen tai sen puuttumisen vuoksi tiedon prosessointi ei ole yksinkertaista tietoaltaissa. Tätä tehtävää kuvataan vahvasti data-analyytikoille suunnatuksi, sillä prosessointi pohjautuu pitkälti analytiikan rakentamiseen ohjelmoimalla (Fang 2015). Prosessointi voi käytännössä tapahtua joko reaaliajassa (tai lähellä sitä), tai tuotantoerissä, jota kutsutaan batch-prosessoinniksi. Suosituista työkaluista esimerkiksi MapRrduce on suunnattu suurelle, tuotantoerissä prosessoitavalle datalle, mutta ei sovellu reaaliajassa tapahtuvaan prosessointiin, toisin kuin Spark, Flink tai Storm. Molempia prosessointi tapoja on mahdollista kuitenkin suorittaa samanaikaisesti, mutta eri tarkoituksiin. Tiedon prosessointia koordinoi "workflow", joka koostuu yksittäisiin tehtäviin tarkoitetuista pienemmistä prosesseista, minkä käytännössä voi ajatella yhtenä työyksikkönä. Prosessoinnin automaatio, jonka workflow mahdollistaa, on tärkeää, sillä uutta tietoa saadaan jatkuvasti ja sen käyttö ei ole mahdollista ilman prosessointia (Mathis 2017).

3 Tekninen vertailu

Tietoaltailla ja yritystason tietovarastoilla on toisiinsa nähden samankaltaisuuksia ja eroavaisuuksia, jotka voidaan nähdä niin hyötyinä kuin haittoinakin, kun näitä verrataan keskenään. Tässä pääpainona pidetään tietoaltaan tuomia hyötyjä yritystason tietovarastoihin nähden, keskittyen teknisestä näkökulmasta tiedon varastointiin sekä sen käsittelyyn, sillä nämä ovat keskisimpiä osa-alueita molempien toiminnassa. On myös hyvä huomioida millaisia haasteita tietoaltaat voivat tuoda tietovarastoihin nähden.

3.1 Tiedon säilöminen

Tietoaltaiden sekä tietovarastojen tarkoituksena on massadatan varastointi, mutta tätä toteutetaan hyvin erilaisilla tavoilla. Kuten aikaisemmin on mainittu, tietoaltaat kykenevät varastoimaan heterogeenistä tietoa sen alkuperäisessä muodossa (Sawadogo ja Darmont 2021), kun taas tietovarastot varastoivat ainoastaan homogeenistä ja tarkasti strukturoitua dataa (Ariyachandra ja Watson 2010). Tietoaltaan etuna varastoinnissa tulee helppous tietovarastoon verrattuna, sillä kaikki kerätty tieto sopii varastoitavaksi nopeasti sellaisenaan, kun taas tietovaraston kohdalla viivettä luo vaadittava tiedon esiprosessointi ennen varastointia (Liu, Isah ja Zulkernine 2020). Helposti säilötyn tiedon ansiosta tietoaltaat sopivat myös reaaliaikaisen tiedon tuottamiseen, toisin kuin tietovarastot, mikä on suuri etu datakriittisessä päätöksenteossa ja toiminnassa (S Prakash 2020). Tietovarasto vaatii erittäin huolellisen suunnittelun arkkitehtuurin ja datamallin ennen sen käyttöönottoa. Prosessina tämä voi olla pitkäkestoinen, sillä suunnittelu ja rakennus riippuvat monista tekijöistä (Ariyachandra ja Watson 2010), kun taas tietoallas on huomattavasti helpompi, nopeampi, sekä kustannustehokkaampi kokonaisuus ottaa käyttöön.

On kuitenkin hyvä huomioida, että kaiken kerätyn tiedon säilöminen tuo omat haasteensa tietoaltaisiin. Tietosuo (Data Swamp) on käsite, jolla tarkoitetaan tietoallasta, jonne kertyy massiivisia määriä tietoa, josta läheskään kaikkea ei kuitenkaan käytetä (Hai, Geisler ja Quix 2016). Tämä voi hankaloittaa oikean tiedon käyttöä. Tietovaraston etuna puolestaan on tarkasti valikoitu tieto, jota on helppo käyttää suoraan varastosta.

3.2 Prosessointi

Tietoaltaiden ja -varastojen välillä prosessointi on toteutettu eri tavoilla, ja nämä vaiheet löytyvät eri osista arkkitehtuuria. Keskinen ero on se, että tietoallas pitää sisällään prosessoinnin (Ravat ja Zhao 2019), kun taas tietovaraston tapauksessa prosessointi on sen ulkopuolella (Ariyachandra ja Watson 2010). Tietoaltaan osalta suuren hyödyn tuo tiedon prosessointi vasta varastoinnin jälkeen. Tietoaltaan schema-on-read mahdollistaa tiedon reaaliaikaisen, sekä batch-prosessoinnin, käyttäen raakaa dataa, kun tietovaraston shema-on-write prosessointi puolestaan suoritetaan pitkälti ETL-toimenpiteenä, batch-muodossa eikä reaaliajassa (Sawadogo ja Darmont 2021). Molemmissa tapauksissa kuitenkin vaaditaan huolellista suunnittelua, ja työ on pitkälti suunnattu data-analytiikoille. Tietovarastossa prosessointi on kuitenkin oltava tehtynä ennen varaston käyttöönottoa, sillä prosessoidun tiedon on vastattava tarkasti varastossa olevaa datamallia, minkä takia tämä vaihe voi olla hidasta kehittää. Tietoaltaissa edun tuo se, että prosessointi voidaan rakentaa varastoinnin jälkeen ja sitä pystytään vapaasti muokkaamaan tarpeiden mukaan.

4 Tietoaltaan hyödyt käytännössä

Lähtökohtana tietoaltaille on toiminut tietovarastojen rajoitukset monimuotoisen tiedon lisääntyessä (Giebler ym. 2019). Laajalti katsoen tietoallas tarjoaa tehokkaamman, monipuolisemman ja kustannusystävällisemmän ratkaisun tietovarastoihin verrattuna. Näihin yksityiskohtiin on hyvä perehtyä, sillä oikean ratkaisun tekeminen ei välttämättä ole yksiselitteistä (Ariyachandra ja Watson 2010).

4.1 Case: lennonjohto

Tutkimuksessa, jonka ovat tehneet Raju, Mital ja Finkelsztein (2018), käsitellään tietoaltaan roolia lennonjohdon työkalujen näkökulmasta. Tutkimuksessa korostetaan tietoaltaan merkitystä ratkaisuna, sillä tietolähteinä massiiviselle määrälle tietoa toimivat esimerkiksi säätiedot, ilmailudata ja ympäristödata, joista saadaan strukturoitua, semi-strukturoitua, kuin strukturoimatonta dataa. Tästä johtuen tyypilliset strukturoidun tiedon varastointimenetelmät, kuten tietovarasto, eivät sovellu käyttötarkoitukseen. Tietovaraston tyypillinen ETL-esiprosessointi ei ole kustannusoptimoitu ratkaisu, vaikka tuloksena saataisiinkin valmista, strukturoitua tietoa, joka olisi helposti käytettävissä. Tarve tietoaltaalle korostuu teknologian valinnassa, sillä se tarjoaa joustavuutta ja responsiivista data-analytiikkaa. Tiedon prosessointi vasta jälkikäteen on tärkeä ominaisuus järjestelmää kehittäessä, sillä sen avulla mahdollistetaan analyytikkojen kehitystyö ilman riippuvuutta kohdejärjestelmästä. Tietovaraston kohdalla tämä ei olisi mahdollista, sillä prosessoidun tiedon tulee vastata tarkasti varaston rakennetta. Tässä tapauksessa tietoaltaan toteutus arkkitehtuurillisesta näkökulmasta on toteutettu seuraavanlaisesti: kaikki tieto ladataan alkuperäisessä formaatissa raa'alle vyöhykkeelle, josta se prosessoidaan ja muokataan prosessointivyöhykkeelle, jonka jälkeen se viimeistellään analysoimalla tieto. Tämän jälkeen se on käytettävissä analyysejä ja visualisointeja varten. (Raju, Mital ja Finkelsztein 2018).

Raju, Mital ja Finkelsztein (2018) tutkimuksessa tuodaan esille tietoaltaan mahdollistama analytiikan tuottaminen. Analyytikot kehittivät analyysin, jonka avulla pystyttiin laskemaan lentokoneen odotuskuvion vaikutusta polttoaineen kulutukseen. Tässä merkittävämmässä

osassa oli tietolähteet, Swim Flight Data Publication Service (SFDPS) lentoreitille ja Aviation Environment Design Tool (AEDT) polttoaineen kulutustiedolle, joka koostetaan reititin, rungon ja moottoreiden tiedoista. Tietoaltaanmerkitys korostuu tässä, sillä molemmat tietolähteet toimivat täysin itsenäisesti omissa ympäristöissään. Perinteinen tiedon varastointi, kuten tietovaraston käyttö, olisi luonut näiden kahden tietolähteen tuottaman tiedon analyysistä hyvin kompleksin tehtävän, mutta koska tieto saatiin tallennettua yhteen paikkaan, oli analyysin tuottaminen yksinkertaista (Raju, Mital ja Finkelsztein 2018).

4.2 Tietoaltaiden haasteet

Tietoaltaat eivät kuitenkaan ole kaikessa määrin parempi ja helpompi ratkaisu, kuin tietovarastot. Tämä tulee hyvin ilmi varsinkin loppukäyttäjien kohdalla tiedon käytössä. Tietovarasto on ennalta suunniteltu tuottamaan kaikki tarvittava tieto, mikä mahdollistaa sen, että loppukäyttäjän on mahdollista päästä tarvittavaan tietoon käsiksi esimerkiksi SQL-kyselyillä. Tietoaltaissa puolestaan tiedon saanti on monimutkaisempaa, sillä suora kysely ei käytännössä ole mahdollista, ja tätä varten on käytettävä siihen suunniteltuja työkaluja, kuten SparkSQL:ää (Sawadogo ja Darmont 2021). Tietoaltaissa olevan tiedon hyödyntäminen on pitkälti data-analyytikoiden tehtävä, kun puolestaan tietovarastot ovat hyödynnettävissä tiedon loppukäyttäjillä (Fang 2015). Tietovarasto soveltuu hyvin esimerkiksi finanssialan järjestelmiin, sillä yhden yrityksen sisällä analyysejä tekevien käyttäjien määrä voi olla massiivinen.

5 Yhteenveto

Tutkimuksesta voidaan todeta tietoaltaiden tuomat hyödyt tietovarastoihin verrattuna, ja miksi ne ovat nousseet suureen suosioon nykypäivän massadatan hallinnassa. Massiivisen heterogeenisen tietomäärän tehokas ja helppo varastoiminen ja sen monipuolinen käsittely on mahdollistanut kyseisen tiedon hyödyntämisen, mikä on ollut aikaisemmin esimerkiksi tietovaraston kohdalla hyvin haasteellista. Altaan sisällä tapahtuva prosessointi nopeuttaa huomattavasti tiedon varastointia tietovarastoon verrattuna, kun tätä ei suoriteta ennen tiedon säilömistä. Vaikka tietovarastot soveltuvat edelleen nykypäivään omassa tehtävässään, tuovat tietoaltaat paljon uusia mahdollisuuksia tehokkaamman, nopeamman, sekä kustannustehokkaamman tiedon käsittelyssä.

Teknisestä vertailusta, sekä case-esimerkissä korostuu kaksi asiaa: mistä tarve tietoaltaille on syntynyt, sekä miten tietoaltaat on rakennettu ratkaisemaan massadatan hallintaan liittyvät ongelmat. Keskeiset erot tiedon käsittelyssä ja sen säilömisessä arkkitehtuurillisesta näkökulmasta korostavat erityisesti tietoaltaan hyötyjä massadatan hallinnassa. Case-esimerkin kohdalla korostuu tietoaltaan heterogeenisen varastoinnin merkitys, sekä prosessoinnin sijainnin merkitys tietoaltaan arkkitehtuurissa analyysiä rakentaessa.

Vaikka tässä tutkielmassa on verrattu tietoallasta ja -varastoa keskenään, keskittyen molempien hyötyihin ja haittoihin toisiinsa nähden, eivät nämä ole toisiaan poissulkevia tekijöitä. Fang (2015) esittelee tutkimuksessaan, miten näitä kahta on mahdollista yhdistää tiedon varastoinnin ja sen käytön optimoimiseksi. Näin voitaisiin mahdollisesti tuoda molemmista parhaat puolet esiin, tietoaltaan kyky varastoida monimuotoista dataa, sekä tietovaraston kyky välittää tietoa loppukäyttäjille.

Lähteet

Ariyachandra, Thilini, ja Hugh Watson. 2010. “Key organizational factors in data warehouse architecture selection”. *Decision Support Systems* 49 (2): 200–212. ISSN: 0167-9236. https://doi.org/https://doi.org/10.1016/j.dss.2010.02.006. https://www.sciencedirect.com/science/article/pii/S0167923610000436.

Blažić, G., P. Poščić ja D. Jakšić. 2017. “Data warehouse architecture classification”. Teoksessa *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO),* 1491–1495. https://doi.org/10.23919/MIPRO.2017.7973657.

Emmanuel, Isitor, ja Clare Stanier. 2016. “Defining Big Data”. Teoksessa *Proceedings of the International Conference on Big Data and Advanced Wireless Technologies.* BDAW ’16. Blagoevgrad, Bulgaria: Association for Computing Machinery. ISBN: 9781450347792. https://doi.org/10.1145/3010089.3010090.

Fang, Huang. 2015. “Managing data lakes in big data era: What’s a data lake and why has it became popular in data management ecosystem”. Teoksessa *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER),* 820–824. https://doi.org/10.1109/CYBER.2015.7288049.

Gardner, Stephen R. 1998. “Building the Data Warehouse”. *Commun. ACM* (New York, NY, USA) 41, numero 9 (syyskuu): 52–60. ISSN: 0001-0782. https://doi.org/10.1145/285070.285080.

Giebler, Corinna, Christoph Gröger, Eva Hoos, Holger Schwarz ja Bernhard Mitschang. 2019. “Leveraging the Data Lake: Current State and Challenges”. Teoksessa *Big Data Analytics and Knowledge Discovery,* toimittanut Carlos Ordonez, Il-Yeol Song, Gabriele Anderst-Kotsis, A Min Tjoa ja Ismail Khalil, 179–188. Cham: Springer International Publishing. ISBN: 978-3-030-27520-4.

Gour, Vishal, SS Sarangdevot, Govind Singh Tanwar ja Anand Sharma. 2010. “Improve performance of extract, transform and load (ETL) in data warehouse”. *International Journal on Computer Science and Engineering* 2 (3): 786–789.

Gupta, Anurag, Deepak Agarwal, Derek Tan, Jakub Kulesza, Rahul Pathak, Stefano Stefani ja Vidhya Srinivasan. 2015. “Amazon Redshift and the Case for Simpler Data Warehouses”. Teoksessa *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data,* 1917–1923. SIGMOD ’15. Melbourne, Victoria, Australia: Association for Computing Machinery. ISBN: 9781450327589. https://doi.org/10.1145/2723372.2742795.

Hai, Rihan, Sandra Geisler ja Christoph Quix. 2016. “Constance: An Intelligent Data Lake System”. Teoksessa *Proceedings of the 2016 International Conference on Management of Data,* 2097–2100. SIGMOD ’16. San Francisco, California, USA: Association for Computing Machinery. ISBN: 9781450335317. https://doi.org/10.1145/2882903.2899389. https://doi.org/10.1145/2882903.2899389.

Liu, Ruoran, Haruna Isah ja Farhana Zulkernine. 2020. “A Big Data Lake for Multilevel Streaming Analytics”. Teoksessa *2020 1st International Conference on Big Data Analytics and Practices (IBDAP),* 1–6. https://doi.org/10.1109/IBDAP50342.2020.9245460.

Mathis, Christian. 2017. “Data Lakes”. *Datenbank-Spektrum* 17, numero 3 (marraskuu): 289–293. ISSN: 1610-1995. https://doi.org/10.1007/s13222-017-0272-7. https://doi.org/10.1007/s13222-017-0272-7.

Miloslavskaya, Natalia, ja Alexander Tolstoy. 2016. “Big Data, Fast Data and Data Lake Concepts”. *Procedia Computer Science* 88 (tammikuu): 300–305. ISSN: 1877-0509. https://www.sciencedirect.com/science/article/pii/S1877050916316957.

Moody, Daniel L, ja Mark AR Kortink. 2000. “From enterprise models to dimensional models: a methodology for data warehouse and data mart design.” Teoksessa *DMDW,* 5.

Raju, Ramakrishna, Rohit Mital ja Daniel Finkelsztein. 2018. “Data Lake Architecture for Air Traffic Management”. Teoksessa *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC),* 1–6. https://doi.org/10.1109/DASC.2018.8569361.

Ravat, Franck, ja Yan Zhao. 2019. “Data Lakes: Trends and Perspectives”. Teoksessa *Database and Expert Systems Applications,* toimittanut Sven Hartmann, Josef Küng, Sharma Chakravarthy, Gabriele Anderst-Kotsis, A Min Tjoa ja Ismail Khalil, 304–313. Cham: Springer International Publishing. ISBN: 978-3-030-27615-7.

Rifaie, Mohammad, Keivan Kianmehr, Reda Alhajj ja Mick J. Ridley. 2008. “Data warehouse architecture and design”. Teoksessa *2008 IEEE International Conference on Information Reuse and Integration,* 58–63. https://doi.org/10.1109/IRI.2008.4583005.

S Prakash, Sidharth. 2020. “Evolution of Data Warehouses to Data Lakes for Enterprise Business Intelligence”. *International Journal of Innovative Research in Computer and Communication Engineering* 8 (huhtikuu): 1038–1042.

El-Sappagh, Shaker H Ali, Abdeltawab M Ahmed Hendawi ja Ali Hamed El Bastawissy. 2011. “A proposed model for data warehouse ETL processes”. *Journal of King Saud University-Computer and Information Sciences* 23 (2): 91–104.

Sawadogo, Pegdwendé, ja Jérôme Darmont. 2021. “On data lake architectures and metadata management”. *Journal of Intelligent Information Systems* 56, numero 1 (helmikuu): 97–120. ISSN: 1573-7675. https://doi.org/10.1007/s10844-020-00608-7. https://doi.org/10.1007/s10844-020-00608-7.