Haaga-Helia
ammattikorkeakoulu Oy

Big data ja Apache Hadoop

Kirsi Honkasalo

Opinnäytetyö
Tietojenkäsittelyn koulutusohjelma
2016

Haaga-Helia
ammattikorkeakoulu Oy

Tiivistelmä

Tekijä(t)
Kirsi Honkasalo

Koulutusohjelma
Tietojenkäsittelyn koulutusohjelma

Raportin/Opinnäytetyön nimi
Big Data ja Apache Hadoop

Sivu- ja liitesivumäärä
30+3

Opinnäytetyön tarkoituksena on luoda tiivistelmä big datasta, Hadoopista ja sen käytöstä. Mitä big data on, mihin sitä käytetään ja mitkä ovat sen hyödyt.

Datan kerääminen sinänsä on suhteellisen helppoa, kerätään data ja säilötään se. Tällä ei kuitenkaan vielä saavuteta minkäänlaista lisäarvoa. Kerätty data täytyy yhdistää muuhun yrityksen dataan. Laitedata (IoT) pitää yhdistää esimerkiksi tuotedataan tai yrityksen keräämä data täytyy yhdistää ulkoisesti hankittuun dataan esimerkiksi säädataan.

Tietolähteinä on käytetty aiheeseen liittyvää kirjallisuutta, lehtiartikkeleita ja videoita. Kaikki lähteet käsittelevät Big dataa.

Opinnäytetyössä käydään läpi big datan työvälineitä Apache Hadoop, MapReduce, HDFS, Pig ja Hive. Työvälineitä käsitellään näkökulmasta mihin ja millaiseen tiedon käsittelyyn kukin näistä sopii.

Internet of things (IoT) on kuvaus sille, että laitteita on kytketty verkkoon. Tämä liittyy oleellisesti big dataan. Laitteista voi verkon kautta kerätä tietoa ja niitä voi ohjata verkon kautta.

Big data ja pilviteknologia kulkevat käsikädessä. Big data tarvitsee useita palvelimia prosessiin, pilvipalvelut pystyvät tämän tarjoamaan.

Asiasanat
Big data, hadoop, pilvipalvelut, hive

Abstract

Haaga-Helia ammattikorkeakoulu Oy

Author(s)
Kirsi Honkasalo

Degree Programme
Business Information Technology

Thesis title
Big Data and Apache Hadoop

Amount of pages
30+3

The purpose of the study was to summarise the essentials of Big data, Apache hadoop, and its use. It explains what Big data is, what it is used for, and what its benefits are.

Collecting data is relatively easy; it is collected and stored. However, this alone is not enough to create any kind of added value. The collected data needs to be combined with a company's other data. The device data (IoT) must be combined with e.g. product data, or the the data collected by the company can be combined with externally acquired data, such as weather data.

The sources used in this work include literature, articles and non-print source material related to the subject. All of the sources analyze Big data.

The study examines tools used with Big data, such as Apache Hadoop, MapReduce, HDFS, Pig and Hive. The tools are analyzed with the emphasis put on what kind of data management each of these is best suited for.

The Internet of Things (IoT) is a term referring to various devices being connected by a network. This is an essential part of Big data. The devices can be used to collect data over the network, which can also be used to control them.

Big data and cloud technology are closely related subjects. Big data processes requires multiple servers, which can be provided by cloud services.

Keywords
Big data, hadoop, cloud services, hive

Sisällys

1 Johdanto

Opinnäytetyössä käsitellään big dataa ja asennetaan Apache Hadoop ja Hive. Opinnäytetyön tavoitteena on tutkia big dataa käsitteenä ja kokeilla pienimuotoisesti datan analysointia Trafin avoimesta datasta. Opinnäytetyössä ei käsitellä tietoturvaa, eikä vertailla datan käsittelyyn soveltuvia välineitä.

Big data on suurten jatkuvasti lisääntyvien tietomassojen keräämistä ja analysointia. Big datalle ei ole yhtä hyväksyttyä määritelmää, vaan se on nimitys suurille datamäärille joihin ei voida käyttää perinteisiä datahallintatapoja. Big datalla on kuitenkin muutamia tunnusomaisia piirteitä, joilla sitä kuvataan:

- Dataa tulee eri muodoissa ja eri lähteistä.
- Usein jonkin laitteen tuottamaa.
- Big datan käyttöä ei välttämättä ole suunniteltu etukäteen ennen keräämistä.
- Sitä ei voida käsitellä käytössä olevilla ohjelmistoilla tai laitteilla.

Big data käsittelee datan määrän kasvua, sisällön monipuolistumista ja tarvetta tunnistaa oleellinen data. Tämä vaatii nopeaa reagointia siitä jalostettuun informaatioon.

Pilvipalvelut tarjoavat alustan suurten datamäärien tallentamiseen ja niiden yhdistelemiseen, joustavasti, kustannustehokkaasti ja nopeasti.

Google, eBay ja LinkedIn kokeilivat ensimmäisenä big dataa. He kehittivät proof of consept:in ja pienen mittakaavan hankkeita oppiakseen voitaisiinko heidän analyyttisiä malleja parantaa uusilla tietolähteillä. Monissa tapauksissa heidän kokeiden tulokset olivat myönteisiä.
Tänä päivänä big data ei ole enää vain kokeellinen työkalu. Monet yritykset ovat alkaneet saavuttaa todellisia tuloksia pyrkiessään keräämään ja käsittelemään enemmän dataa.

Käsitteet

Data	äänteet, kirjaimet, bitit
Hadoop	joukko Open Source -projekteja
HDFS	Hadoop Distributed File System
Hive	mahdollistaa Big Datan käsittelyn pelkistetyllä SQL:llä
IaaS	Pilvipalveluiden infrastruktuuri palveluna
MapReduce	java-pohjainen kehys
Metatieto	on tietoa tiedosta, kuvailevaa ja määrittävää tietoa jostakin tietovarannosta tai sisältöyksiköstä
PaaS	Pilvipalveluiden sovellusalusta palveluna
Pig	skriptikieli joka mahdollistaa Big Datan käsittelyn ilman javaa
Relaatiotietokanta	taulujen tiedot yhdistetään toisiinsa toisen taulun avaimella
SaaS	Pilvipalveluiden sovellukset palveluna
Sudo	sudo käyttäjänä ja sudo komennolla voi komentoja surittaa pääkäyttäjäoikeuksilla Linuxissa.
Tietokanta	tietovarasto, kokoelma tietoja joilla on yhteys toisiinsa
Tervapallo	on tar-ohjelmalla tehty pakattu tiedostoarkisto

2 Big datan määrite

Big data on dataa, joka ylittää tavanomaisten tietokantojen jalostuskapasiteetin. Tietoa on liikaa ja se liikkuu liian nopeasti. Jotta suuria tietomassoja voidaan käsitellä, täytyy valita vaihtoehtoisia tapoja käsitellä sitä. (Dumbill 2012, 10.)

Usein relaatiotietokannoissa puhutaan "schema-on-write" -ajattelusta (tiedetään mihin tauluihin ja sarakkeisiin tallennettava tieto viedään transaktiossa), big datan yhteydessä puhutaan "schema-on-read" -mallista.
Suuri määrä syntynyttä dataa halutaan talteen tallennusjärjestelmään, mutta ei olla vielä kiinnostuneita siitä, millaisiin tietorakenteisiin data menee ja kuinka sitä tullaan käsittelemään. Tärkeintä on kerätä suurella nopeudella syntyvä datamassa talteen skaalautuvaan ja varmistettuun tallennusjärjestelmään, josta dataa voidaan myöhemmin hakea, tarkastella ja analysoida eri menetelmin. (Hotti 2014.)

Data jaetaan karkeasti kahteen eri tyyppiin: strukturoituun ja strukturoimattomaan dataan. Strukturoitu data on esimerkiksi avainsanoilla varustettu videomateriaali, strukturoimatonta on itse video. Näiden kahden välimuoto on semistrukturoitu data. Esimerkkinä voisi olla videomateriaali jossa datan yhteyteen on liitetty metatietoja. (Salo 2013, 25.)

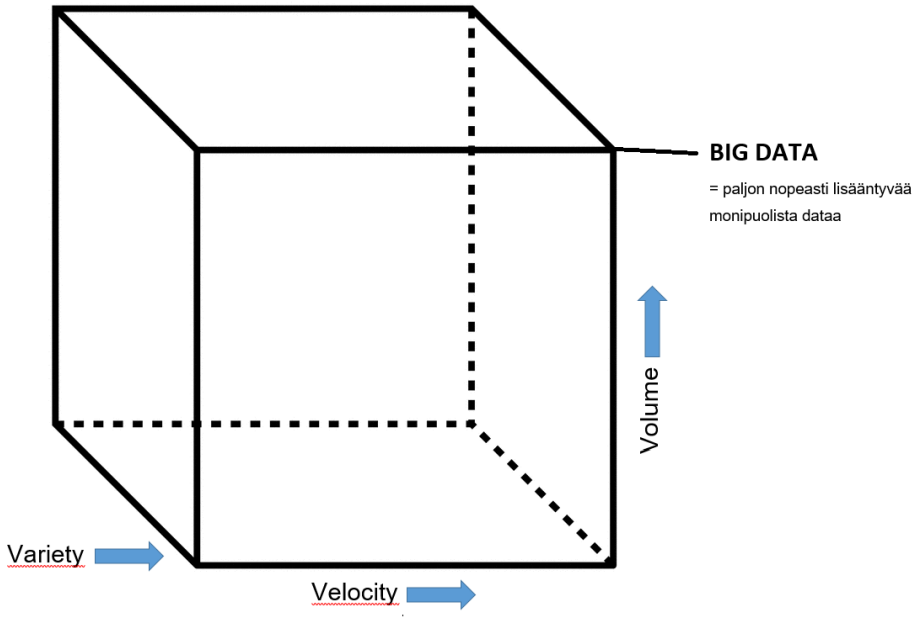
Big datan liitetään usein kolme V-kirjainta (kuva1).

- Volyme (volyymi), kyky käsitellä suuria tietomääriä.
- Variety (vaihtelevuus), kyky käsitellä vaihtelevamuotoista dataa.
- Velocity (vauhti), kyky käsitellä dataa nopeasti.

Tietoa on paljon, sen määrä kasvaa ja se on vaihtelevaa.

Big datan tiedostojärjestelmästä ei poisteta dataa. Sinne luodaan koko ajan uutta. Järjestelmä skaalautuu lisäämällä klusteriin solmuja, datan lukeminen on nopeaa koska se tehdään sekventiaalisesti ja rinnakkaisprosessoinnilla solmujen kesken. (Hotti 2015.)

Ei ole kovin kauan aikaa sitten, kun teratavua pidettiin suurena. Tänä päivänä luodaan 2,5 triljoonaa tavua dataa päivässä. Luomme tietoa niin nopeasti, että 90 prosenttia nykymaailman tiedosta on luotu kahden viimeisen vuoden aikana. On selvää, että perinteisiä tapoja hallita ja käsitellä tietoa on muutettava.



Kuva 1: Volyme, Variety, Velocity

Koska datan määrä kasvaa kiihtyvällä vauhdilla, datasta pitäisi saada poimittua talteen oleellinen informaatio. Markkinoilla on jo suurten tietomassojen käsittelyyn tarkoitettuja sovelluksia ja palveluita. Yksi tunnetuimmista on Apache Hadoop. Hadoop mahdollistaa tiedon analysoimisen ja käsittelyn nopeasti. (Mäkinen 2014.)

2.1 Big datan hyödyt

Tietoa saadaan joka puolelta. Tätä tietomäärää hyödyntämällä ja analysoimalla yritykset voivat kehittää liiketoimintaansa.
Asiakkaiden seuraaminen voi herättää helposti negatiivisia käsityksiä. Verkkopalveluissa voi seurata, missä kohtaa asiakas ei löydä etsimäänsä, tai he eivät pääse eteenpäin. Näin voidaan tunnistaa asiakkaan ongelmat. Yksi big datan lisäarvo on se, että asiakasta voidaan auttaa, kun havaitaan ongelmakohtia vaikkapa teknologiatuotteiden käytössä. Maailmalla on digitaalisissa palveluissa paljon tunnettuja asiakaskäyttäytymisen ja -kokemuksen analysoijia, kuten esimerkiksi Netflix, Amazon, Google. (Helin 2013.)

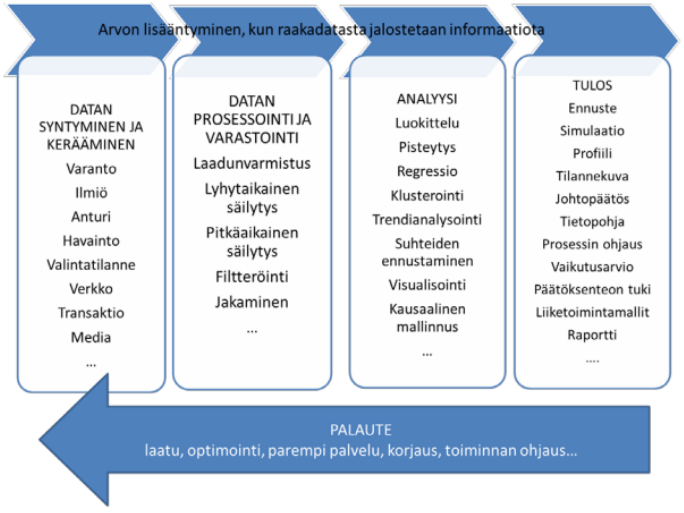
Big dataa voidaan hyödyntää lähes missä tahansa. Palveluyritykset voivat hyödyntää sitä asiakkaidensa tarpeiden ymmärtämisessä, operaattorit voivat hinnoitella liittymiä, päivtäi-

tavarakaupoille on hyödyllistä tietää millaisia asiakkaita kaupassa käy ja mitä he ostavat. (Remes 2013, 9.)

Vaihtoehtoisesti monimutkaisempia ennakoiva ja ohjaileva mallinnus voi auttaa yrityksiä ennakoimaan liiketoimintamahdollisuuksia ja tehdä päätöksiä, jotka vaikuttavat voittoon. Kohdistamalla markkinointikampanjoita ja vähentämällä laitevikoja asiakasvaihtuvuus vähenee.
Big data on tehokas tapa hyödyntää tietoa kuluttajien käyttäytymisestä ja näin saada suurempi hyöty esimerkiksi markkinointikampanjoista. Mitä enemmän yritys hyödyntää big dataa, sitä todennäköisemmin se saavuttaa tai ylittää asetetut tavoitteet. Siltikään monetkaan yritykset eivät hyödynnä big dataa tehokkaasti, koska eivät ymmärrä big dataa tai siitä saatuja etuja. (Rubin, 2013.)

Esimerkkinä big data analyysistä voisi olla se, että sillä voitaisiin lisätä sadon määrää. Se auttaisi maanviljelijöitä tekemään parempia päätöksiä esimerkiksi siitä, milloin istuttaa tai korjata sato.
Climate Corporationin perusti kaksi googlen entistä työntekijää ja maataloudessa toiminut Mosanto vuonna 2013. Se toimii pilvipohjaisena viljelytietojärjestelmänä, jossa otetaan huomioon sään mittaukset 2,5 miljoonasta paikasta päivittäin. Se käsittelee säädatan lisäksi myös 150 miljardia maaperähavaintoa, ja tuottaa 10 triljoonaa sääsimulointi datapistettä. Näiden tietojen avulla yhtiö sanoo voivansa tarjota Yhdysvaltain viljelijöille lämpötilan, sateen ja tuulen ennusteet niinkin pienelle alueelle kun 80 hehtaaria. (Rubens, 2014.)

> ”Kerättävää tai käytettävää dataa analysoimalla haetaan monipuolista ymmärrystä ilmiöistä, josta dataa on kerätty. Analyysien tavoitteina voivat olla muun muassa tuki päätöksille, erilaisten profiilien luonti, erilaiset simulaatiot ja prosessien ohjaukseen vaikuttavat tekijät. Raakadatasta luodaan käsittelyn ja analyysin jälkeen jalostuneempaa informaatiota, jonka muotoja on esitetty datan hyödyntämisen arvoketjua esittävän kuvan 1 TULOS-laatikossa.” (kuva 2) (Rastas & Asp 2014.)



Kuva 2: TULOS-laatikko, (Liikenne- ja viestintävirasto 2014)

3 Esineiden internet

Esineiden internet (internet of things) on kuvaus sille, että laitteita on kytketty verkkoon. Niistä voi verkon kautta kerätä tietoa ja niitä voi verkon kautta ohjata. (Meriläinen-Tenhu 2015.)

Yhtiöt kehittelevät uutta verkostoa keskenään kommunikoiville arkipäivän esineille. Verkostosta käytetään nimitystä Internet of Things, (IoT) tai Teollisuus 4.0. IoT:ssä esineet ovat toisiinsa yhteydessä, jolloin ne voivat vaihtaa keskenään tietoja suorittaakseen käyttäjän asettamat tehtävät. Teollisuus 4.0 on älykkäiden esineiden verkko. Teollisuuden koneet, laitteet ja prosessit antavat käskyjä ja kommunikoivat toistensa kanssa.

Jääkaapit pystyvät pian kommunikoimaan paitsi älypuhelinten, myös tuottajan palvelinfarmin tai energialaitoksen kanssa. Uudesta teknologia- ja kommunikointibuumista vastaavat yhtiöt tulevat kaikilta aloilta: uutta suuntaa eivät ota pelkästään isojen ohjelmistojen tahot kuten Google, Microsoft ja Apple, vaan myös suuret vakuutusyhtiöt, lisälaitteiden tuottajat ja autonvalmistajat edistävät IoT:tä.
Tässä yhtälössä internet toimii kommunikoinnin mekanismina, mikä esineiden kantilta merkitsee tiedon välittämistä. Yhtälön toinen puolisko eli esineet viittaavat alati kasvavaan määrään älykkäitä ja toisiinsa yhteydessä olevia esineitä, jotka muuttavat käsityksemme esineistä perustavanlaatuisesti. Yhdistettävyys laajentaa esineiden toimintakykyä ja tuottaa arvokasta dataa, joka esimerkiksi auttaa yhtiöitä ymmärtämään kuluttajakäyttäytymistä paremmin.

IoT tuo myös muutoksen siihen, miten valmistajat ja palveluyritykset vuorovaikuttavat asiakkaiden kanssa. Tätä nykyä yhtiö myy tuotteen asiakkaalle ja odottaa asiakkaan yhteydenottoa siinä tapauksessa, että jokin menee pieleen. Sen vuoksi luotamme massiivisiin puhelinpalveluihin ja kehittyneisiin asiakaspalveluosastoihin. IoT on selvästi aikeissa muuttaa tämän: tuotteet ovat suorassa yhteydessä palveluun, joka arvioi niiden tilanteen ennen asianomaiseen toimeen ryhtymistä.
IoT tekee älypuhelimistamme kaukosäätimiä, joilla voidaan säädellä lukuisia arkielämämme asioita. IoT:ssa vallankumouksellista on se, että laitteet tavallaan tuntevat meidät ja auttavat säästämään aikaa vaikkapa mobiilimaksuilla tai paikallistamisjärjestelmillä, joiden avulla pääsemme nopeasti sijainnista toiseen.

Älypuhelimemme ovat yhä tiiviimmässä vuorovaikutuksessa ympäristömme kanssa, joka tulee olemaan täynnä meille näkymättömiä sensoreja. Ne antavat mobiililaitteillemme ar-

vokasta tietoa ja käynnistävät sovelluksia puolestamme, mikä on nopeampaa kuin manuaalinen käyttö. (Farnham, 2015.)

IoT:stä tulee varsinainen tietotehdas: sen avulla yhtiöt pystyvät keräämään enemmän dataa kuin koskaan aiemmin. On sanomattakin selvää, että IoT:n tuottaman datamäärän myötä data-analyytikot ja strategit tarvitsevat uudet- tai laajennetut roolit.
Sellaiset yhtiöt, jotka ottavat vastaan kaikista toisiinsa yhdistyneistä esineistä ja laitteista tulvivaa dataa, tarvitsevat asianomaiset työkalut tiedon jäsentelyyn ja analysointiin, jotta ne voisivat selvittää kuluttajien ja työvoiman taipumuksia. Johtajat voivat tarkkailla läheisesti työryhmiensä käyttäytymistä ja tottumuksia, valita niistä parannettavia seikkoja ja muokata yhtiön toimintaperiaatteita ja työympäristöjä siten, että ne vastaavat ammattilaisten tarpeita ja edistävät heidän tuottavuuttaan.
On selvää, että lukemattomien laitteiden lähettämän jatkuvan tietovirran vuoksi tekniikka-alan on pystyttävä käsittelemään valtavaa tietomäärää tai keksittävä uusia ja tehokkaita tapoja ohjelmoida verkostoja ja laitteita sekä myöskin otettava uudet lähteet osaksi toimiaan. (Chuin M, Löffler M, Roberts R. 2010.)

4 Big data ja pilvipalvelut

Big data ja pilviteknologia kulkevat käsikädessä. Big data tarvitsee useita palvelimia prosessiin, pilvipalvelut pystyvät tämän tarjoamaan.
Avoin data ja pilvipalvelut ovat luoneet erilaisia datavarastoja, jotka muuttavat tiedon käyttöä ja keruuta.

Big Data asettaa paljon uusia vaatimuksia palvelinsaleille ja Cloud-infrastruktuurille. Useat Big Data -ratkaisut, kuten Hadoop, olettavat että solmut käyttävät keskenään erillisiä levyjä siten, että levyrikon sattuessa ei menetetä jaettujen levyjen vuoksi usean näennäisesti replikoidun solmun dataa. Tarvittavat kapasiteetit big data -kontekstissa ovat suuria sekä levyn koossa mitattuna, että muistin määrässä. Myös muistia suuremmat flash-levyt ovat hyödyllisiä joissakin sovelluksissa niiden tarjoaman olemattoman hakuajan vuoksi. (Keski-Valkama 2014.)

NIST:n (National Institute of Standars and Technology) määritelmä pilvipalvelimille:

> “Pilvipalvelut on toimintamalli, joka mahdollistaa pääsyn vapaasti konfiguroitaviin ja skaalautuviin tietotekniikkaresursseihin, jotka voidaan ottaa käyttöön tai poistaa käytöstä helposti ja nopeasti.”

Pilvipalvelu on malli, jossa kokonainen tai osittainen palvelu siirretään pois yrityksen lähdeverkosta.

Yleisen määrittelyn lisäksi pilvipalvelulla on myös muutamia ominaispiirteitä.

- Itsepalvelullisuus
- Palveluihin pääsy eri päätelaitteilla
- Resurssien yhteiskäyttö
- Nopea joustavuus
- Käytön tarkka mittaaminen

Itsepalvelullisuus tarkoittaa, että resursseja saa käyttöön ja niiden käytön voi lopettaa ilman että tarvitsee olla yhteydessä palveluntarjoajaan.
Palveluihin pääsy eri päätelaitteilla tarkoittaa, että palveluiden käyttö onnistuu työasemalla, kannettavalla tietokoneella ja mobiililaitteella.
Resurssien yhteiskäyttö tarkoittaa sitä, että palveluntarjoajan resurssien käyttöaste on korkea. Lukuisat asiakkaat käyttävät samaa laitteisto- ja ohjelmistokapasiteettiä toisistaan tietämättä tai riippumatta. (Salo, 2014, s.93-94.)

Nopea joustavuus tarkoittaa, että tarjotut ja käytössä olevat palvelut skaalautuvat joustavasti ja nopeasti ylös- ja alaspäin.
Käytön tarkka mittaaminen tarkoittaa, että asiakas maksaa vain käyttämästään kapasiteetista.

Pilvipalvelut jakautuvat kolmeen eri kategoriaan: Infrastruktuuri palveluna (IaaS), sovellusalustat palveluna (PaaS) ja sovellukset palveluna (SaaS).

IaaS-mallissa palveluntarjoaja tarjoaa resurssit asiakkaan käyttöön palveluna. Jotta voidaan puhua pilvipalvelusta, tulee sen olla joustava, skaalautuva ja itsepalveluna käytettävä, vailla minimiostovelvoitetta. Kapasiteettia voi ottaa joustavasti käyttöön tai poistaa sitä.

PaaS tarjomat tarjoavat alustan ja rajapinnat, joissa sovelluksia voidaan kehittää ja joilla niitä voidaan testata ja ylläpitää. Kun käytetään alustoja, kehitystyöstä tulee nopeaa ja kustannustehokasta. Lopputulos skaalautuu massiivisiin käyttäjäryhmiin saakka ilman lisätyötä. PaaS-tarjoomat tukevat modulaarista ajattelua. Niiden tarjooma osa-alueet ovat valmiiksi palvelullistettuja ja fyysiset resurssit abstraktioiden takana. Tämä tarkoittaa sitä, että käyttöön otettaviin SaaS tarjoomiin voidaan rakentaa helposti omia laajennoksia tai kehittää alusta loppuun omia sovelluksia.

SaaS tarjoomat tarjoavat sovellukset palveluna. Omistamisen, asentamisen, ylläpidon ja päivittämisen sijaan, voidaan ostaa sovellukset käyttöön tarvittaessa. Tämä toimintamalli alentaa ohjelmistoihin ja niihin liittyvien laitteistoihin sidotun pääoman määrää, poistaa ylläpidon ja päivittämisen tarpeen. (Apprenda.)

4.1 Pilvipalveluiden riskit

Tyypillisiä pilvipalveluihin liittyviä riskejä ovat:

Data on sellaisenaan arvoton, mutta jalostettaessa siitä voidaan saada paljon hyötyä. Tämän ajatuksen ympärille rakentuu myös big datan idea. Kaikkiin pilvipalveluihin liittyy jollain tavalla datan tallentaminen, käsittely ja liikuttelu. Datan tallentamiseen liittyy myös tietoturvahuoli, pilvipalvelu on otollinen kohde tietomurroille.
Lainsäädännöllisistä syistä tiedon säilyttämisen tavoista ja fyysisestä sijainnista on usein vaatimuksia. Esimerkiksi henkilötietoja ei välttämättä haluta säilyttää Suomen tai EU:n ulkopuolella.
Pilvipalveluun tallennetun dataan liittyy myös saavutettavuus ja pysyvyys huolet. Jos pilveen ei saada yhteyttä, ei voida myöskään käyttää siellä olevia tietoja.

Tietojärjestelmien heikoin lenkki on käyttäjä, myös pilvipalveluissa. Käyttäjä voi olla huolimaton, pahantahtoinen tai tietämätön, tämä aiheuttaa ongelmia joihin on hankalaa varautua etukäteen. Käyttäjän huolimattomuus ja välinpitämättömyys ovat yleisiä heikkoja kohtia. Käyttäjät käyttävät samaa salasanaa moneen eri palveluun tai käyttävät helposti arvattavaa salasanaa.

Jos kapasiteetti ei olekaan tarvittaessa käytössä eli saatavuudessa on ongelmia. Ongelmia tulee myös, jos verkkoyhteydet pettävät tai niiden kapasiteetti alenee merkittävästi. Koska palvelu on kaukana käyttäjästä, se on riippuvainen tietoliikenneyhteyksistä. Jos pilvipalvelu hidastelee, voi syy olla käyttäjän päätelaitteessa ja siihen integroidusta (pilvipalvelu) järjestelmästä. Jos pilvipalvelussa ei ole vikaa, on yleensä ongelman paikantaminen hidasta, joskus jopa lähes mahdotonta.

Yleisesti palveluntarjoajan fyysisiin tiloihin ei pääse, eikä silloin ole myöskään kykyä varmistaa tilojen, laitteiston, tai henkilöstöön liittyviä turvallisuus- ja muita seikkoja.

Koska pilvipalveluiden keskeisimpiä valtteja ovat skaalautuvuus ja itsepalvelullisuuden mahdollistaman käyttöönotto ja nopeat muutokset, on palveluntarjoajilla vaihteleva mittaristo palveluidensa suorituskyvystä.

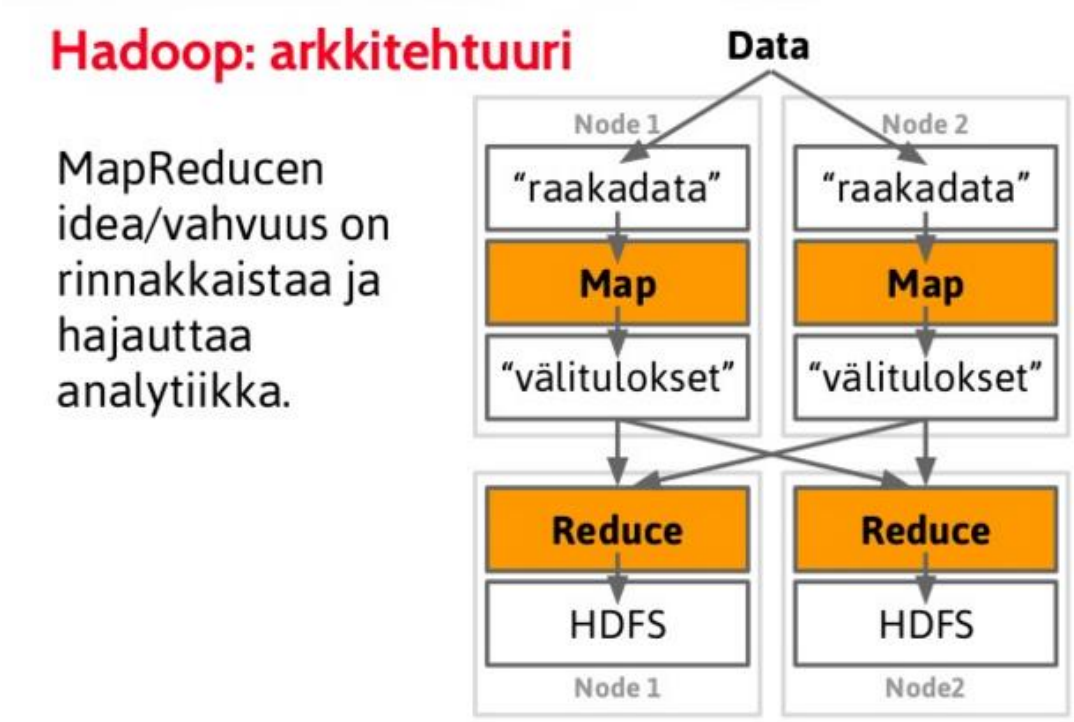
Jos palvelun tuottamisen tarvittavat resurssit ja niihin liittyvät säännölliset huolto- ja muut toimenpiteet ovat heikosti asiakkaan tiedossa, asiakkaan on vaikea arvioida riskejä ja toteutumistodennäköisyyksiä. (Salo, 2014 s.104-111.)

5 Apache Hadoop

Hadoop, viralliselta nimeltään Apache Hadoop, on Apache Software Foundation projekti ja avoimen lähdekoodin skaalautuva ohjelmistoalusta, hajautettuun laskentaan.
Hadoop analysoi luotettavasti ja nopeasti sekä jäsenneltyä tietoa että jäsentelemätöntä tietoa.
Apache Hadoopin ohjelmistokirjasto on pohjimmiltaan kehys, joka mahdollistaa hajautettujen suurten tietomäärien käsittelyn käyttäen yksinkertaista ohjelmointimallia. Hadoop skaalautuu jopa yksittäisistä palvelimista tuhansiin koneisiin, joissa jokaisessa on paikallinen laskenta ja talletus.

Apache Hadoop ei ole relaatiokanta, se on tiedostojärjestelmä, joka tukee hajauttamista ja ottaa vastaan kaikenlaista dataa.
Hadoop on optimoitu suurten datamassojen tallentamiseen ja niiden nopeaan rinnakkaiseen käsittelyyn peräkkäisluennalla - jopa satojen tai tuhansien palvelimien osallistuessa datan prosessointiin.

Käytännössä Apache Hadoop on joukko Open Source -projekteja (kuva 3) joista tärkeimmät ovat Hadoopin ydin; HDFS (Hadoop Distributed File System) sekä MapReduce. (Hovi 2014.)

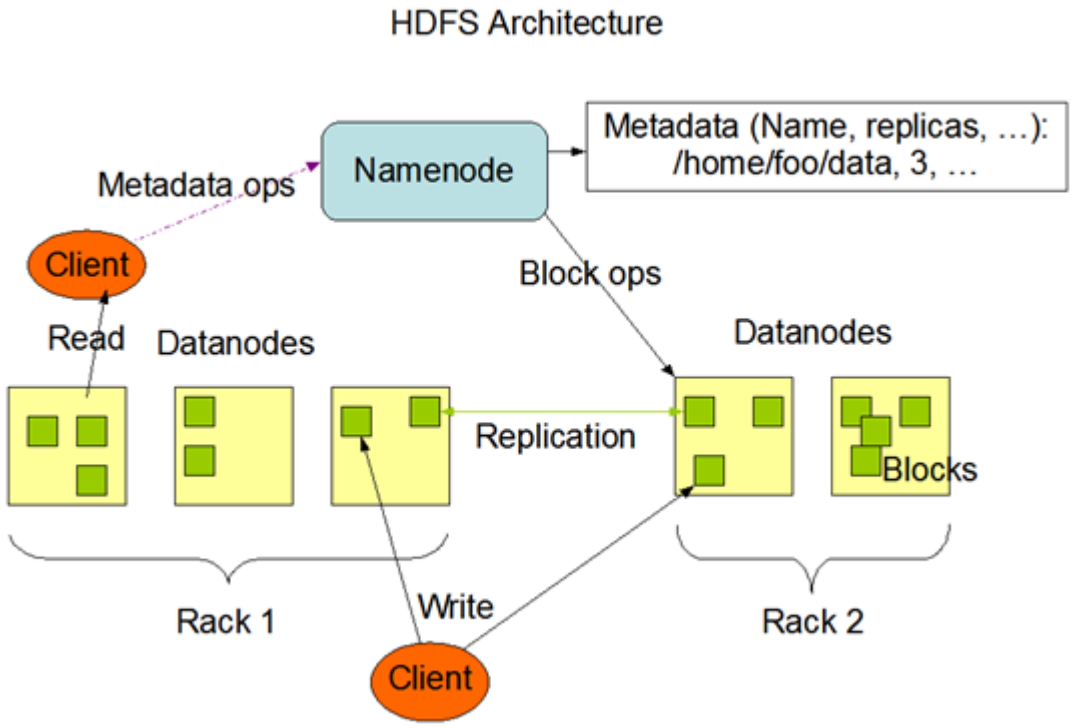


Kuva 3: Hadoop arkkitehtuuri (Ivorio 2013)

5.1 HDFS

HDFS tarjoaa hajautetun redundantin tiedostojärjestelmän, joka voidaan rakentaa halvoista levypalvelimista. Käytännössä HDFS virtualisoi jopa tuhansilla palvelimilla olevan levytilan yhdeksi hakemistorakenteeksi, johon voidaan kirjoittaa mitä tahansa dataa. HDFS tyypillisesti viipaloi datan 64-256 megatavun paloihin ja kopioi dataviipaleet oletusarvoisesti vähintään kolmelle eri palvelimelle. Tällä tavalla HDFS on vikasietoinen.

HDFS:ssä on isäntä-/orja-arkkitehtuuri (kuva 4). HDFS-klusteri koostuu yhdestä namenodesta eli isäntäpalvelimesta, joka hallinnoi tiedostojärjestelmän nimiavaruutta ja säätelee asiakasohjelmien pääsyä tiedostoihin. Lisäksi klusteriin kuuluu useita datanodeja, jotka hallinnoivat käyttämiensä solmujen tallennustilaa. HDFS paljastaa tiedostojärjestelmän nimiavaruuden ja mahdollistaa käyttäjätietojen tallentamisen tiedostoihin. Tiedosto on jakautunut yhteen tai useampaan lohkoon, jotka on varastoitu datanodeihin. Namenode käynnistää tiedostojärjestelmän nimiavaruuteen liittyviä toimintoja, kuten tiedostojen ja kansioiden avaamisen, sulkemisen ja uudelleennimeämisen. Namenode myös päättää lohkojen liittämisestä datanodeihin. Datanodejen vastuulla on käsitellä tiedostojärjestelmän asiakasohjelmien lähettämiä luku- ja kirjoituspyyntöjä. Datanodet myös luovat, poistavat ja kopioivat lohkoja namenoden ohjeiden mukaisesti.



Kuva 4: HDFS Arkkitehtuuri (Apache Hadoop 2013)

Namenode ja datanode ovat ohjelmiston osia, jotka on suunniteltu toimimaan kotitietokoneilla. Näiden koneiden käyttöjärjestelmänä on tavallisesti GNU/Linux. HDFS on rakennettu Java-kielellä: mikä tahansa Java-kieltä tukeva kone voi siis käyttää namenode- tai datanodeohjelmaa. Varsin liikuteltavan Java-kielen käyttö tarkoittaa, että HDFS:ää voidaan käyttää useissa koneissa. Tavallisessa järjestelyssä yksi kone on varattu namenodeohjelmiston käyttöön. Kaikki klusterin muut koneet käyttävät yhtä datanodeohjelmistoa. Tällainen arkkitehtuuri ei estä useiden datanodejen käyttöä samalla koneella, mutta oikeassa järjestelyssä niin tehdään harvoin.

Yhden namenoden käyttäminen klusterissa yksinkertaistaa järjestelmän arkkitehtuuria suuresti. Namenode välittää ja säilyttää HDFS:n kaiken metadatan. Järjestelmä on suunniteltu siten, etteivät käyttäjän tiedot kulkeudu koskaan namenoden läpi.
HDFS tukee perinteistä hierarkkista tiedostojärjestelyä. Käyttäjä tai sovellus voi luoda kansioita ja tallentaa niihin tiedostoja. Tiedostojärjestelmän nimiavaruuden hierarkia on samanlainen kuin enimmissä olemassa olevissa tiedostojärjestelmissä: käyttäjä voi luoda ja poistaa tiedostoja, siirtää tiedoston kansiosta toiseen tai nimetä tiedoston uudelleen. HDFS:ssä ei ole käyttäjäkiintiöitä. HDFS ei tue kovia eikä pehmeitä linkkejä. HDFS-arkkitehtuuri ei kuitenkaan estä näiden toimintojen lisäämistä. (Apache Hadoop 2013.)

Namenode hallinnoi tiedostojärjestelmän nimiavaruutta. Namenode tallentaa kaikki muutokset tiedostojärjestelmän nimiavaruuteen ja sen ominaisuuksiin. Sovellus pystyy määrittämään, montako tiedostokopiota HDFS:n tulee säilyttää. Namenode varastoi tiedot siitä, montako kopiota tiedostosta on olemassa.

5.2 MapReduce

MapReducella käsitellään HDFS-tiedostojärjestelmässä olevaa dataa halutulla tavalla, kun ollaan kiinnostuneita datan sisällöstä. MapReduce-prosessi on eräajopohjainen tapa lukea ja analysoida big dataa - ja kaikki muutkin Hadoopin teknologiat käyttävät alla MapReduce-menetelmää.

Hadoop MapReduce on ohjelmistokehys, jolla on helppo kirjoittaa suuria tietomääriä (useiden teratavujen kokoisia tiedostoja) prosessoivia sovelluksia, jotka toimivat rinnakkain suurten kotikoneklusterien kanssa (tuhansia solmuja) luotettavalla ja vikasietoisella tavalla.

MapReduce-tehtävässä syötetyt tiedot pilkotaan paloiksi, jonka tehtävät prosessoivat täysin rinnakkaisesti. Kehys lajittelee tuotetut tiedot, jotka syötetään sitten Reduce-tehtäviin.

Tehtävää varten syötetyt tiedot ja sen aikana tuotetut tiedot varastoidaan tavallisesti tiedostojärjestelmään. Kehys ajoittaa tehtävät, tarkkailee niitä ja käynnistää uudelleen epäonnistuneet tehtävät.

Laskenta- ja varastosolmut ovat tavallisesti samat, eli MapReduce-kehys ja HDFS käyttävät samoja solmuja. Tällaisen järjestelyn ansiosta kehys pystyy jakelemaan tehtäviä tehokkaasti solmuihin, joissa tietoja on jo valmiina. Näin klusteriin saadaan varsin suuri kaistanleveys.
MapReduce-kehys koostuu yhdestä isäntänä toimivasta JobTrackerista ja yhdestä orjana toimivasta TaskTrackerista per klusterin solmu. Isännän vastuulla on jakaa tehtävien osat orjille, tarkkailla niitä ja käynnistää uudelleen epäonnistuneet tehtävät. Orjat suorittavat tehtävät isännän määräysten mukaisesti.
Sovellukset määrittelevät syöttö- ja ulostulosijainnit sekä suorittavat Map- ja Reduce-palveluja asianomaisten käyttöliittymien ja/tai abstraktien luokkien kautta. Tehtävien määrittely koostuu näistä ja muista tehtäväparametreista. Hadoopin JobClient lähettää tehtävän (jar, exe. ym.) ja määritelmät JobTrackerille, jonka vastuulla on lähettää ohjelmisto/määrittely orjille, jakaa tehtäviä ja tarkkailla niitä sekä lähettää JobClientille tilanne- ja vianmääritystietoja. (Apache Hadoop 2013.)

5.3 Pig

Pig on skriptikieli, joka mahdollistaa big datan käsittelyn ilman Javaa. Se pystyy käsittelemään sekä strukturoitua että ei-strukturoitua dataa. Pig-skriptikielellä voidaan tuottaa suurista massoista big dataa järjestäytynyttä ja analysoitua tietoa, jota voidaan myöhemmin hyödyntää eri tavoin. (Hortonworks.)

Pig on korkean tason skriptikieli, jota käytetään Apache Hadoopissa. Pigin erikoisalaa on kuvailla data-analyysin ongelmia tietovirtoina. Apache Hadoopin vaatima datamanipulaatio voidaan tehdä kokonaan Pigillä. Hyödynnettäessä käyttäjän määrittelemiä toimintoja (User Defined Functions, UDF) Pig pystyy myös kirjoittamaan koodia monilla muilla kielillä kuten JRubylla, Jythonilla ja Javalla. Samaten Pigin skriptejä voidaan käyttää muilla kielillä. Sen ansiosta Pigiä voidaan käyttää hankalien yritysongelmien ratkaisemiseen tarkoitettujen suurten ja monimutkaisten sovellusten rakennuksessa.

Hyvä esimerkki Pigin soveltamisesta on ETL-siirtomalli, joka kuvaa, miten prosessi kerää lähteestä tietoa, muokkaa sitä asetettujen sääntöjen mukaisesti ja lataa sen datasäiliöön. Pig voi kerätä dataa tiedostoista, suoratoistoista ja muista lähteistä käyttäjän määrittelemien toimintojen UDF (User Defined Functions) avulla. Hankittuaan datan se voi suorittaa

valikoinnin, toiston ja muita datan muokkaustoimenpiteitä. UDF-toiminto mahdollistaa datan välittämisen monimutkaisempien algoritmien läpi muokkausta tehtäessä. Pig pystyy tallentamaan tulokset HDFS:ään. Pig-skriptit käännetään MapReduce-tehtäviksi tai Tez DAG:ksi, jotka suoritetaan Apache Hadoop klusterissa. Käännöksen osana Pig-tulkki optimoi skriptejä, jotta Apache Hadoopin suorituskyky nopeutuisi. (Apache Hadoop.)

5.4 Hive

Hive mahdollistaa big datan käsittelyn SQL:n menetelmin. Hivellä voidaan luoda big datasta käsittelyä varten skeemoja eli tietokantatauluja sarakkeineen. Hiven avulla voidaan kirjoittaa HDFS:ään, eli Hadoopin tiedostojärjestelmään ja tallennusformaatti natiivisti on peräkkäistiedosto. Tällä tavalla Hive on tietovarastointiteknologia ja infrastruktuuri sille. Hive-kyselyt generoivat MapReduce -prosesseja joita ajetaan rinnakkaisesti big dataa vasten. (Hotti 2015.)

Hive määrittelee yksinkertaisen SQL:ää muistuttavan kyselykielen nimeltään QL, jolla SQL:ään perehtyneet käyttäjät voivat tehdä datasta kyselyjä. MapReduce-kehykseen perehtyneet ohjelmoijat voivat tehdä hienostuneempia analyysejä tämän kielen avulla hyödyntämällä omia ohjelmiaan, joita kieli ei välttämättä tue oletusarvoisesti. QL:ää voidaan laajentaa myös itsetehdyillä skaalausfunktioilla UDF, aggregaateilla UDAF (User-Defined Aggregation Functions) ja taulukkotoiminnoilla UDTF (User Defined Table Function).

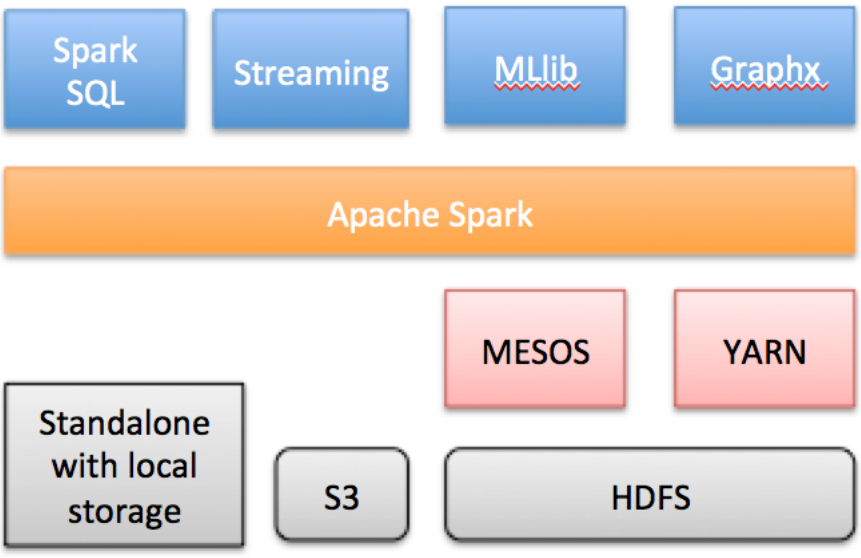
Hive ei valtuuta “Hive-formaatissa” olevan datan lukua tai kirjoittamista, koska sellaista ei ole olemassa. Hive toimii Thriftillä, rajoitetuilla kontrolleilla ja omilla dataformaateilla.
Hivea ei ole suunniteltu OLTP-tehtäviin, eikä sillä voi tehdä reaaliaikaisia kyselyjä tai rivin päivityksiä. Se soveltuu parhaiten suurten liitetietojen (kuten verkkolokien) käsittelyyn.
Hivessä arvokkainta ovat skaalautuvuus (Hadoop-klusteriin voidaan lisätä koneita dynaamisesti), laajennettavuus (MapReduce-kehyksellä ja UDF/UDAF/UDTF-toiminnoilla), vikasietokyky ja sisääntuloformaattien riippumattomuus toisistaan.
Hiven komponentteihin kuuluvat HCatalog ja WebHCat.

5.5 Spark

Apache Spark on avoimen lähdekoodin kehys, jolla prosessoidaan suuria datamääriä. Se pohjautuu nopeuteen, helppokäyttöisyyteen ja hienostuneeseen analyysiin. Se kehitettiin UC Berkeleyn AMPLabissa vuonna 2009, ja siitä tuli Apache-projekti, kun sen lähdekoodi tehtiin avoimeksi vuonna 2010.
Sparkissa on useita etuja verrattuna muihin suurten datamäärien käsittelijöihin ja MapReduce-välineisiin kuten Hadoopiin ja Stormiin.

Spark tarjoaa kattavan ja yhtenäisen kehyksen suurten datamäärien prosessoinnin vaatimuksille. Se sisältää monipuolisia tiedostoja (tekstidataa, kaaviodataa ym.) ja datalähteen (erissä oleva data vs. reaaliaikaisen suoratoiston data).
Sparkin avulla Hadoop-klusterien sovellukset voivat toimia jopa 100 kertaa nopeammin muistia käytettäessä ja 10 kertaa nopeammin levyllä suoritettaessa.
Sparkin avulla voidaan kirjoittaa sovelluksia nopeasti Javalla, Scalalla ja Pythonilla. Sen mukana tulee yli 80 korkeatasoista operaattoria, ja sillä pystyy tekemään datasta kyselyjä interaktiivisesti kuoren sisällä.

Map- ja Reduce-tehtävien lisäksi Spark tukee SQL-kyselyjä, datan suoratoistoa, koneoppimista ja kaaviodatan prosessointia. Kehittäjät voivat käyttää näitä toimintoja erikseen tai yhdessä, jolloin data kulkeutuu niiden kaikkien läpi.
Spark vie MapReducen seuraavalle tasolle. Sen avulla datan prosessointi on halvempaa. Datan tallentaminen muistiin ja lähes reaaliaikainen prosessointi tekevät suorituskyvystä useita kertoja nopeamman kuin muilla suuren datamäärän käsittelyvälineillä.
Spark tukee myös suurten datakyselyjen laiskaa suorittamista, mikä helpottaa dataprosessoinnin vaiheiden optimoimista. Sparkin korkeatasoinen API parantaa kehittäjien tuottavuutta, ja yhtenäinen arkkitehtuurimalli (kuva 5) tukee suurten datamäärien prosessointiratkaisuja.



Kuva 4: Spark arkkitehtuuri (Vellore, T 2015)

Spark säilyttää välivaiheen tulokset muistissa levylle kirjoittamisen sijaan, mikä on hyödyllistä etenkin silloin, kun samoja tietoja on työstettävä useasti. Spark on suunniteltu ohjelmaksi, joka työskentelee sekä muistilla että levyllä. Spark-operaattorit suorittavat ulkoisia

tehtäviä, kun data ei mahdu muistiin. Sparkia voidaan käyttää prosessoimaan tiedostoja, jotka ovat suurempia kuin klusterin kokonaismuisti.
Spark pyrkii tallentamaan mahdollisimman paljon dataa muistiin ja sen jälkeen levylle. Se pystyy tallentamaan osan tiedoista muistiin ja loput levylle.

Vaadittava muistin määrä tulee arvioida datan perusteella. Spark on tehokkaampi muistiintallennustoimintonsa ansiosta.

Muita Sparkin toimintoja ovat:

- Tukee muitakin kuin Map- ja Reduce-toimintoja.
- Optimoi operaattorikaaviot. Suurten datakyselyjen laiska suorittaminen, mikä auttaa optimoimaan datan prosessointia.
- Tiiviit ja yhtenäiset API:t Scalassa, Javassa ja Pythonissa.
- Interaktiiviset kuoret Scalalle ja Pythonille. Tätä ei ole vielä saatavana Javassa.

Spark on kirjoitettu Scala-ohjelmointikielellä, ja se toimii Java Virtual Machine -ympäristössä (JVM). Tällä hetkellä se tukee seuraavia kieliä Sparkia käyttävien sovellusten luomisessa: Scala, Java, Python, Clojure ja R

6 Hadoop, Hive ja Avoin data

Hadooppia käytettäessä Map- ja Reduce mallit voidaan ohjelmoida Javalla omina funktioinaan.
Skripti-kieliä varten on oma rajapinta, joka sallii minkä tahansa kielen käyttämisen. Hadoop käyttää myös omaa tiedostojärjestelmää (HDFS). HDFS hoitaa datan hajauttamisen niin monelle fyysiselle levylle kuin on tarpeen. Tästä syystä analysoitava tiedosto voi olla paljon suurempi kuin yksittäisen koneen tallennuskapasiteetti.
Tieto ladataan ensin HDFS-tiedostojärjestelmään, jonka jälkeen se on Hadoopin käytettävissä. Hadoopilla analysoitava tieto on usein rakenteetonta dataa, joka on jo tiedostoihin tallennettu. MapReduce -ohjelmilla pystytään poimimaan tiedostoista merkitykselliset asiat, ja sen jälkeen niitä voidaan analysoida.
Hadoopin vahvuudet ei välttämättä tule parhaiten esille tiedon analysoimisessa, vaan rakenteettoman tiedon muuttamisessa rakenteelliseksi. Tämän jälkeen data voidaan edelleen ladata tietokantaan analysoitavaksi. (Laukkanen, 2014.)

Hadoopin voi ottaa myös valmiiksi paketoituna kokonaisuutena, palveluja tarjoaa tällä hetkellä mm. Amazon (AWS EMR), Cloudera (CHD), Hortonworks (HDP), MapR (M7), Microsoft (HDInsight), Pivotal (Pivotal HD) ja IBM (IHC)

Projektin tarkoituksena oli asentaa Apache Hadoop ja Hive. Asentamisen jälkeen tein hakuja Trafin tietokannasta.

6.1 Suunnittelu

Jotta pystyn hyödyntämään Hadoopia ja Hiveä, täytyy olla dataa ja kysymyksiä. Päädyin Trafin avoimeen dataan, koska sain ulkopuoliselta taholta kysymyksiä joita he voivat käyttää hyödyksi markkinoinnissa, työssä ja koulutusten suunnittelussa. Tehtäväkseni siis jäi Hadoopin ja Hiven asentaminen ja halutun tiedon hakeminen datasta.

Trafin ajoneuvojen avoimen datan aineistossa 4.4 on 5021314 kpl rivejä, ja sen julkaisuajankohta on 13.1.2016
Datassa oli joitain rajoittavia tekijöitä; ajoneuvon malliin ei pysty tekemään suoria hakuja, koska se sisältää myös muita ajoneuvon teknisiä tietoja. Ajoneuvon mallin saa eroteltua ainoastaan osittaisesta valmistenumerosta, joka tarkoittaa, että minun tulisi tuntea mallinumeroiden rakenne.

Projektissa käytin HP Pavilion kannettavaa, käyttöjärjestelmänä on xubuntu.
Hadoop versio on 2.6.4. Hiven versio on 1.0.1 ja Trafin ajoneuvojen avoin data on 4.4-aineisto.

6.2 Hadoopin asentaminen

Työn ensimmäinen vaihe oli asentaa Hadoop. Netistä on saatavilla ohjeita asennukseen, mutta useimmat niistä ovat melko puutteellisia. Apache Hadoopin sivuilla on melko hyvät ohjeet asennukseen.
Apache Hadoopin sivuilta löytyy Hadoopin päivitetyt versiot ja kertomukset mitä päivityksissä on lisätty / poistettu.

Java on oliopohjainen ohjelmistokieli. Se on perusta kaikenlaisille virtuaalisille aplikaatiolle. Aloitin työn javan asentamisella ja version tarkastamisella (kuva 6). Tähän löytyy myös suositellut versiot asennusohjeesta.

```
xubuntu@xubuntu:~$ java -version
java version "1.7.0_95"
OpenJDK Runtime Environment (IcedTea 2.6.4) (7u95-2.6.4-0ubuntu0.14.04.1)
OpenJDK 64-Bit Server VM (build 24.95-b01, mixed mode)
xubuntu@xubuntu:~$
```

Kuva 5: Java version tarkastaminen

Luon uuden ryhmän ja käyttäjän. Annan käyttäjälle sudo oikeudet. Tämä ei ole pakollista, mutta se on suositeltavaa, koska se auttaa erottamaan Hadoop asennuksen muista ohjelmistosovelluksia ja käyttäjätilejä käyttävistä käyttäjistä.

Ryhmä on siis tässä tapauksessa Hadoop ja käyttäjä hduser.

6.2.1 SSH

SSH on etäkäyttöohjelmisto jolla voidaan luoda salattuja yhteyksiä järjestelmästä toiseen.
Asensin SSH:n ja muokkasin asetuksia (kuva 7).

```
xubuntu@xubuntu:~$ which ssh
/usr/bin/ssh
xubuntu@xubuntu:~$ which sshd
/usr/sbin/sshd
xubuntu@xubuntu:~$
```

Kuva 6: SSH polkujen tarkastaminen

Hadoop käyttää SSH:ta (käyttääkseen sen noodeja). Tämä vaatii normaalisti käyttäjän salasanaa. Poistin tämän vaatimuksen ja määritin sen sallimaan SSH:n julkisen avaimen.

Komennoilla saadaan lisättyä juuri luotu avain valtuutettujen avainten luetteloon, jotta Hadoop voi käyttää SSH:a kysymättä salasanaa.

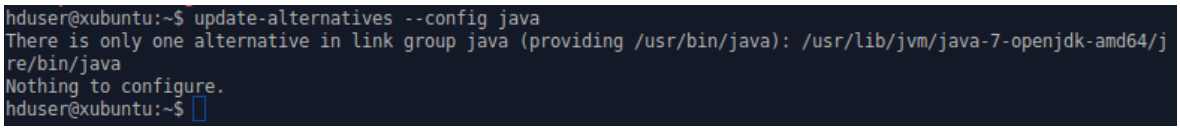
6.2.2 Hadoop asetukset

Hadoopin asentamisen tärkein osa on tiedostojen oikeanlainen konfigurointi. Muokattavia tiedostoja ei ole kuin kuusi, mutta ne ovat sitäkin tärkeämpiä.
Aloitin asentamisen lataamalla Hadoopin tervapallon ja purin sen. Jatkon helpottamiseksi loin kansion nimeltään ’hadoop’ / usr / local /- hakemistoon. Siirsin Hadoop latauksen sinne.

Jotta asennus saadaan toimivaksi, täytyi muokata muutamia tiedostoja:

- ~/.bashrc
- /usr/local/hadoop/etc/hadoop/hadoop-env.sh
- /usr/local/hadoop/etc/hadoop/core-site.xml
- /usr/local/hadoop/etc/hadoop/yarn-site.xml
- /usr/local/hadoop/etc/hadoop/mapred-site.xml.template
- /usr/local/hadoop/etc/hadoop/hdfs-site.xml

Ennen .bashrc muokkaamista tarvitsin tiedon mihin java on asennettu (kuva 8).



Kuva 7: Javan polun tarkastaminen

Bashrc-tiedosto toimii oletuskomentotulkkina. Lisäsin ~/.bashrc tiedoston loppuun tekstin:

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib"
```

Tekstissä täytyy varmistua, että polut ovat oikein. Tallentamisen jälkeen otin vielä muuttujat käyttöön. Bashrc tiedosto varmistaa, että muuttujat ovat aina käytettävissä kun VPS (Virtual Private Server) käynnistyy.

Hadoop-env.sh tiedostoon asetetaan JAVA_HOME

export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64

Hadoop-env.sh tiedosto varmistaa, että arvo JAVA_HOME on saatavilla, kun Hadoop käynnistetään.

core-site.xml Tiedosto sisältää kokoonpano ominaisuudet, joita Hadoop käyttää käynnistettäessä.

Tätä tiedostoa voidaan käyttää ohittamaan oletusasetukset, joista Hadoop käynnistyy.

```
<property>
  <name>hadoop.tmp.dir</name>
  <value>/app/hadoop/tmp</value>
 </property>
 <property>
  <name>fs.default.name</name>
  <value>hdfs://localhost:54310</value>
 </property>
```

Oletuksena / usr / local / hadoop / etc / hadoop / -kansio sisältää / usr / local / hadoop / etc / hadoop / mapred-site.xml.template tiedoston, jonka nimeksi on vaihdettava mapred-site.xml. Tätä tiedostoa käytetään määrittelemään mitä kehyksiä käytetään Map Reduce:sa

```
<property>
  <name>mapred.job.tracker</name>
  <value>localhost:54311</value>
 </property>
```

yarn-site.xml tiedosto sisältää kokonpano ominaisuudet, joita MapReduce käyttää käynnistettäessä. Tällä tiedostolla voidaan ohittaa oletusasetukset, joilla MapReduce käynnistyy.

```
<property>
   <name>yarn.nodemanager.aux-services</name>
   <value>mapreduce_shuffle</value>
</property>
<property>
   <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
   <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
```

/ usr / local / hadoop / etc / hadoop / hdfs-site.xml Tiedosto täytyy määrittää kullekin host klusterille, joka on käytössä.

Sitä käytetään määrittämään hakemistot, joita käytetään namenode, datanode, ja host:ssa.

Ennen tämän tiedoston muokkausta, tein kaksi hakemistoa, jotka sisältävät namenode ja datanode:n tälle Hadoop asennukselle.

HDFS-tiedostoon lisäsin sisällöksi:

```
<property>
  <name>dfs.replication</name>
  <value>1</value>
 </property>
 <property>
   <name>dfs.namenode.name.dir</name>
   <value>file:/usr/local/hadoop_store/hdfs/namenode</value>
 </property>
 <property>
   <name>dfs.datanode.data.dir</name>
   <value>file:/usr/local/hadoop_store/hdfs/datanode</value>
 </property>
```

Tiedostojärjestelmä piti alustaa, jotta sitä voi alkaa käyttää. Alustuskomennolle on myönnettävä kirjoitusoikeudet koska se luo nykyisen hakemiston / usr / local / hadoop_store / hdfs / namenode / -kansion.

```
16/03/22 18:38:21 INFO namenode.FSNamesystem: Append Enabled: true
16/03/22 18:38:21 INFO util.GSet: Computing capacity for map INodeMap
16/03/22 18:38:21 INFO util.GSet: VM type       = 64-bit
16/03/22 18:38:21 INFO util.GSet: 1.0% max memory 889 MB = 8.9 MB
16/03/22 18:38:21 INFO util.GSet: capacity      = 2^20 = 1048576 entries
16/03/22 18:38:21 INFO namenode.NameNode: Caching file names occuring more than 10 times
16/03/22 18:38:21 INFO util.GSet: Computing capacity for map cachedBlocks
16/03/22 18:38:21 INFO util.GSet: VM type       = 64-bit
16/03/22 18:38:21 INFO util.GSet: 0.25% max memory 889 MB = 2.2 MB
16/03/22 18:38:21 INFO util.GSet: capacity      = 2^18 = 262144 entries
16/03/22 18:38:21 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.9990000128746033
16/03/22 18:38:21 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
16/03/22 18:38:21 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension     = 30000
16/03/22 18:38:21 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
16/03/22 18:38:21 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry exp
iry time is 600000 millis
16/03/22 18:38:21 INFO util.GSet: Computing capacity for map NameNodeRetryCache
16/03/22 18:38:21 INFO util.GSet: VM type       = 64-bit
16/03/22 18:38:21 INFO util.GSet: 0.029999999329447746% max memory 889 MB = 273.1 KB
16/03/22 18:38:21 INFO util.GSet: capacity      = 2^15 = 32768 entries
16/03/22 18:38:21 INFO namenode.NNConf: ACLs enabled? false
16/03/22 18:38:21 INFO namenode.NNConf: XAttrs enabled? true
16/03/22 18:38:21 INFO namenode.NNConf: Maximum size of an xattr: 16384
16/03/22 18:38:21 INFO namenode.FSImage: Allocated new BlockPoolId: BP-334626206-127.0.1.1-1458664701230
16/03/22 18:38:21 INFO common.Storage: Storage directory /usr/local/hadoop_store/hdfs/namenode has been success
fully formatted.
16/03/22 18:38:21 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
16/03/22 18:38:21 INFO util.ExitUtil: Exiting with status 0
16/03/22 18:38:21 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at xubuntu/127.0.1.1
************************************************************/
```

Kuva 8: Hadoop namenode format

Hadoop namenode -formaatissa komennon saa suorittaa vain kerran (kuva 9), ennen kuin alkaa käyttää Hadoop:ia. Jos komennon suorittaa uudelleen Hadoopin käyttämisen jälkeen, se tuhoaa kaikki tiedot Hadoopin tiedostojärjestelmästä.
Tarkastukset (kuva 10) on hyvä suorittaa myös aika ajoin asentamisen aikana. Mahdollisen asennusvirheen sattuessa, virheen kohdistaminen on helpompaa.

```
hduser@xubuntu:~$ cd /usr/local/hadoop/sbin
hduser@xubuntu:/usr/local/hadoop/sbin$ ls
distribute-exclude.sh  kms.sh                    start-balancer.sh    stop-all.cmd        stop-yarn.cmd
hadoop-daemon.sh       mr-jobhistory-daemon.sh   start-dfs.cmd        stop-all.sh         stop-yarn.sh
hadoop-daemons.sh      refresh-namenodes.sh      start-dfs.sh         stop-balancer.sh    yarn-daemon.sh
hdfs-config.cmd        slaves.sh                 start-secure-dns.sh  stop-dfs.cmd        yarn-daemons.sh
hdfs-config.sh         start-all.cmd             start-yarn.cmd       stop-dfs.sh
httpfs.sh              start-all.sh              start-yarn.sh        stop-secure-dns.sh
hduser@xubuntu:/usr/local/hadoop/sbin$
```

Kuva 9: Tiedostojen tarkastaminen

Käynnistin single node klusterin ja tarkastin että kaikki toimii (kuva 11).

```
hduser@xubuntu:~$ jps
13257 ResourceManager
13111 SecondaryNameNode
13675 Jps
12904 DataNode
12786 NameNode
13377 NodeManager
hduser@xubuntu:~$
```

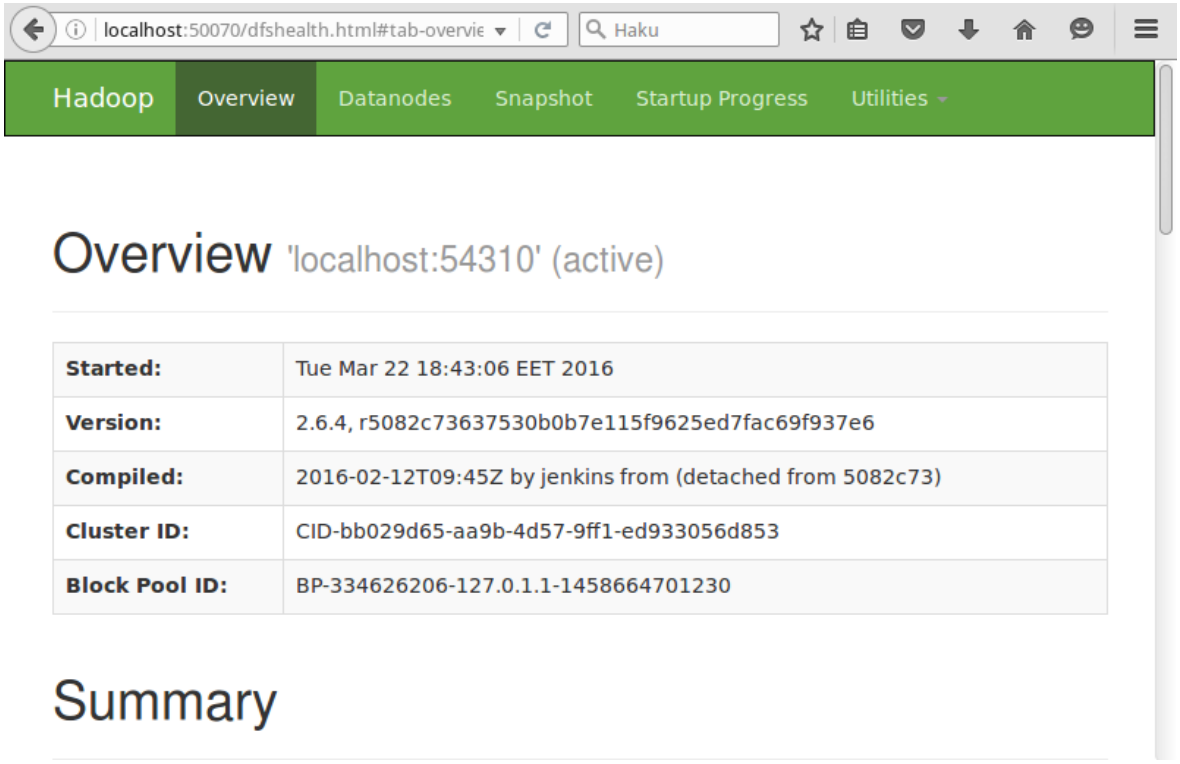
Kuva 10: Hadoopin käynnistymisen tarkastaminen

Toinen tapa tarkistaa on käyttää netstat:ia (kuva 12).

```
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-nodemanager-xubuntu.out
hduser@xubuntu:~$ netstat -plten | grep java
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp        0      0 0.0.0.0:50070           0.0.0.0:*               LISTEN      1001       157481      14439/ja
va
tcp        0      0 0.0.0.0:50010           0.0.0.0:*               LISTEN      1001       155493      14562/ja
va
tcp        0      0 0.0.0.0:50075           0.0.0.0:*               LISTEN      1001       156423      14562/ja
va
tcp        0      0 0.0.0.0:50020           0.0.0.0:*               LISTEN      1001       156426      14562/ja
va
tcp        0      0 127.0.0.1:54310         0.0.0.0:*               LISTEN      1001       157938      14439/ja
va
tcp        0      0 0.0.0.0:50090           0.0.0.0:*               LISTEN      1001       158869      14780/ja
va
tcp6       0      0 :::57936                :::*                    LISTEN      1001       163894      15056/ja
va
tcp6       0      0 :::8088                 :::*                    LISTEN      1001       158926      14931/ja
va
tcp6       0      0 :::13562                :::*                    LISTEN      1001       161427      15056/ja
va
tcp6       0      0 :::8030                 :::*                    LISTEN      1001       158916      14931/ja
va
tcp6       0      0 :::8031                 :::*                    LISTEN      1001       160971      14931/ja
va
tcp6       0      0 :::8032                 :::*                    LISTEN      1001       158922      14931/ja
va
tcp6       0      0 :::8033                 :::*                    LISTEN      1001       159532      14931/ja
va
tcp6       0      0 :::8040                 :::*                    LISTEN      1001       163901      15056/ja
va
tcp6       0      0 :::8042                 :::*                    LISTEN      1001       161428      15056/ja
va
hduser@xubuntu:~$ cd /usr/local/hadoop/sbin
```

Kuva 11: Netstat tarkastaminen

Kirjauduin selaimella osoitteeseen: http://localhost:50070 (kuva 13).



Kuva 12: Hadoop toiminnassa selaimella

6.3 Hiven asentaminen

Hivessä tiedoista luodaan virtuaalitauluja, joihin Hadoopin analysoima tieto kartoitetaan. Analysoitavien tiedostojen olisi hyvä olla puoli-rakenteellisia tiedostoja, esimerkiksi cvs-tiedostoja, jotta niille on helppo antaa rakenne Hive-tauluun.
Kun Hivelle on annettu tieto tiedoston kentistä, voidaan tiedoston sisältämä data ladata virtuaalitauluun. Tämän jälkeen siihen voidaan kohdistaa kyselyjä. Hive muuttaa kyselyt automaattisesti taustalla MapReduce-proseseiksi.
Tietoa voidaan tuoda myös ulkopuolisista lähteistä. Hive sisältää myös metatieto-osan, joka tallentaa tietoa luoduista virtuaalitauluista.

Aloitin tarkastamalla javan ja Hadoopin versiot (kuva 14). Näitä tarvitaan konfiguroinnissa, joten ne on hyvä olla muistissa.

Kuva 13: Javan ja Hadoopin versiot

Jatkoin asentamista lataamalla hiven Apachen sivuilta, versio 1.0.1. Koska latasin Hiven tervapallona, minun täytyi purkaa se ja siirtää oikeaan kansioon / usr / local / hive. Tarkastin myös, että Hadoopin kaikki demonit ovat toiminnassa (kuva 15), joten jatkoin asennusta.

Kuva 14: Hadoop on toiminnassa

Muokkasin .bashrc tiedostoon oikeat ympäristömuuttujat:

```
export HADOOP_HOME=/usr/local/hadoop/hadoop-0.20.1
export PATH=$PATH:$HADOOP_HOME/bin
export HIVE_HOME=/usr/local/hive/hive-0.9.0-bin
export PATH=$PATH:$HIVE_HOME/bin
```

Tein ”warehouse” kansion. Tämä on paikka, johon tallennetaan taulukon tai dataan liittyvät tiedot Hivessä.
”tmp” on paikka, johon tallennetaan väliaikaistiedostot. DFS / tmp käytetään pääasiassa väliaikaisena varastointi paikkana MapReduce käytön aikana. Tiedot poistuvat automaattisesti, kun MapReducen toiminnan suoritus on valmis. Annoin edellä luoduille kansioille luku- ja kirjoitusoikeudet.
Lisäsin Hadoop polun Hiveen. Tämän jälkeen pääsin testaamaan Hiven toimivuutta (kuva 16).

```
hduser@xubuntu:~$ hive

Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-common-1.0.1.jar!/hive-log4j.prope
rties
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/
impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/hive-jdbc-1.0.1-standalone.jar!/org/slf4j/impl/StaticLogg
erBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
hive> show tables;
OK
Time taken: 0.831 seconds
```

Kuva 15: Hive toiminnassa

6.4 Toimivuuden testaus

Tein ensin yksinkertaisen harjoituksen Hivellä. Latasin sivulta MapR.doc sivulta tekstitiedoston ja kokeilin toimivuutta.

Loin taulun ja latasin sample-table.txt-tiedoston tiedot.

```
CREATE TABLE web_log(viewTime INT, userid BIGINT, url
STRING, referrer STRING, ip STRING) ROW FORMAT DELIMITED FIELDS
TERMINATED BY '\t';
LOAD DATA LOCAL INPATH '/home/mapr/sample-table.txt' INTO TABLE
web_log;
```

Tein yksinkertaisen kyselyn, joka näytti kaikki taulukon tiedot.

Seuraavaksi tein kyselyn joka vastaa haluttuun merkkijonoon (kuva 17).

Ensimmäinen haku näyttää kaikki taulukon tiedot. Seuraava haku näyttää vain ne tiedot, jotka vastaavat haluttua merkkijonoa, tässä tapauksessa ’doc’.

```
SELECT web_log.* FROM web_log;
SELECT web_log.* FROM web_log WHERE web_log.url LIKE '%doc';
```

```
hive> CREATE TABLE web_log(viewTime INT, userid BIGINT, url STRING, referrer STRING, ip STRING) ROW FORMAT DELI
MITED FIELDS TERMINATED BY '\t';
OK
Time taken: 1.724 seconds
hive> LOAD DATA LOCAL INPATH '/home/xubuntu/Lataukset/sample-table.txt' INTO TABLE web_log;
Loading data to table default.web_log
Table default.web_log stats: [numFiles=1, totalSize=220]
OK
Time taken: 2.071 seconds
hive> SELECT web_log.* FROM web_log;
OK
1320352532      1001    http://www.mapr.com/doc http://www.mapr.com     192.168.10.1
1320352533      1002    http://www.mapr.com     http://www.example.com  192.168.10.10
1320352546      1001    http://www.mapr.com     http://www.mapr.com/doc 192.168.10.1
Time taken: 0.544 seconds, Fetched: 3 row(s)
hive> SELECT web_log.* FROM web_log WHERE web_log.url LIKE '%doc';
OK
1320352532      1001    http://www.mapr.com/doc http://www.mapr.com     192.168.10.1
Time taken: 0.201 seconds, Fetched: 1 row(s)
hive>
```

Kuva 16: Haut toimii

Kun sain todettua että Hive toimii, jatkoin Trafin avoimeen dataan.

7 Tulokset

Tutkimuskysymyksinä oli etsiä mahdollisimman monipuolista tietoa ajoneuvohuollolle Trafin avoimesta datasta.Projektin tuloksena sain tutkimuskysymysten asettajalle vastaukset kysymyksiin, millaisia ja minkä ikäistä autokantaa ajoneuvohuollon lähettyvillä on. Erityisesti he halusivat tietoa ajoneuvojen merkeistä, malleista, käyttöönottopäivämääristä ja henkilö- / pakettiautojen kappalemääristä lähialueilla.
Edellä mainituilla tiedoilla ajoneuvohuolto pystyy ennakoimaan esimerkiksi työvälineiden hankintaa, koulutusten tarpeellisuutta, markkinointia ja mahdollisesti jopa työvoiman lisäämistä.
Valitettavasti lopulliset tulosten hyödyt, joita Trafin avoimesta datasta sain, selviävät vasta pidemmällä aikavälillä.

8 Yhteenveto

Tietoa saadaan joka puolelta. Tätä tietomäärää hyödyntämällä ja analysoimalla yritykset voivat kehittää liiketoimintaansa.
Big datalta edellyttämät palvelut vaihtelevat paljon erilaisten liiketoimintojen ja toimialojen kesken. (Pervilä 2015.)

> ”Big Data muuttaa tietojärjestelmien investointirakennetta siten, että algoritmien ja analyysien tekemisestä tulee suurempi investoinnin osa-alue. Laitteet, käyttöjärjestelmät ja tietokannat jäävät huomattavasti pienemmälle painoarvolle. Resursseja jää enemmän laadukkaampien käyttöliittymien, visualisoinnin, analysoinnin ja algoritmien rakentamiseen.” (Vakkuri 2015.)

Datan yhdistely ja sen arvo kasvaa jatkuvasti, koska datassa olevien yhteyksien määrä kasvaa eksponentiaalisesti avoimen datan määrään nähden.
Lähitulevaisuudessa Avoin Data sisältää myös henkilökohtaista ja yksityistä big dataa, jaettuna kolmansille osapuolille. Henkilökohtainen genomidata tulee olemaan keskeinen sovellus tässä trendissä, tämä sisältää myös muita asioita, kuten henkilökohtaisen GPS-paikkadatan luovuttamisen kolmansille osapuolille mainosten kohdentamiseksi ja palveluiden mahdollistamiseksi.

Ohjelmistoteollisuus on asiantuntijan roolissa big datassa, ja sen vastuulla on evankelistojen käyttäminen potentiaalisten asiakkaiden ja kokonaisten liiketoiminta-alueiden kouluttamiseksi. Ohjelmistoteollisuuden on kommunikoitava Suomessa potentiaalisesta arvosta, jonka big data tuo saataville. Että voisimme antaa mielekästä ohjeistusta ja ohjausta, ohjelmistoteollisuuden täytyy pitää yllä intiimiä dialogia asiakasmarkkina-alueiden ja liiketoiminta-alueiden kanssa. (Keski-Valkama 2014.)

Pelkkä data ei monestikaan ole juuri minkään arvoista. Arvokkaaksi sen tekee prosessointi, yhdistely, analysointi ja sen mukanaan tuomat tulokset. Datasta tehdyt analytiikan tulokset määrittävät sen arvon.
Datan tuottamisessa voidaan törmätä myös siihen, että tuotetaan väärää dataa, jolloin väärin tuotetulla datalla voidaan muokata analyysin tuloksia. Esimerkiksi pörssikaupankäynnissä tuotetaan paljon osto- ja myyntitoimeksiantoja analytiikan pohjalta. Jos järjestelmään pääsisi joku murtautumaan, saisi selville käytettävien algoritmien logiikan ja pystyisi muuttamaan lähdetietoon riittävän vääristymän, hän saisi automatisoidun kaupankäynnin toimimaan itselleen edullisella tavalla. (Salo 2014, 52.)

Tulevaisuuden kauppa ja digitalisaatio kulkevat käsi kädessä. Kaikki kaupan alat tutkivat millä voidaan tarjota asiakkaalle parempi ostokokemus. Kun asiakkaalle pystytään tarjoamaan parempaa palvelua, kohdistetumpia tarjouksia, oikealla hinnalla, oikeaan aikaan ja

oikean kanavan kautta tällä pystytään pääsemään jo lähelle tavoitteita. Suurin kysymys onkin tällä hetkellä, miten valtavia tietomassoja käytetään oikein.
Kaupan alan on pystyttävä tunnistamaan se data, josta saadaan välitöntä hyötyä ja lisäarvoa. Haasteena on myös varmistaa tietojen eheys ja järjestelmä, ettei arkaluonteisia tietoja pääse vuotamaan.
Haasteeksi muodostuu myös asiantunteva työvoima, joka osaa käsitellä big dataa ja sen mukanaan tuomia haasteita. Yrityksellä täytyy olla kestäviä liiketoimintamalleja, joka on mukautettavissa analogisille että digitaalisille kuluttajaryhmille. (Tieto.)

Big dataa on käytetty pilvipalvelutarjoajien myyntiargumenttina pitkään, tähän on ihan selkeä syy. Global Pulse -projektissa todetaan, että big datan kannattajat ja skeptikot ovat molemmat väärässä. Big data ei tule ratkaisemaan kaikkia ihmiskunnan ongelmia. Se täyttää lupauksensa, jos ymmärrämme oikein sen rajoitukset, ja ymmärrämme tukea sillä muuta päätöksentekoa. (Tilastokeskus, 2012)

8.1 Pohdinta

Opinnäytetyön tavoitteena oli tutustua big dataan ja Apache Hadoopiin hieman laajemmin ja tehdä hakuja avoimesta datasta. Projektissa selvitettiin mitä big data on, missä sitä syntyy ja kuinka sitä voidaan käsitellä.
Big data ja pilvipalvelut kappaleessa käytiin hieman läpi pilvipalveluita yleensä, miten big data liittyy siihen ja mitkä ovat pilvipalveluiden riskit.
Apache Hadoop kappaleessa perehdyttiin hieman big datan käsittelyyn ja käytiin läpi työvälineitä.
Kappaleessa Adoop, Hive ja Avoin data asennettiin Hadoop ja Hive, kokeiltiin hakuja avoimesta datasta.

Opinnäytetyön tekeminen oli työlästä ja osittain hyvinkin haastavaa, koska minulla ei ollut ennalta kovin suurta tietopohjaa aiheesta. Osittain haasteelliseksi koin sen, että vaikka tietoa on runsaasti, sitä ei kuitenkaan ole. Kohtasin opinnäytetyötä tehdessäni, että vaikka tietolähteitä on paljon, melko monet lähteistä sisälsivät lähes identtistä tietoa aiheesta. Pelkästään oikeanlaisen tiedon hakuun meni runsaasti aikaa, tämä tuli pienenä yllätyksenä, koska big datasta on kuitenkin kirjoitettu tähän mennessä jo melko paljon.

Hadoopin ja Hiven asentaminen oli ajoittain työlästä. Jotkin ongelmista (liittyivät lähinnä konfigurointiin) olivat sellaisia, että niistä löytyi paljon tietoa, mutta ratkaisut täytyi lopulta hakea monesta eri lähteestä. Tämä kuitenkin auttoi ymmärtämään Hadoopin ja hiven rakennetta.

Avoimesta datasta tietojen kerääminen osoittautui helpommaksi mitä olin ennen projektia ajatellut. Haasteita oli toki tässäkin, mutta tietoa kohtaamistani ongelmista löysin runsaasti ja ongelmien ratkaisut löytyivät melko pienellä vaivalla.

Oman oppimisen kannalta työ oli erittäin onnistunut. Vain pieni osa lukemistani artikkeleista, kirjoista ja katsomistani videoista päätyivät lopulliseen työhön. Tällä sain kuitenkin hyvän perustietopohjan itselleni big datasta.

Lähteet

Apache hadoop. 2013. MapReduce Tutorial. hadoop. Luettavissa: https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html Luettu: 16.1.2016

Apache hadoop. 2016. Hadoop: Setting up a Single Node Cluster. hadoop. Luettavissa: http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/SingleCluster.html Luettu: 13.2.2016

Apprenda. IaaS, PaaS, SaaS (Explained and Compared). Luettavissa: https://apprenda.com/library/paas/iaas-paas-saas-explained-compared/ Luettu: 24.3.2016

Argillander, T. 2012. Miksi iso data on iso juttu?. Digital Media Finland. Luettavissa: http://www.digitalmedia.fi/miksi-iso-data-on-iso-juttu Luettu: 15.12.2015

Borthakur, D. 2013. HDFS Architecture guide. hadoop. Luettavissa: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html Luettu: 13.1.2016

Chakravart, A. 2012. Big Data – Hadoop from an Infrastructure Perspective. Cisco Blogs. Luettavissa: http://blogs.cisco.com/datacenter/big-data-hadoop-from-an-infrastructure-perspective Luettu: 26.1.2016

Chuin, M. Löffler, M. Roberts, R. 2010. The Internet of Things. McKinsey&Company. Luettavissa: http://www.mckinsey.com/industries/high-tech/our-insights/the-internet-of-things Luettu: 17.2.2016

Davenport, T. Three big benefits of big data analytics. SAS. Luettavissa: https://www.sas.com/en_ca/news/sascom/2014q3/Big-data-davenport.html Luettu: 3.2.2016

Dumbill, E. 2012. Big Data Now. O'ReillyMedia, Inc. Cambridge, s. 10.

Farnham, K. 2015. Verizon's ThingSpace Platform Facilitates Development of New Internet of Things Applications. InfoQ. Luettavissa: http://www.infoq.com/news/2015/11/verizon-thingspace Luettu: 9.3.2016

Helin, S. 2013. Asiakastiedon iso kuva – viisi vinkkiä big datan hyödyntämiseen. Steeri. Luettavissa: http://steeri.fi/fi/asiakastiedon-iso-kuva-%E2%80%93-viisi-vinkki%C3%A4-big-datan-hy%C3%B6dynt%C3%A4miseen Luettu: 6.1.2016

Hotti, M. 2012. Pikaperehdytys Big Dataan: Mikä on Apache Hadoop, entä Hive? Tivi. Luettavissa: http://www.tivi.fi/Arkisto/2012-11-28/Pikaperehdytys-Big-Dataan-Mik%C3%A4-on-Apache-Hadoop-ent%C3%A4-Hive-3196648.html Luettu: 6.1.2016

Hovi, A. 2014. Hadoop ja SQL – onnellinen avioliitto. Ari Hovi. Luettavissa: http://www.arihovi.com/hadoop-ja-sql-onnellinen-avioliitto/ Luettu: 2.1.2016

IBM. What is Hive. IBM. Luettavissa: https://www-01.ibm.com/software/data/infosphere/hadoop/hive/ Luettu: 19.1.2016

Ivoriofinland. 2013. Big data esitys. SlideShare. Luettavissa: http://www.slideshare.net/ivoriofinland/big-data-esitys-14112013-ivorio-oy Luettu: 20.12.2015

Keski-Valkama, T. 2014. Big Data Suomessa – NoSQL & Cloud. Cybercon Group. Luettavissa: http://www.cybercom.com/fi/Suomi/Yritys/Blogit/Blogit/Big-Data-Suomessa---NoSQL--Cloud/ Luettu: 8.3.2016

Keski-Valkama, T. 2014. Big Data Suomalaisessa Teollisuudessa ja Liiketoiminnassa. Cybercon Group. Luettavissa: http://www.cybercom.com/fi/Suomi/Yritys/Blogit/Blogit/Big-Data-Suomalaisessa-Teollisuudessa-ja-Liiketoiminnassa/ Luettu: 4.1.2016

Laukkanen, J. 2014. Big Datan analysointi. cs-helsinki. Luettavissa: https://www.cs.helsinki.fi/u/jjlaukka/semma_dev.pdf Luettu: 8.3.2016

MapR.doc. MapR Overwiew. Luettavissa: http://doc.mapr.com/display/MapR/MapR+Overview Luettu: 14.2.2016

Meriläinen-Tenhu, M. 2015. Big Data: Isot aineistot esillä Tekniikan päivillä 2015. Helsingin yliopisto. Luettavissa: http://www.helsinki.fi/ml/ajankohtaista/2015/2015bigdata.html?rss=mltdk Luettu: 4.1.2016

Mortar Cocs. Pig Help and Resousces. Mortar Docs. Luettavissa: http://help.mortardata.com/technologies/pig/pig_help_and_resources Luettu: 23.1.2016

Mäkinen, K. 2014. Big data – tiedon käsittelyn seuraava mullistus. Paikkatietoikkuna. Luettavissa: http://www.paikkatietoikkuna.fi/web/fi/positio_1_2014_murroksessa_big_data Luettu: 6.1.2016

Penchikala, S. 2015. Big Data Processing whit Apache Spark. InfoQ. Luettavissa: http://www.infoq.com/articles/apache-spark-introduction Luettu: 18.1.2016

Pervilä, M. 2015. Big data leviää it-osastojen ulkopuolelle – CIO soittaa toista viulua. Tivi. Luettavissa: http://www.tivi.fi/CIO/big-data-leviaa-it-osastojen-ulkopuolelle-cio-soittaa-toista-viulua-3483939 Luettu: 5.1.2016

Rastas, T, Asp, E. 2014. Big datan hyödyntäminen. Liikenne- ja viestintäministeriö. Luettavissa: http://www.lvm.fi/julkaisu/4417803/big-datan-hyodyntaminen
Luettu: 22.12.2015.

Remes, M. 2013. Big datasta kaikki hyöty irti. View, 8-9. Luettavissa: http://issuu.com/kpmgfinland/docs/asiakaslehti-view-1-2013/8?e=6625711/1463933 Luettu: 10.12.2015

Rubens, P, 2014. Can big data crunching help feed the world? BBC. Luettavissa: http://www.bbc.com/news/business-26424338 Luettu: 15.12.2015

Rubin, J. 2013. Survey Demonstrates The Benefits Of Big Data. Forbes. Luettavissa: http://www.forbes.com/sites/forbesinsights/2013/11/15/survey-demonstrates-the-benefits-of-big-data/#62005d2127a4 Luettu: 7.3.2016

Salo, I. 2014. Big data & pilvipalvelut. Docendo Oy: Jyväskylä, s. 52.

Salo, I. 2013. Big data tiedon vallankumous. Docendo Oy: Jyväskylä, s. 25.

Tieto. Big datasta tulee smart dataa – tulevaisuuden kauppa on jo täällä. Tieto. Luettavissa: http://www.tieto.fi/nakemyksia-ja-visioita/big-datasta-tulee-smart-dataa-tulevaisuuden-kauppa-on-jo-taalla Luettu: 10.3.2016

Tilastokeskus. 2012. Iso data -suuret lupaukset ja pullonkaulat. Luettavissa: http://www.stat.fi/artikkelit/2012/art_2012-07-04_001.html Luettu: 24.3.2016

Vakkuri, M. 2013. Big Data muuttaa maailmaa. Talouselämä. Luettavissa: http://www.talouselama.fi/kumppaniblogit/tieto/big-data-muuttaa-maailmaa-3440603 Luettu: 28.12.2015

Vellore, T. 2015. Implementing Real Time Trending Engine on Clickstream data using Flume and Spark Streaming. Techkites blogspot. Luettavissa: http://techkites.blogspot.fi/2015/02/implementing-real-time-trending-engine.html Luettu: 31.1.2016

Yli-Pietilä, M. Big data – määritelmiä. Big data. Luettavissa: http://www.bigdata.fi/artikkelit/big-datan-m%C3%A4%C3%A4ritelm%C3%A4 Luettu: 7.12.2015.

Liite 1. Hadoop asennuksen komennot

```
$ sudo apt-get install default-jdk
```

```
$ java -version
```

```
$ sudo addgroup hadoop
```

```
$ sudo adduser --ingroup hadoop hduser
```

```
$ sudo adduser hduser sudo
```

```
$ su hduser
```

```
$ ssh-keygen -t rsa -P ""
```

```
$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

```
$ ssh localhost
```

```
$ wget http://mirrors.sonic.net/apache/hadoop/common/hadoop-2.6.4/hadoop-2.6.4.tar.gz
```

```
$ tar xvzf hadoop-2.6.4.tar.gz
```

```
$ sudo mv * /usr/local/hadoop
```

```
$ sudo chown -R hduser:Hadoop /usr/local/hadoop
```

```
$ update-alternatives --config java
```

```
nano ~/.bascrc
```

```
source ~/.bascrc
```

```
$ nano /usr/local/hadoop/hadoop-2.6.4/etc/hadoop/hadoop-env.sh
```

```
$ nano /usr/local/hadoop/hadoop-2.6.4/etc/hadoop/core-site.xml
```

```
$ nano /usr/local/hadoop/hadoop-2.6.4/etc/hadoop/mapred-site.xml.template
/usr/local/hadoop/hadoop-2.6.4/etc/hadoop/mapred-site.xml
```

```
$ sudo mkdir -p /usr/local/hadoop_store/hdfs/namenode
$ sudo mkdir -p /usr/local/Hadoop_store/hdfs/datanode
$ sudo chown -R hduser:Hadoop /usr/local/Hadoop_store
```

```
$ hadoop namenode -format
```

```
$ cd /usr/local/Hadoop/sbin
```

```
$ start-all.sh
```

```
$ jps
```

Liite 2. Hive asennuksen komennot

```
$ java -version
```

```
$ tar -xzvf hive-1.0.1.bin.tar.gz
```

```
$ sudo mkdir /usr/local/hive
```

```
$ mv hive-1.0.1-bin /usr/local/hive
```

```
$ jps
```

```
$ nano ~/.bashrc
```

```
$ hadoop fs .mkdir /user/hive/warehouse
```

```
$ hadoop fs -mkdir /tmp
```

```
$ hadoop fs -chmod g+w  /user/hive/warehouse
```

```
$ hadoop fs -chmod g+w /user/tmp
```

```
$ hive
```