SAVONIA

OPINNÄYTETYÖ - **AMMATTIKORKEAKOULUTUTKINTO**
TEKNIIKAN JA LIIKENTEEN ALA

HADOOPIN TOTEUTUS VIRTUAALIYMPÄRISTÖSSÄ

Implementation of Hadoop in a Virtual Environment

TEKIJÄ: Ville Heikkinen

SAVONIA-AMMATTIKORKEAKOULU

OPINNÄYTETYÖ
Tiivistelmä

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma/Tutkinto-ohjelma Tietotekniikan koulutusohjelma	
Työn tekijä(t) Ville Heikkinen	
Työn nimi Hadoopin toteutus virtuaaliympäristössä	
Päiväys 15.2.2017	Sivumäärä/Liitteet 29/15
Ohjaaja(t) lehtori Keijo Kuosmanen, lehtori Jussi Koistinen	
Toimeksiantaja/Yhteistyökumppani(t) Savonia-ammattikorkeakoulu	

Tiivistelmä

Tämän opinnäytetyön aiheena oli tutkia sekä toteuttaa Hadoop-klusteri Savonia-ammattikorkeakoulun virtuaaliympäristössä. Opinnäytetyön aihetta oli tärkeä tutkia, koska Savonialla investoitiin kahteen Big Data -palvelimeen, joita on tarkoitus hyödyntää sekä koulutus- että hankekäytössä.

Opinnäytetyön teko aloitettiin tutustumalla Big Datan sekä Hadoopin teoriaan sekä Hadoopin vaatimiin komponentteihin. Näitä ovat esimerkiksi HDFS, SSH ja MapReduce. Hadoopin yhteydessä keskeinen käsite on myös virtualisointi, joka mahdollistaa suurien Hadoop-klusterien teon virtuaalikoneiden avulla. Tässä opinnäytetyössä virtuaalikoneiden alustana toimi Big Data -palvelimille asennettu Microsoft Hyper-V.

Opinnäytetyön käytännön osuudessa luotiin tarkat Hadoopin asennusohjeet sekä tulevaa käyttöä varten valmiit yhden sekä monen solmun klusteri-templeitit. Ohjeet sekä valmiit templeitit luotiin Ubuntun käyttöjärjestelmälle. Hadoop-ympäristön luonnin jälkeen tutkittiin Big Data -palvelimien yhteyksiä. Big Data -palvelimien lisäksi käytössä oli varapalvelin. Varapalvelimelle luotiin suunnitelma niin, että saataisiin vähennettyä turhaa kuormaa Big Data -palvelimilta.

Työn lopputulos oli erittäin onnistunut. Asennusohjeet testattiin ja todettiin, että niiden avulla asennus on mahdollista toteuttaa onnistuneesti. Klusterin IP-osoite -testit osoittivat sen, että virtuaalikoneet saavat IP-osoitteet Savonian laboratorioverkon kautta. Tästä syystä klusterin solmut voivat olla eri palvelimilla. Opinnäytetyön alkupalaverissa asetettuihin tavoitteisiin päästiin.

Avainsanat
Big Data, Hadoop, Virtualisointi

SAVONIA UNIVERSITY OF APPLIED SCIENCES

THESIS
Abstract

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Ville Heikkinen			
Title of Thesis Implementation of Hadoop in a Virtual Environment			
Date	15 February 2017	Pages/Appendices	29/15
Supervisor(s) Mr Keijo Kuosmanen, Lecturer and Mr Jussi Koistinen, Lecturer			
Client Organisation /Partners Savonia University of Applied Siences			

Abstract

The subject of this thesis was to study and implement Hadoop in a virtual environment. The thesis was done for Savonia University of Applied Sciences. It was important to study this subject, because Savonia has invested in two Big Data servers, which are ment to be used for both educational and project purposes.

This thesis was started by studying the theory of Big Data and Hadoop. It was also important to study different components used by Hadoop, for example HDFS, SSH and MapReduce. An important concept when talking about Hadoop is virtualization. Virtualization allows easy creation for large Hadoop clusters. In this thesis the virtual machines were created in Microsoft Hyper-V.

Detailed instructions on how to install Hadoop and templates of the already installed Hadoop clusters were created in the practical part of this thesis. Instructions and templates were created using the Ubuntu operation system. After this the connection between two servers was tested. Alongside two Big Data servers a back up-server was installed. A plan was created to this back up-server on how to reduce the load from the Big Data servers.

As a result of this thesis, a detailed instraction of how to install Hadoop was successfully created. Tested showed that Hadoop could be installed using the instructions. Server tests proved that because the virtual machines got their IP addresses from Savonia's laboratory network, cluster nodes could be created in both servers and they would still work together.

Keywords
Big Data, Hadoop, Virtualization

ESIPUHE

Haluan kiittää lehtori Keijo Kuosmasta saamastani opinnäytetyön aiheesta. Keijo toimi myös erittäin hyvänä ohjaana opinnäytetyölle. Ensimmäisestä palaverista lähtien aihe tuntui erittäin mielenkiintoiselta, koska Big Data ja Hadoop olivat tuntemattomia käsitteitä. Työ opettikin erittäin paljon ja tavoitteena on jatkaa aiheen opiskelua valmistumisen jälkeen.

Iso kiitos kuuluu myös perheelle sekä läheisille. Olen saanut valtavasti tukea näiden kolmen vuoden aikana.

Kuopiossa 15.2.2017
Ville Heikkinen

SISÄLTÖ

TERMISTÖ

Big Data = Big Data tarkoittaa erittäin suurta data määrää, jota tutkitaan ja käytetään eri tarkoituksiin.

Hadoop = Hadoop on ohjelmistoprojekti, jonka avulla prosessoidaan suuria datamääriä, Big Dataa.

Hadoop Distributed File System (HDFS) = HDFS on tietojärjestelmä, jolla on mahdollisuus tallettaa suuria datamääriä hajautetussa järjestelmässä.

Klusteri = Englanniksi Cluster. Klusteri tarkoittaa ryhmää järjestelmiä, jotka toimivat yhtenä järjestelmänä.

Solmu = Englanniksi Node. Solmu tarkoittaa klusterin osaa, konetta.

Datalohkot = Englanniksi Blocks. Hadoopiin syötetty data jaetaan datalohkoihin.

NameNode = NameNode-prosessi toimii master-solmussa ja se säätelee HDFS:än toimintaa.

DataNode = DataNode-prosessi toimii slave-solmuissa ja säätelee datalohkoja.

Secondary NameNode (SNN) = Secondary NameNode-prosessi toimii NameNoden varmuuskopiona.

JobTracker = JobTracker-prosessi toimii luodun MapReduce-työn toimeenpanijana.

TaskTracker = TaskTracker-prosessi hoitaa yksittäisten MapReduce-tehtävien toimeenpanon slave-solmuissa.

MapReduce = MapReducen avulla kirjoitetaan ja toteutetaan töitä Hadoopissa.

SSH (Secure Shell) = SSH-protokollan avulla luodaan suojatut yhteydet solmujen välillä.

Hyper-V = Microsoft Hyper-V:n avulla luoda virtuaalikoneita.

Ubuntu = Ubuntu on Linux-pohjainen käyttöjärjestelmä.

1 JOHDANTO

Tämän opinnäytetyön aiheena on tutkia Big Datan ja Hadoopin teoriaa sekä luoda toimiva Hadoop-ympäristö ohjeineen. Opinnäytetyön tilaaja on Savonia-ammattikorkeakoulun lehtori Keijo Kuosmanen. Hän toimii samalla työn ohjaajana. Aihe on tärkeä, koska Savonia-ammattikorkeakouluun on asennettu uudet Big Data -palvelimet koulutus- sekä hankekäyttöä varten. Opinnäytetyön aihe on rajattu niin, että siitä saadaan mahdollisimman suuri hyöty kumpaakin käyttötarkoitusta varten. Tämä työ tulee toimimaan ohjekirjana Big Datan sekä Hadoopin perusteita varten.

Työ aloitetaan tutustumalla Big Datan sekä Hadoopin teoriaan. Teorian jälkeen tarkastellaan Hadoopin rakennetta sekä Hadoop-ympäristön vaatimia komponentteja. Teoriaosuuden jälkeen luodaan valmiit Hadoop-ympäristöt koulun Big Data -palvelimelle. Tämän jälkeen ympäristöstä tehdään asennusohjeet sekä luodaan asennetuista virtuaalikoneista valmiit templeitit tulevaa käyttöä varten. Viimeisenä vaiheena tutkitaan, kuinka saadaan parhaiten hyödynnettyä kahta identtistä Big Data -palvelinta keskenään sekä varapalvelinta Big Data -palvelinten rinnalla.

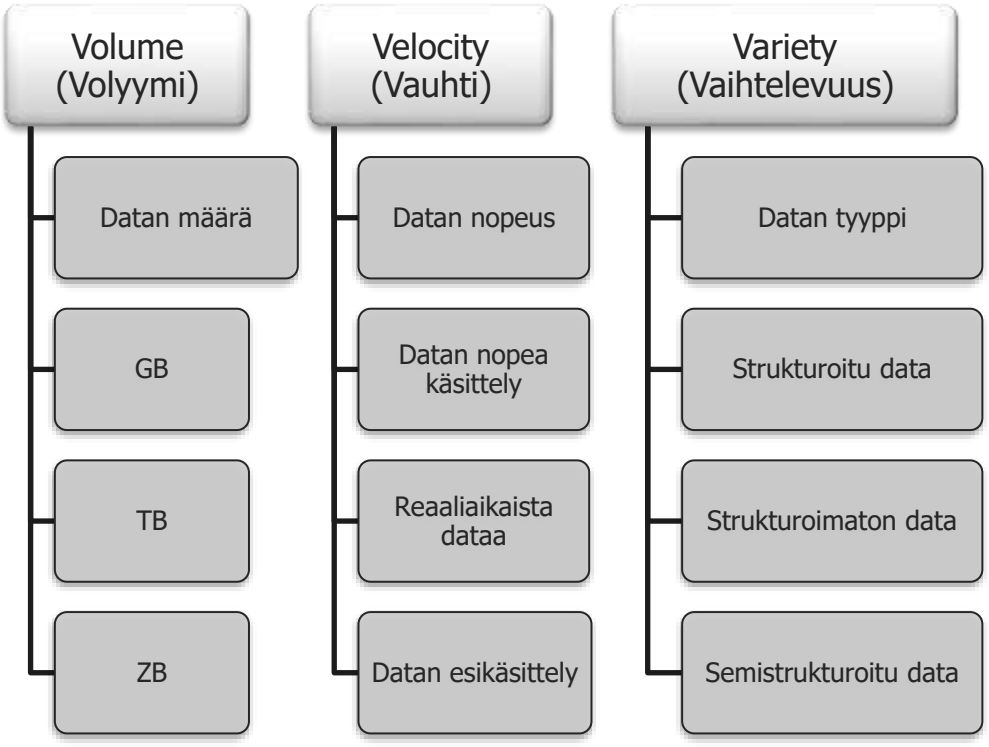
2 BIG DATA

Nykypäivänä tietoa syntyy minuutissa niin paljon (Kuva 1), että 2000-luvulla saatua dataa on enemmän kuin koko tunnetun historiamme aikana. Tällä hetkellä digitaalisen datan määrän yksikkönä toimii tsettatavu. Yksi tsettatavu on 1 000 000 000 teratavua. Tämä tarkoittaa 200 miljoonaa kappaletta muistiltaan 5 TB kokoista ulkoista kovalevyä. Vuonna 2005 tästä ilmiöstä käytettiin ensimmäisen kerran nimitystä Big Data. Tällä hetkellä valloillaan oleva Big Datan vallankumous alkoi kuitenkin vasta vuonna 2011. (Salo, 2014, s. 26)

Kuva 1. Minuutissa tuotettua dataa muun muassa sosiaalisessa mediassa

2.1 Volume, Velocity, Variety

Datan määrä ja laatu kasvaa niin nopeasti, että Big Datan käsittämisen helpottamiseksi on otettu käyttöön kolme V-kirjainta: volume, velocity ja variety, suomeksi *volyymi*, *vaihtelevuus* ja *vauhti*. Data on helppo määritellä Big Dataksi, kun se täyttää kaikki kolme määritelmää (Kuva 2). (Salo, 2014, s. 26)



Kuva 2. Big Datan kolme ulottuvuutta: volyymi, vauhti sekä vaihtelevuus

2.1.1 Volume (Volyymi)

Volyymi tarkoittaa datan suurta määrää. Nykypäivänä tuotetaan yhteensä noin 2,5 eksatavua (2,5 EB = 2 500 000 TB) uutta dataa päivässä ja arviolta maailmassa on tällä hetkellä 5-6 tsettatavua tietoa (VCloudNews, 2016). Vielä vuonna 2011 sitä oli vain 2 ZB. (Salo, 2014, s. 26)

Nykypäivänä yhä useampi laite tuottaa dataa siihen lisättyjen sensoreiden avulla. Näistä laitteista käytetään yleisnimitystä Internet of Things (IoT). IoT tulee lisäämään datan määrää räjähdysmäisesti, koska sensoreiden avulla tullaan keräämään dataa mitä erilaisimmista asioista. Esimerkki tällaisestä hyötykäytöstä on koripalloon lisättyt sensorit. Koripallon heittäjä saa valmistajan sovelluksesta heti heiton jälkeen näkyville tietoa pallon lentoradasta, kierteestä sekä heiton nopeudesta. Nämä tiedot auttavat sekä harjoittelijaa että koripallon valmistajaa. Kun harjoittelija saa välitöntä palautetta heitostaan, menee samat tiedot myös valmistajalle, joka voi käyttää niitä haluamallaan tavalla. IoT:sta on tässä esimerkissä suurta hyötyä koripallon valmistajalle. Sensoreiden avulla se saa suoraa palautetta tuotteestaan. (Kalyvas & Overly, 2015, s. 4)

Toinen esimerkki on aktiivisuusrannekkeet. Ne antavat käyttäjälle tietoa päivässä kävellyistä askelista, paikallaan olon ajasta sekä unen laadusta. Yleisesti ottaen, mitä kalliimpi ranneke on, sitä enemmän siinä on sensoreita, jotka tuottavat dataa.

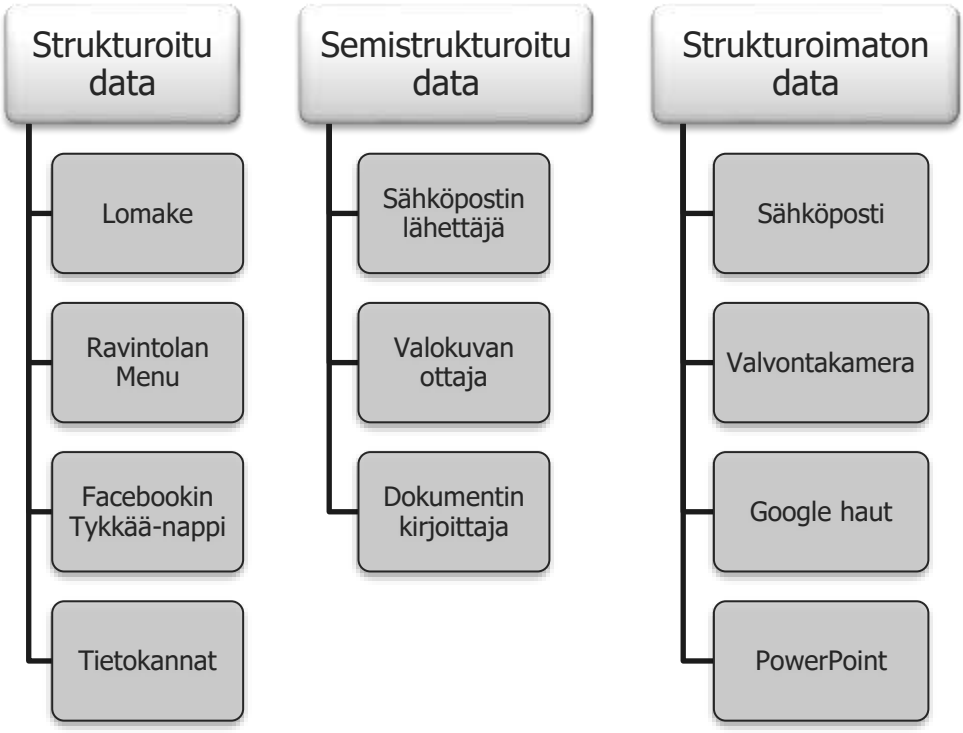
2.1.2 Velocity (Vauhti)

Toinen V, vauhti viittaa datan syntymisen nopeuteen ja sen nopeaan käsittelyyn. Datamäärien kasvaessa on entistä tärkeämpää osata esikäsitellä datavirrat. Näin saadaan hyödynnettyä Big Dataa tehokkaimmin päätöksenteossa. (Salo, 2014, s. 27)

Datan nopea käsittely mahdollistaa nopean päätöksen teon. Nopea päätöksien teko on erittäin tärkeää esimerkiksi kaupankäynnissä. Prosessoidulla datalla voidaan havaita mahdollisuuksia sekä uhkia. Data voi kertoa kaupantekijälle, joka voi olla ihminen tai automaatio, pitäisikö ostaa vai myydä. (Hansen, 2013)

2.1.3 Variety (Vaihtelevuus)

Kolmas V, vaihtelevuus tarkoittaa datan laatua. Tuotettu data voi olla strukturoitua tai strukturoimatonta. Näiden kahden rajapinnan väliltä löytyy vielä semistrukturoitu data (Kuva 3). Data voi olla esimerkiksi peräisin sensorista, lentokoneesta, laivasta tai sosiaalisesta mediasta (Facebook, Twitter yms.). (Salo, 2014, s. 27)



Kuva 3. Esimerkkejä strukturoidusta, semistrukturoidusta sekä strukturoimattomasta datasta

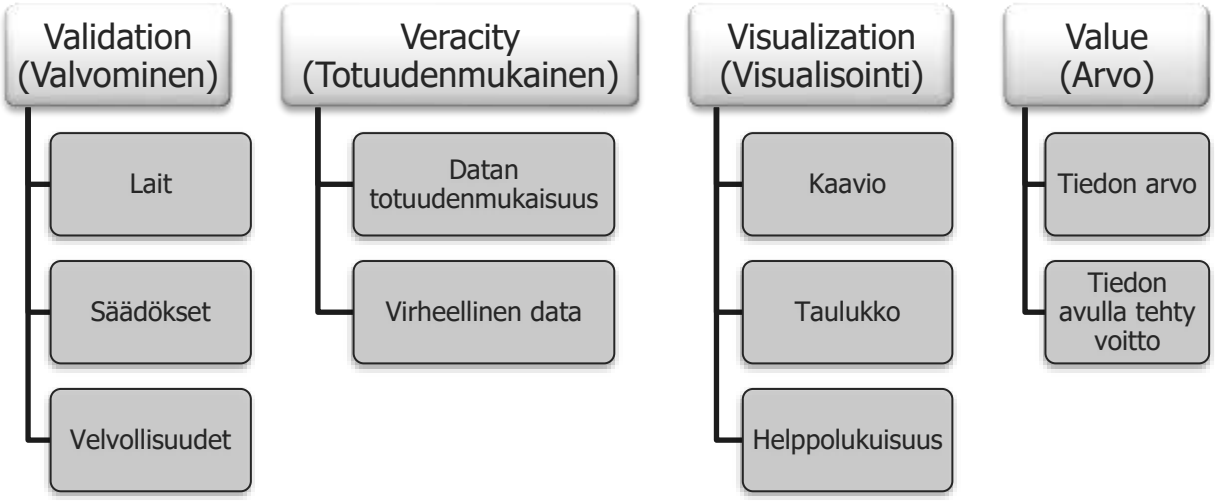
Strukturoitu data on helppoa syöttää, varastoida ja analysoida. Strukturoitu data tallennetaan tiettyihin kenttiin, jotka voivat olla eri tyyppisiä. Datatyyppejä ovat esimerkiksi numeerinen, valuutta, nimi, päivämäärä tai osoite. (Beal, ei pvm)

Strukturoimatonta dataa ei voida luokitella tai sovittaa siistiin muotoon. Esimerkkinä tästä voivat olla sähköpostiviestit tai PowerPoint-esitykset. (Beal, ei pvm)

Semistrukturoitu data on strukturoidun ja strukturoimattoman datan välimuoto. Semistrukturoitu data on strukturoitua dataa ilman tiukkoja rakenteita. Esimerkiksi sähköpostiviestissä teksti on strukturoimatonta dataa. Lähettejä, päivämäärä ja aika ovat semistrukturoitua dataa. (Beal, ei pvm)

2.2 Validation, Veracity, Visualization ja Value

Kolmen alkuperäisen V-kirjaimen lisäksi on muodostunut lisää ilmaisuja (Kuva 4), joiden avulla Big Dataa yritetään tehokkaammin kuvata ja määritellä. Uusia määrittelyjä ovat validation (valvominen), veracity (totuudenmukaisuus), visualization (visualisointi) ja value (arvo).



Kuva 4. Uusia Big Dataa kuvailevia ilmaisuja

Yrityksen Big Data -strategiassa tulee olla valvomisen taso. Valvomisella tarkoitetaan sitä, että yrityksien tulee valvoa, kuinka lait, säädökset ja sopimuksen mukaiset velvollisuudet vaikuttavat esimerkiksi Big Data -järjestelmän arkkitehtuuriin, Big Data -algoritmien suunnitteluun sekä saadun datan tallentamiseen ja jakeluun. (Kalyvas & Overly, 2015, s. 5)

Käsiteltävän datan tulee olla tarkkaa ja totuudenmukaista. Parhaan tuloksen saavuttamiseksi virheellisen datan pitäminen poissa järjestelmistä on erittäin tärkeää. Esimerkiksi markkinointijärjestelmässä on turhaa pitää yhteystietoja, jotka on virheellisesti täytetty tai luotu tarkoituksella väärillä tiedoilla. (DeVan, 2016)

Suuria tekstimääriä sisältävien raporttien tekeminen ei nykypäivänä ole enään järkevää. Kaavioiden, taulukoiden ja kuvaajien käyttäminen isojen datamäärien visualisointiin on paljon parempi vaihtoehto. Big Datan avulla saadaan suuresta tekstimäärästä nopeasti oleellinen tieto esille esimerkiksi taulukko- tai kaaviomuotoon. (DeVan, 2016)

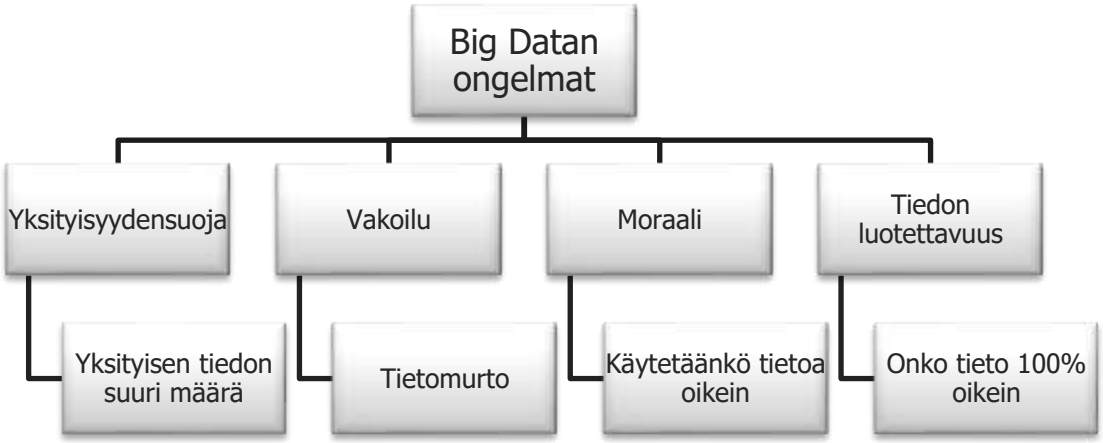
Big Datan arvo havaitaan sen jälkeen, kun on otettu kantaa volyymiin, vauhtiin, vaihtelevuuteen, vahvistamiseen, totuudenmukaisuuteen sekä visualisointiin. Yrityksen tulee varmistua siitä, että se saa analysoiduista tiedoista mahdollisimman paljon hyötyä päätöksentekoprosesseihinsa. (DeVan, 2016)

2.3 Big Datan ongelmat

Big Datan ongelmiin kuuluvat ollennaisena osana yksityisyydensuoja, vakoilu, moraali sekä tiedon luotettavuus (Kuva 5). Pelkästään prosessoimattoman datan päätyminen vääriin käsiin, voi olla kohtalokasta yritykselle, prosessoidusta datasta puhumattakaan. Tästä syystä on varmistettava, että data ei pääse missään elinkaarensa vaiheessa asiattomien haltuun. (Salo, 2014, ss. 50-52)

Big Datan datamäärät ovat niin suuria, että tietomurto tai vakoilu keskittyy yleensä prosessoituun dataan. Data, joka ei ole yrityksen omistamaa, on se yleensä kaikkien käytettävissä. Eli kuka vain voi sitä kerätä itselleen. Tietomurtojen kannalta oleellista on kuinka tai miten dataa on käsitelty. Prosessoitujen tietojen avulla tietomurron tekijät voivat päästä käsiksi hyvin arkaluontoiseen tietoon. (Salo, 2014, ss. 50-52)

Tietouhkien vaarana ei ole ainoastaan datan varastaminen. Hyvänä esimerkkinä pörssikaupankäynnin järjestelmä. Jos järjestelmään päästään murtautumaan, on mahdollista vääristää päätöksentekoalgoritmeja. Päätöksentekoalgoritmit tuottaisivat väärää tietoa, joka aiheuttaisi vääriä osto- tai myyntitoimeksiantoja. (Salo, 2014, s. 52)



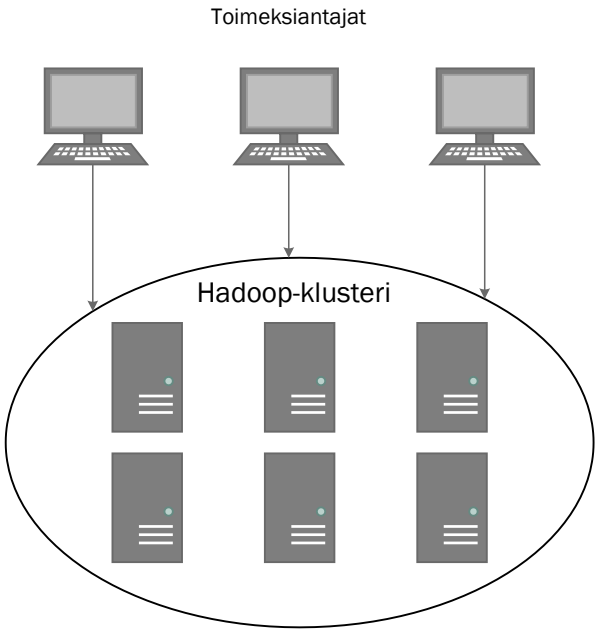
Kuva 5. Esimerkkejä Big Datan ongelmista

3 APACHE HADOOP

Hadoop on avoimen lähdekoodin ohjelmaprojekti, jolla pystytään prosessoimaan suuria datamääriä. Nykypäivänä suositaan järjestelmää, jossa kytketään monta pienempää järjestelmää yhdeksi isoksi hajautetuksi järjestelmäksi (distributed systems). Hajautettu järjestelmä tarkoittaakin ryhmää koneita, jotka toimivat ja esiintyvät yhtenä isona järjestelmänä (Kuva 6). (Lam, 2011, s. 4)

Hadoopia luonnehditaan helppopääsyiseksi (accessible), koska se toimii isossa klusterissa. Se on myös kestävä järjestelmä (robust), joka on suunniteltu selviytymään laitteistovioista. Hadoop-klusteri on helppo skaalata (scalable) tarpeiden mukaan. Se mahdollistaa yksinkertaisten (simple) koodien kirjoittamisen ja töiden toteuttamisen. (Lam, 2011, s. 4)

Tänä päivänä käytössä oleva järjestelmä vaatii noin kolme tuntia 4 TB tiedoston lukemiseen. Hadoopin avulla sama datan määrä (4 TB) jaetaan pienempiin, yleensä 64 MB datalohkoihin (blocks). Nämä lohkot levitetään kuvan 6 mukaisen klusterin (cluster) sisällä oleville järjestelmille Hadoop Distributed File System (HDFS) avulla. Klusterin sisällä olevat koneet lukevat datan suuremmalla suoritusteholla, kuin mihin yksi kone pystyy. Tavallisista virtuaalikoneista luotu klusteri on myös halvempi, kuin yksi korkean tason järjestelmä. Hadoop eroaakin muista hajautetuista järjestelmistä siten, että se ei siirrä isoa datamäärää paikasta toiseen, vaan keskittyy koodin siirtämiseen datan luokse. Puhutaan niin sanotusta move-code-to-data filosofiasta. (Lam, 2011, s. 6)



Kuva 6. Hadoop-klusterissa rinnakkain olevat koneet säilövät ja käsittelevät samanaikaisesti suuria datamääriä. Toimeksiantajakoneet (client) lähettävät klusteriin tehtäviä töitä. (Lam, 2011, s. 6)

Hadoop on Apache-lisenssoitu. Tämä tarkoittaakin sitä, että sen lähdekoodi on jokaisen käytettävissä. Maksullisia Hadoop-jakeluita tarjoavat muun muassa Cloudera, Hortonworks, MapR, Pivotal, Amazon, IBM ja Microsoft. Hadoopin voi kuitenkin asentaa myös itse. Tällöin luotu Hadoop-klusteri on ilmainen. (Salo, 2014, ss. 72, 86)

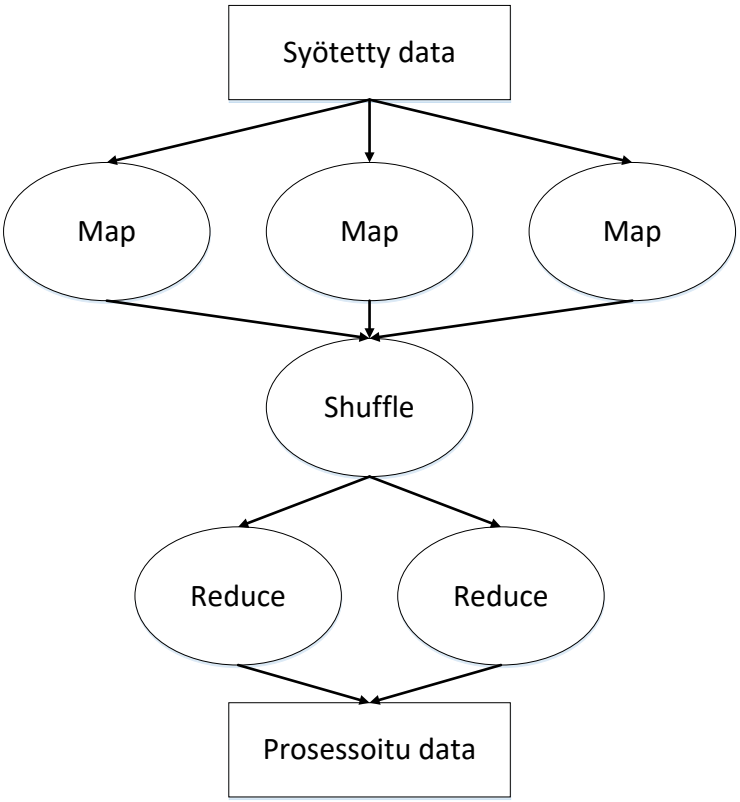
3.1 Hadoop Distributed File System (HDFS)

HDFS on Javalla kirjoitettu tiedostojärjestelmä, joka on suunniteltu suuren mittakaavan data prosessointiin Hadoop-klusterissa. HDFS pystyy tallettamaan isoja datamääriä, esimerkiksi 100 TB yhteen isoon tiedostoon. Tämä tiedosto on kuitenkin jaettu monelle eri solmulle. (Lam, 2011, s. 38)

Hadoopin sisällä on shell-komentoja, joiden avulla pystytään suoraan kommunikoimaan HDFS:n kanssa. Komento ”bin/hdfs dfs -help” listaa kaikki Hadoopin tukemat HDFS-komennot. (Lam, 2011, s. 38)

3.2 Hadoop MapReduce

MapReduce on Java pohjainen ohjelma, jonka avulla kirjoitetaan ja toteutetaan töitä Hadoopissa. Näiden MapReduce-töiden avulla pystytään käsittelemään isoja datamääriä suurissa klustereissa. MapReduce sisältää kolme tärkeää tehtävää (Kuva 7). Ensimmäisenä on Map-vaihe, joka hoitaa suodatuksen. Esimerkiksi nimien järjestämisen etunimen mukaan. Seuraavana on vuorossa Shuffle-vaihe. Viimeisenä suoritetaan Reduce-vaihe, joka hoitaa yhteenvedeot. Esimerkiksi sanojen lukumäärän tekstissä. (Lam, 2011, s. 42)

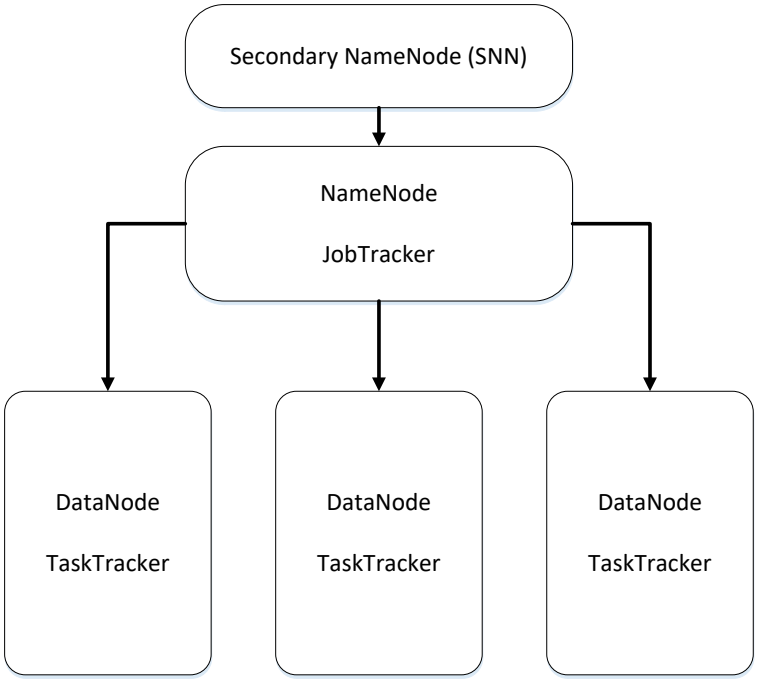


Kuva 7. MapReduce-työn kulku (Lam, 2011, s. 42)

Syötetty työ jaetaan solmuille, joissa Map-tehtävät työskentelevät sille osoitetulla datalohkolla. Tämän jälkeen Mapper lähettää datan Shuffle-osiolle, joka työskentelee solmujen välissä ja jakaa Map-periltä saadun datan Reduce-tehtäville. (Lam, 2011, s. 42)

3.3 Hadoop-klusteri (Cluster)

Täysin konfiguroidussa Hadoop-klusterissa on käynnissä useita prosesseja samanaikaisesti (Kuva 8). Prosessit ovat NameNode, DataNode, Secondary NameNode, JobTracker ja TaskTracker. Jokaisella prosessilla on oma roolinsa klusterin sisällä. Hadoop-klusterissa NameNode, Secondary NameNode sekä JobTracker ovat master-prosesseja. DataNode ja TaskTraker ovat slave-prosesseja. Vielä tarkemmin määriteltynä NameNode, DataNode sekä Secondary NameNode kuuluvat HDFS-prosesseihin. JobTracker sekä TaskTracker kuuluvat MapReduce-prosesseihin. (Lam, 2011, s. 21)



Kuva 8. Hadoop-klusterin rakenne (Lam, 2011, s. 21)

3.3.1 NameNode

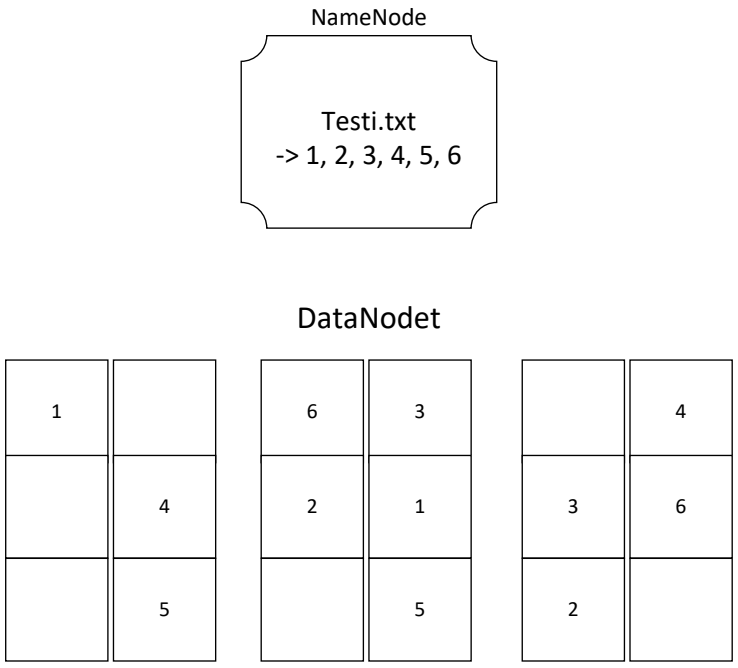
Tärkein prosessi Hadoopin sisällä on NameNode. NameNode toimii Hadoop Distributed File System (HDFS) masterina. NameNode on HDFS:n kirjanpitäjä. Se pitää kirjaa siitä, kuinka HDFS:ään luotu tiedosto on jaettu datalohkoihin sekä mitkä slave-solmut säilövät jaettuja lohkoja. Tämän lisäksi NameNode pitää huolta järjestelmän yleiskunnosta. (Lam, 2011, s. 22)

NameNode toimii yhdellä palvelimella eikä se jakaudu kuten DataNode tai TaskTracker. Tästä syystä se on Hadoopin ainoa kohta, joka vikatilanteessa aiheuttaa koko järjestelmän toimimattomuuden. Virhetilanteiden ehkäisemiseksi kannattaa NameNode pitää itsenäisessä solmussa ilman muita prosesseja. NameNodella on sisäänrakennettu verkkopalvelin, josta näkee helposti koko klusterin tilan. (Lam, 2011, s. 22)

3.3.2 DataNode

Klusterin slave-solmuissa on käynnissä DataNode-prosessi. Kun HDFS-tiedosto halutaan lukea tai kirjoittaa NameNode kertoo, missä klusterin DataNodeissa tiedoston lohkot sijaitsevat. DataNode kommunikoi klusterin muiden DataNodejen kanssa varmuuskopioidakseen omat datalohkonsa (Kuva 9). DataNodet antavat NameNodelle jatkuvasti tietoa ja pitävät sen ajan tasalla metatiedoston muutoksista. (Lam, 2011, s. 22)

Työn alussa DataNode kertoo, mitä tiedoston lohkoja se säilyttää omalla paikallisella levyllään. Työn aikana DataNodet tarjoavat tietoa työn edistymisestä sekä saavat ohjeita, jos niiden tulee luoda, siirtää tai poistaa tiedoston osia solmun omalta paikalliselta tallennuslevyltä. (Lam, 2011, s. 22)



Kuva 9. NameNode jakaa Testi.txt-tiedoston kuuteen datalohkoon ja siirtää ne DataNodeleille. DataNodet lähettävät varmuuskopioita toisille DataNodeille (Lam, 2011, s. 22)

3.3.3 Secondary NameNode (SNN)

Secondary NameNode (SNN) on apuprosessi NameNodelle. Toisin kuin NameNode, SNN ei käsittele reaaliaikaista tietoa metadatan muutoksista, vaan se valvoo klusterin tilaa ja kopioi tietyin väliajoin HDFS:n metadatan itselleen. Kopioiden tiheys pystytään määrittelemään klusterin konfiguroinneissa. NameNoden tavoin Secondary NameNoden kannattaa toimia omalla laitteellaan ilman DataNode- tai TaskTracker prosessien häiriöitä. (Lam, 2011, s. 23)

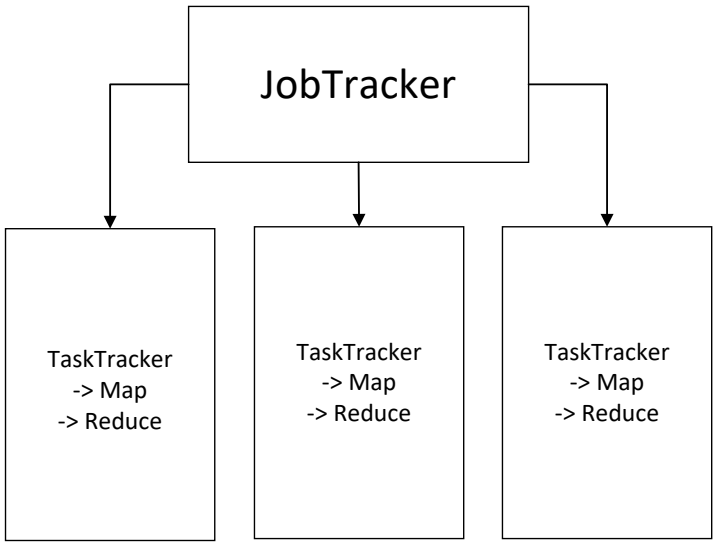
NameNoden vikatilanteessa SNN:n avulla minimoidaan järjestelmän alhaalla oloaika sekä pystytään vähentämään datan häviämistä. SNN voidaan konfiguroida toimimaan pääasiallisena NameNode-prosessina. Koska SNN on ottanut kopioita NameNodesta, se tietää missä datalohkot sijaitsevat. (Lam, 2011, s. 23)

3.3.4 JobTracker

JobTracker toimii luodun MapReduce-työn toimeenpanijana. JobTracker päättää toteutussuunnitelman, kun koodi on lähetetty klusteriin. Se määrittää, mitä tiedostoja käsitellään, määrää solmuja tietyille tehtäville ja valvoo jokaisen tehtävän suorittamista. Mikäli jokin tehtävä epäonnistuu, JobTracker aloittaa sen automaattisesti uudelleen jossakin toisessa solmussa. Hadoop-klusterissa toimii vain yksi JobTracker-prosessi ja se pyörii palvelimella, joka toimii klusterin master-solmuna. (Lam, 2011, s. 24)

3.3.5 TaskTracker

Edellä mainittu JobTracker on MapRecude-ohjelmien päätoimeenpanija. TaskTracker hoitaa yksittäisten tehtävien toimeenpanon jokaisessa slave-solmussa (Kuva 10). TaskTrackerin tulee jatkuvasti olla yhteydessä JobTrackeriin. Jos TaskTracker epäonnistuu informaation välittämisessä, JobTracker olettaa sen kaatuneen ja antaa sille määrätyn tehtävän jollekin toiselle solmulle. (Lam, 2011, s. 24)



Kuva 10. JobTracker jakaa sille annetun MapReduce-työn solmuille, jossa TaskTracker pitää sen jälkeen huolen tehtävien toimeenpanosta (Lam, 2011, s. 24)

3.4 Hadoopin konfigurointi

Suurin osa Hadoopin asetuksista sijaitsee .xml-tiedostoissa. Hadoopin ensimmäisissä versioissa kaikki konfiguroinnit tehtiin hadoop-site.xml-tiedostoon. Versiosta 0.20 ja sen jälkeen tämä tiedosto on jaettu neljäksi itsenäiseksi konfigurointitiedostoksi. Tiedostojen nimet ovat core-site.xml, hdfs-site.xml, yarn-site.xml sekä mapred-site.xml. (Lam, 2011, s. 28)

Oletuksena kaikki neljä tiedostoa ovat tyhjiä. Tällöin Hadoop toimii vain yhdellä paikallisella koneella eikä se käytä HDFS:ää tai käynnistä prosesseja. Tyhjän konfiguroinnin tarkoituksena on luoda ympäristö, jossa voidaan helposti kehittää ja testata MapReduce-ohjelmia. (Lam, 2011, s. 28)

Hadoop-env.sh-tiedosto sisältää muuttujia, joita käytetään Hadoop-skripteissä (Big Data University, 2016).

3.4.1 Core-site.xml

Core-site.xml-tiedosto sisältää informaatiota, joka ylikirjoittaa Hadoopin oletusasetukset (Big Data University, 2016).

Alla olevassa konfiguroinnissa annetaan NameNode-prosessille nimi ja sijainti HDFS varten:

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

3.4.2 Hdfs-site.xml

Hdfs-site.xml-tiedostossa konfiguroidaan asetukset HDFS-prosesseille. Prosessit ovat NameNode, DataNode sekä Secondary NameNode (SNN). (Big Data University, 2016)

Alla olevassa esimerkissä määritetään varmuuskopioiden määrä ja annetaan NameNode- sekä DataNode-prosessille hakemistopolku:

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

3.4.3 Mapred-site.xml

Mapred-site.xml-tiedostoon konfiguroidaan MapReduce-prosessien, JobTracker- ja TaskTracker-asetukset (Big Data University, 2016).

Alla olevassa esimerkissä konfiguroidaan portti, jossa MapReduce JobTracker-prosessi toimii sekä annetaan JobHistory-prosessille osoite:

```
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>localhost:9001</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.address</name>
    <value>localhost:10020</value>
  </property>
</configuration>
```

3.4.4 Yarn-site.xml

Yarn-site.xml-tiedostoon konfiguroidaan Resource Manager- sekä NodeManager-prosessien tiedot (Big Data University, 2016).

Esimerkki konfiguraatiossa määritetään MapReduce-ohjelmia varten shuffle-asetus sekä sen osoite:

```
<configuration>
  <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>
</configuration>
```

3.5 SSH (Secure Shell)

SSH-protokollalla luodaan suojattu yhteys koneiden välillä Hadoop-klusterissa. SSH antaa mahdollisuuden hallita järjestelmiä etäyhteydellä. Tällöin on mahdollista syöttää komentoja ja siirtää tiedostoja toiselta koneelta toiselle. SSH-protokollan on kehittänyt Tatu Ylönen vuonna 1995. (Rouse, Secure Shell (SSH), 2016)

3.6 Hadoop ja SQL

SQL (structured query language) on tarkoitettu strukturoidun datan käsittelyyn. Hadoopiin luodaan töitä, joiden avulla käsitellään strukturoimatonta dataa. Big Data onkin yleensä strukturoimatonta dataa, esimerkiksi tekstiä. (Lam, 2011, s. 7)

SQL-kielen käyttö Hadoopin kanssa ei onnistu suoraan. Tähän tarkoitukseen on syntynyt avoimen lähdekoodin projekteja, jotka mahdollistavat SQL-kielen käytön Hadoopiin tallennetun datan kanssa. Yksi sivuprojekti on Apache Hive, joka on Hadoopin päälle rakennettu datan käsittelyohjelma. Se toimii SQL-kielen tapaisilla komennoilla. Tätä kieltä kutsutaan HiveQL-kieleksi. HiveQL antaa paremmat mahdollisuudet käyttää strukturoitua dataa kuin MapReduce. Hive on vapaasti ladattavissa verkosta ja sitä konfiguroidaan hive-site.xml-tiedostossa. (Lam, 2011, s. 247)

4 VIRTUALISOINTI

Fyysinen laite vaatii huoltoa, tilaa, viilennystä ja sähköä. Tästä syystä on huomattavasti kustannustehokkaampaa käyttää virtuaalijärjestelmiä fyysisten laitteiden sijaan. Virtualisointi tarkoittaakin fyysisen laitteen resurssien jakamista. Virtualisoinnin muotoja ovat palvelimen, muistin, tietoverkon, työpöydän sekä sovelluksen virtualisointi. (De Tender, 2015, s. 4)

Virtualisoitujen järjestelmien tehokkuus ei yleensä vastaa fyysisen laitteen tehokkuutta. Virtualisoidun järjestelmän periaate on siinä, ettei sen tarvitse käyttää kaikkia tehoja fyysisestä laitteesta. Näin saadaan luotua useita joustavia, helposti hallittavia sekä eristettyjä järjestelmiä yhteen fyysiseen laitteeseen. (Rouse, Virtualization, 2016)

Virtualisointi on oleellinen termi Hadoopin yhteydessä. Kuten aiemmin on todettu, ei ole järkevää luoda klusteria monesta fyysisestä laitteesta, vaan on huomattavasti tilaa ja kustannuksia säästävämpää käyttää virtuaalikoneita.

4.1 Palvelimen, työaseman ja muistin virtualisointi

Palvelimen virtualisoinnilla tarkoitetaan fyysistä palvelinta, jonka resurssit, kuten muisti, levytila ja prosessorit, jaetaan moneksi palvelimeksi. Esimerkiksi fyysinen palvelin, jolla on 64 GB RAM, 2 TB levytilaa ja 4 prosessoria voidaan jakaa kahdeksi palvelimeksi. Näille palvelimille resurssit jaetaan joko suoraan puoliksi tai tarpeen mukaan niitä voidaan jakaa myös epätasaisesti. (De Tender, 2015, s. 5)

Työaseman virtualisointi toimii samalla periaatteella kuin palvelimen virtualisointi. Työaseman virtualisoinnissa käytetään kuitenkin työaseman käyttöjärjestelmää palvelimen käyttöjärjestelmän sijasta. (De Tender, 2015, s. 5)

Muistin virtualisointi tarkoittaa fyysisten kovalevyjen yhdistämistä. Ne on jaettu ja varattu palvelimille tai sovelluksille. Hadoop Distributed File System (HDFS) toimii tällä periaatteella. (De Tender, 2015, s. 5)

4.2 Tietoverkon ja sovelluksen virtualisointi

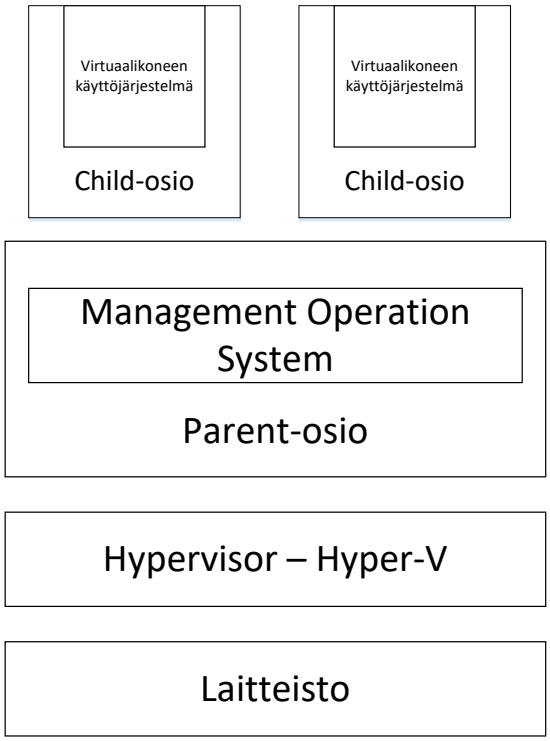
Yhdessä fyysisessä tietoverkossa toimii monta virtualisoitua tietoverkkoa. Jokainen virtualisoitu tietoverkko toimii niin, kuin se olisi ainoa käytössä oleva verkko. (De Tender, 2015, s. 6)

Sovelluksen virtualisoimisella tarkoitetaan sovelluksen ajamista eristetyllä virtualisoidulla tavalla. Virtualisoinnin hyötyjä ovat helpompi päivittäminen sekä yhteensopivuusongelmien minimoiminen. Virtualisoitua sovellusta ei ole suoraan asennettu käyttöjärjestelmään. (De Tender, 2015, s. 6)

5 MICROSOFT HYPER-V

Hypervisor tarkoittaa ohjelmaa tai laitetta, joka luo ja pitää yllä virtuaalikoneita. Hyper-V on tyypin 1 hypervisor. Se ei siis ole ohjelma tai käyttöjärjestelmän komponentti, vaan sillä on suora hallinta laitteistoon, johon se on asennettu. Hypervisor hallinnoi monia Hyper-V:een sisältämiä osioita. Management Operation System on virtuaalijärjestelmä, joka toimii Parent-osiossa (Kuva 11). (Siron & Syrewicze, 2014, s. 35)

Parent-osio on ainoa, joka kommunikoi suoraan hypervisorin kanssa. Se luo ja ylläpitää Child-osioita sekä tarjoaa virtuaalikoneille tarvittavat ajurit käytössä olevasta fyysisestä laitteesta. Child-osio säilöö virtuaalikoneen käyttöjärjestelmän, datan sekä sovellukset (Kuva 11). Vaikka Hyper-V on Microsoftin tuote, on sen avulla mahdollista luoda virtuaalikone mille tahansa tietokoneen käyttöjärjestelmälle. (Siron & Syrewicze, 2014, s. 35)



Kuva 7. Hyper-V -arkkitehtuuri (Siron & Syrewicze, 2014, s. 35)

5.1 Ubuntu

Tässä opinnäytetyössä Hadoopia tutkitaan Ubuntu-käyttöjärjestelmässä. Ubuntu eroaa yleisimmistä käyttöjärjestelmistä, kuten Microsoft Windows tai Mac OS X sillä, että sitä voi käyttää, muokata ja kehittää ilman rajoituksia. Tämän tekee mahdolliseksi Ubuntun käyttämä GNU General Public License (GPL). GPL-lisenssin on kehittänyt vuonna 1989 Richard Stallman ja sen tarkoituksena on antaa käyttäjille vapaa mahdollisuus ajaa, kopioida, jakaa, tutkia, muokata, kehittää ja parantaa ohjelmia. (Esengulov, 2012)

Ubuntu on Linuxin jakelu, eli se käyttää käyttöjärjestelmässään Linuxin kerneliä. Linux kernel toimii käyttöjärjestelmän ytimenä ja se auttaa ohjelmien kommunikointia laitteiston kanssa. Linuxin on kehittänyt Linus Torvalds vuonna 1991. Ubuntun tavoin myös Linux käyttää GPL-lisenssiä. (Esengulov, 2012)

5.2 Hyper-V, Virtualbox & VMware

Hadoop-asennusta varten voi käyttää haluamaansa virtuaalikoneiden luontiin tarkoitettua ohjelmaa. Tärkeintä on löytää ohjelma, joka vastaa omia tarpeita. Esimerkiksi Virtualbox on ilmainen ohjelma ja se sopii hyvin peruskäyttäjälle. Se kuitenkin vie käytössä olevan työaseman suorituskykyä. Tästä syystä isomman klusterin tekoon paras vaihtoehto on Hyper-V-palvelin. Kolmas yleisesti käytetty ohjelma on VMware, josta on saatavilla sekä maksullinen että maksuton versio.

6 HADOOPIN ASENNUSOHJE JA TEMPLATE

Tässä opinnäytetyössä oli tarkoituksena luoda kattavat ohjeet Hadoopin asennukseen niin koulutus- kuin hankekäyttöä varten (Liite 1). Hadoopin asennusta varten luotiin virtuaalikone ja siihen asennettiin käyttöjärjestelmäksi Ubuntun 16.04.1-työpöytäversio. Graafinen käyttöliittymä antaa ensimmäistä Hadoop-asennusta tekevälle paremman käsityksen asennuksen vaiheista. Kokeneempi Ubuntun käyttäjä osaa ohjeiden avulla luoda Hadoop-ympäristön myös palvelinversiolle, koska kaikki asennuksen vaiheet on mahdollista toteuttaa Terminaalilla eivätkä välttämättä vaadi graafista käyttöliittymää.

Asennusohjeessa on aluksi esitelty Ubuntun peruskomentoja, joita Hadoopin asennuksessa tarvitaan, kuten esimerkiksi sudo, mkdir ja cd. Ensimmäisessä vaiheessa on neuvottu Ubuntun päivittäminen ja opastettu Javan asentaminen vaihe vaiheelta.

Seuraavaksi tarkastellaan asennuksen valmistelu. Luodaan Hadoopia varten oma käyttäjä, hduser, sille käyttäjäryhmä hadoop ja annetaan hduser-käyttäjälle admin-oikeudet. Useamman koneen klusteria varten asennetaan SSH, jonka avulla saadaan suojattu yhteys klusterin koneiden välille. Viimeiseksi poistetaan IPv6 käytöstä.

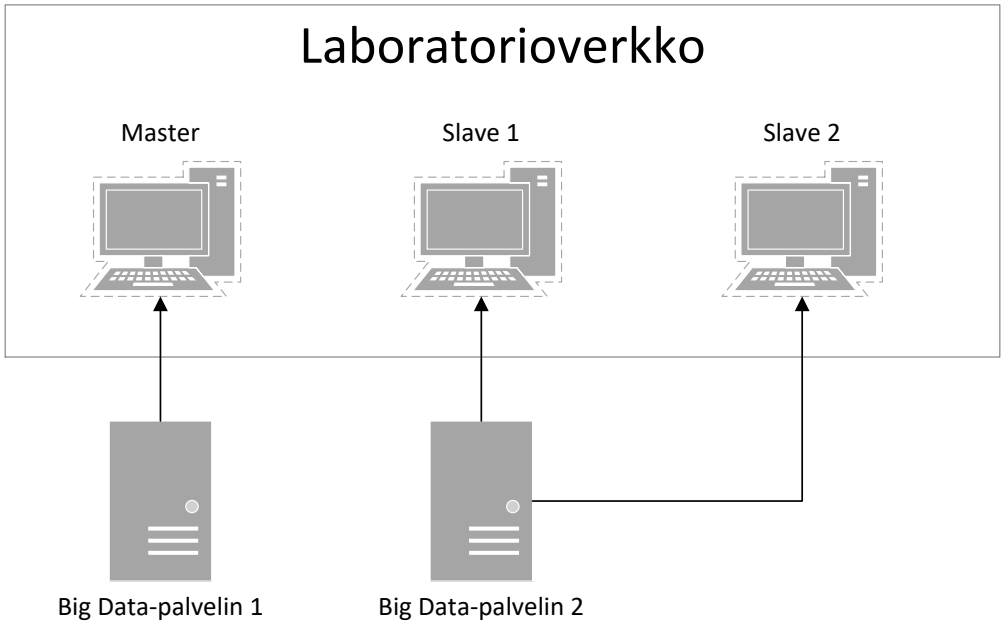
Kun asennuksen valmistelu on tehty, ohjeessa käydään tarkasti läpi yhden solmun klusterin asennus. Käydään läpi Hadoop-asennuspaketin lataus, sille hakemiston luominen sekä käyttäjäoikeuksien antaminen. Ohjeissa annetaan valmiit konfiguroinnit hadoop-env.sh-, .bashrc-, yarn-site.xml-, core-site.xml-, hdfs-site.xml- sekä mapred-site.xml-tiedostojen konfigurointiin. Tiedostojen konfiguroinnin jälkeen on asennus suoritettu sekä prosessit valmiit käynnistettäväksi.

Ohjeet jatkuvat siten, että muokataan yhden solmun klusterista monen solmun klusteri. Ohjeissa luodaan yksi master- sekä kaksi slave-konetta. Koneille tehdään tarvittavat konfiguraatiomuutokset, luodaan SSH-yhteydet sekä annetaan koneille oikeat master/slave-suhteet. Ohjeiden lopussa käydään vielä läpi, mitä tulee muistaa, jos haluaa lisätä uusia solmuja klusteriin sekä käydään läpi vianetsintää, jos asennus tai konfiguroinnit eivät jostain syystä onnistuneet.

Koulutus- sekä hankekäyttöä varten tavoitteena oli luoda templeitit valmiista Hadoop-asennuksista. Koulun uudelle Big Data -palvelimelle luotiin kolme virtuaalikonetta. Yhteen virtuaalikoneeseen asennettiin yhden solmun klusterin Hadoop-asennus. Kahdesta muusta virtuaalikoneesta luotiin usean solmun klusteri. Toinen kone toimi master-koneena ja toinen slave-koneena. Näiden koneiden rinnalle on tarvittaessa helppoa luoda suurempi klusteri asennusohjeita käyttäen.

7 BIG DATA -PALVELIMET

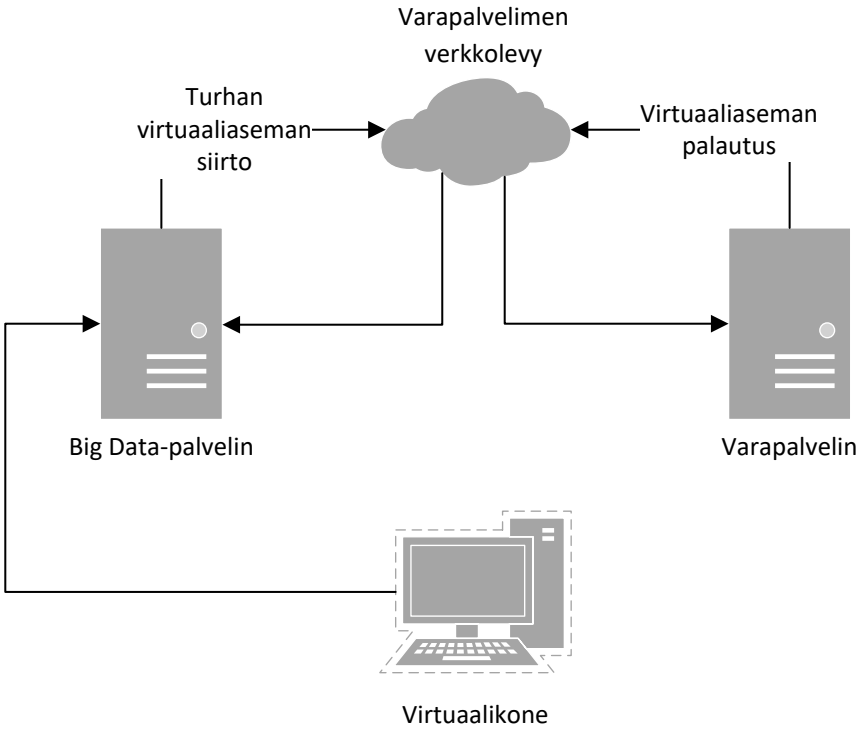
Big Datan käyttöä varten on Savonia-ammattikorkeakoululle hankittu kaksi identtistä Big Data -palvelinta. Molemmille palvelimille on mahdollista luoda virtuaalikoneita. Hadoop-klusteri on mahdollista luoda niin, että virtuaalikoneita luodaan molemmille palvelimille. Virtuaalikoneita luodessa niiden verkoksi valitaan koulun laboratorioverkko. Tästä verkosta koneet saavat IP-osoitteet samalta alueelta ja toimivat näin ollen klusterissa (Kuva 12).



Kuva 8. Kahdelle palvelimelle luotu klusteri. Toiselle palvelimelle on luotu master-solmun kone ja toiselle kaksi slave-konetta

Kahden palvelimen avulla on mahdollista jakaa koulutuksen sekä hankkeiden aiheuttamaa kuormaa. Esimerkiksi toinen palvelin voi olla pelkästään koulutus- ja toinen hankekäyttössä. Toinen vaihtoehto on luoda esimerkiksi master-solmu sekä koulutuskäyttö samalle palvelimelle ja hankekäytön slave-solmut toiselle. IP-alueiden takia (Kuva 12) palvelinten kuormitusta on helppo säädellä.

Big Data -palvelimille on varattu myös varapalvelin. Sille on tarkoitus varmuuskopioida ja siirtää turhia sekä palauttaa haluttuja virtuaalikoneita tarpeen mukaan (Kuva 13). Näin ne eivät vie tilaa Big Data -palvelimella. Tiedostojen siirron voi tehdä verkkolevyn avulla, joka luodaan varapalvelimelta. Kun verkkolevy on valmis, lisätään se Big Data -palvelimen resurssienhallintaan ja sieltä tiedostoja voi siirtää varapalvelimelle (Kuva 13). Tiedostojen siirtoaika tulee olemaan pitkä, koska virtuaalikoneet vievät paljon levytilaa.



Kuva 9. Varapalvelimen verkkolevyn hyödyntämisen arkkitehtuurikuvaus

Edellä esitellyllä tavalla on mahdollista luoda erilaisia Hadoop-klustereita, jotka sopivat eri tarkoituksiin. Pitämällä niitä varapalvelimella, ne eivät vie turhaa levytilaa Big Data -palvelimelta. Lisäksi be ovat helposti käytettävissä tarpeen tullen.

8 YHTEENVETO JA POHDINTA

Opinnäytetyön aihe oli erittäin mielenkiintoinen, koska termi Big Data -termi oli melko vieras. Hadoop-termi tuli ensimmäistä kertaa esille, kun Keijo Kuosmanen tarjosi opinnäytetyön aihetta. Työ alkoi opettelemalla Big Datan sekä Hadoopin teoriaa perusteista lähtien. Aiheesta löytyi hyvin paljon kirjallisuutta. Keijo Kuosmanen suositteli opinnäytetyön alkupalaverissa tutustumaan Big Data University-sivustoon. Sivusto tarjosi erittäin monipuolisia Big Dataa käsitteleviä kursseja, mikä helpotti oppimisprosessia sekä opinnäytetyön tekemistä.

Hadoop-ympäristön luominen koulun palvelimille sekä asennusohjeiden laatiminen onnistuivat erittäin hyvin. Ohjeet testattiin ja niiden avulla Savoniassa alkavan Big Data -kurssin opiskelijat tulevat onnistuvat Hadoopin asennuksessa. Työtä tehdessä Ubuntun käyttöjärjestelmä tuli hyvin tutuksi. Opinnäytetyötä tehdessä ei ilmennyt ylitsepääsemättömiä ongelmia. Pienet haasteet tai ongelmakohdat olivat helposti ratkaistavissa.

Tämä opinnäytetyö opetti itsenäisen työn tekemiseen sekä antoi hyvän pohjan tulevaan työelämään. Jatkokehitys työlle olisi seuraavaksi MapReduce-töiden opiskelu sekä ohjeistuksen tekeminen.

LÄHTEET JA TUOTETUT AINEISTOT

Beal, V. (ei pvm). *Structured data*. (Webopedia) Haettu 27. Joulukuu 2016 osoitteesta http://www.webopedia.com/TERM/S/structured_data.html

Big Data University. (17. Maaliskuu 2016). *Hadoop 101*. Haettu 27. Joulukuu 2016 osoitteesta https://courses.bigdatauniversity.com/courses/course-v1:BigDataUniversity+BD0111EN+2016/courseware/abee0aa450b24b97bbf39a3272a06891/9971509d64984f8ebd7aeeea770aa394/

De Tender, P. (2015). *Mastering Hyper-V.* Birmingham: Packt Publishing Ltd.

DeVan, A. (7. Huhtikuu 2016). *The 7 V's of Big Data*. (Impact Radius) Haettu 14. Joulukuu 2016 osoitteesta https://www.impactradius.com/blog/7-vs-big-data/

Esengulov, A. (20. Tammikuu 2012). *Ubuntu: A Beginner's Guide*. (MakeUseOf) Haettu 4. Tammikuu 2017 osoitteesta http://www.makeuseof.com/tag/ubuntu-an-absolute-beginners-guide/

Hansen, D. (2013). *Oracle Fast Data: Real-Time Strategies for Big Data and Business Analytics.* Redwood Shores: Oracle Corporation.

Kalyvas, J. R.;& Overly, M. R. (2015). *Big Data, A Business and Legal Guide.* Boca Raton: Taylor & Francis Group, LLC.

Lam, C. (2011). *Hadoop in Action.* Stamford: Manning Publications Co.

Rouse, M. (23. Maaliskuu 2016). *Secure Shell (SSH)*. (TechTarget) Haettu 20. Tammikuu 2017 osoitteesta http://searchsecurity.techtarget.com/definition/Secure-Shell

Rouse, M. (Lokakuu 2016). *Virtualization*. (TechTarget) Haettu 1. Helmikuu 2017 osoitteesta http://searchservervirtualization.techtarget.com/definition/virtualization

Salo, I. (2014). *Big Data & Pilvipalvelut.* Jyväskylä: Docendo Oy.

Siron, E.;& Syrewicze, A. (2014). *Hyper-V Security.* Birmingham: Packt Publishing Ltd.

VCloudNews. (5. Huhtikuu 2016). *EVERY DAY BIG DATA STATISTICS – 2.5 QUINTILLION BYTES OF DATA CREATED DAILY.* (VCloudNews) Haettu 5. Helmikuu 2017 osoitteesta http://www.vcloudnews.com/every-day-big-data-statistics-2-5-quintillion-bytes-of-data-created-daily/

LIITE 1: HADOOPIN ASENNUSOHJE

Tässä asennusohjeessa käydään läpi Hadoopin asennuksen eri vaiheet Ubuntun työpöytä-käyttöjärjestelmässä.

KÄYTTÄJÄN TIEDOT

Valmista Hadoop-templeittiä käyttäessä Ubuntu-tunnukset ovat seuraavat:

Käyttäjänimi: student & hduser
Salasana: hadoop

Student-käyttäjä on luotu käyttöjärjestelmän luonnin yhteydessä. Hduser-käyttäjää tulee käyttää Hadoopin yhteydessä.

KOMENNOT

sudo = suoritetaan komentoja root-käyttäjän oikeuksilla
apt-get = hakee asennuspaketit verkosta
mkdir = luo hakemiston
install = asentaa valitun tiedoston
chmod = muutta tiedoston oikeuksia
chown = muuttaa hakemiston tai tiedoston omistajaa
cp = kopioi tiedoston tai hakemiston
cd = siirtyy hakemistosta toiseen
tar = purkaa pakattuja tiedostoja
rm = poistaa tiedoston tai hakemiston
cat = tulostaa valitun tiedoston
mv = siirtää tiedoston tai hakemiston
gedit = tekstieditori
adduser = lisää käyttäjän
addgroup = lisää käyttäjäryhmän

1 UBUNTUN PÄIVITTÄMINEN, VIM-EDITORITYÖKALU JA JAVAN ASENNUS

1.1 UBUNTUN PÄIVITTÄMINEN

Avaa Terminaali:

Hae ja asenna viimeisimmät päivitykset komennoilla:

$ sudo apt-get update

$ sudo apt-get upgrade

Terminaali kysyy ”Do you want to continue? [Y/n]”. Kirjoita Y ja paina Enter.

1.2 VIM-EDITORITYÖKALU

Vim-editorilla voi muokata konfigurointi-tiedostoja. Sen saa asennettua komennolla:

$ sudo apt-get install vim

Vastaa Y-kirjain ja Enter. Vim toimii gedit-komennon tavoin. Kun tiedosto on avattu Vimillä, sitä pääsee muokkaamaan painamalla i-näppäintä. Tietojen muokkauksen voi lopettaa painalla Esc-näppäintä ja sen jälkeen kirjoittamalla ":wq" (w = write ja q = quit) ja Enter.

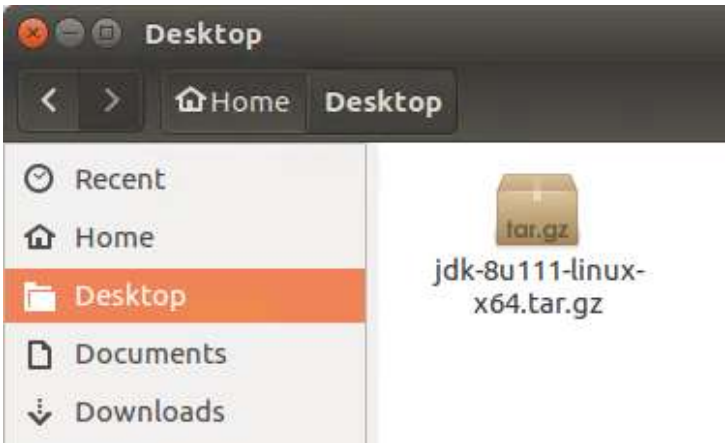
1.3 JAVAN ASENNUS

Avaa FireFox Web Browser ja kopio URL:

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

Klikkaa "I Agree" ja tämän jälkeen hyväksy Javan lisenssi klikkaamalla "Accept License Agreement". Etsi listasta kohta "Linux x64" ja klikkaa .tar.gz päätteistä lataustiedostoa. Valitse "Save File". Latauksen päätyttyä etsi tiedosto klikkaamalla file-ikonia:

Ja sen jälkeen tuplaklikkaamalla Downloads-kansiota. Downloads-kansiossa klikkaa ladattua tiedostoa hiiren oikealla näppäimellä, valitse copy tai cut ja siirrä tiedosto työpöydälle.



Seuraavaksi siirretään tiedosto kansiopolkuun /usr/local/java/. Ensimmäisenä siirrytään terminaalissa työpöydälle:

Seuraavaksi luodaan kansiopolku komennolla:

$ sudo mkdir /usr/local/java

Annetaan luodulle kansiolle käyttäjäoikeudet komennolla:

$ sudo chmod 777 /usr/local/java

Luku 777 antaa jokaiselle käyttäjälle täydet käyttöoikeudet.

Seuraavaksi kopioidaan työpöydällä oleva tiedosto uuteen polkuun:

$ sudo cp ~/Desktop/jdk-8u???-linux-x64.tar.gz /usr/local/java

HUOM!!! Vaihda komennon kysymysmerkit lataamasi tiedoston versionumeroon, esimerkiksi 8u111.

Seuraavaksi siirry terminaalissa luotuun kansioon:

$ cd /usr/local/java

Voit varmistaa tiedoston siirtymisen ls-komennolla:

```
student@HadoopSRV:/usr/local/java$ ls
jdk-8u111-linux-x64.tar.gz
```

1.3.1 JAVAN .TAR.GZ-TIEDOSTON PURKAMINEN

Puretaan .tar.gz-tiedosto komenolla:

$ sudo tar xvzf jdk-8u???-linux-x64.tar.gz

Tiedoston purkamisen jälkeen ls-komennolla nähdään purettu tiedosto ja vanha pakattu tiedosto voidaan poistaa:

$ sudo rm jdk-8u???-linux-x64.tar.gz

Seuraavaksi muokataan /etc/profile/-tiedostoa. Avaa tiedosto komennolla:

$ sudo gedit /etc/profile/

Rullaa alas ja viimeisien ”fi”-kirjainten jälkeen kirjoita seuraavat komennot:

```
if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
  unset i
fi
```

```
JAVA_HOME=/usr/local/java/jdk1.8.0_111
JRE_HOME=$JAVA_HOME/jre
PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
export JAVA_HOME
export JRE_HOME
export PATH
```

Vaihda ensimmäisen rivin loppu lataamasi tiedoston versionumeroon. Tallenna ja sulje tiedosto.

1.3.2 UUDEN JAVAN SIJAINNIN MAINOSTAMINEN

Ubuntun asennuksen mukana tulee Java. Informoidaan Ubuntua, missä uusi Java sijaitsee komennoilla:

$ sudo update-alternatives –install ”/usr/bin/java” ”java” ”/usr/local/java/jdk1.8.0_???/jre/bin/java” 1

$ sudo update-alternatives –install ”/usr/bin/javac” ”javac” ”/usr/local/java/jdk1.8.0_???/bin/javac” 1

Tämän jälkeen pakotetaan Ubuntua käyttämään uutta Javaa oletuksena:

$ sudo update-alternatives –set java /usr/local/java/jdk1.8.0_???/jre/bin/java

$ sudo update-alternatives –set javac /usr/local/java/jdk1.8.0_???/bin/javac

Seuraavaksi päivitetään vielä profiili-tiedosto:

$. /etc/profile

Tarkista Java-versiosi:

```
student@HadoopSRV:/usr/local/java$ java -version
java version "1.8.0_111"
Java(TM) SE Runtime Environment (build 1.8.0_111-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.111-b14, mixed mode)
```

2 HADOOP-ASENNUKSEN VALMISTELU

Vaihdetaan terminaalissa oma käyttäjä root-käyttäjäksi:

$ sudo su

Terminaali pyytää student-käyttäjän salasanaa, jonka jälkeen käyttäjä muuttuu.

```
root@HadoopSRV:/home/student#
```

2.1. HADOOP-KÄYTTÄJÄN LUOMINEN

Hadoopia varten luodaan root-käyttäjänä uusi käyttäjä sekä käyttäjäryhmä.

Root-käyttäjänä sudo-komennon käyttö ei ole pakollista. Luodaan käyttäjäryhmä komennolla:

$ sudo addgroup hadoop

Ja Hadoop-käyttäjä:

$ sudo adduser hduser

Anna hduser-käyttäjälle salasanaksi hadoop, muut tiedot ovat valinnaisia. Lopuksi hyväksy uusi käyttäjä kirjoittamalla Y ja Enter. Tämän jälkeen lisätään luotu hduser-käyttäjä hadoop-käyttäjäryhmään.

$ sudo adduser hduser hadoop

Lisätään hduser-käyttäjä sudoers-listaan, jolloin käyttäjä saa admin oikeudet.

$ sudo visudo

Etsitään kohta ”# Allow members of group sudo to execute any command” ja lisätään oikeudet hduserille:

```
# Allow members of group sudo to execute any command
hduser ALL=(ALL) ALL
%sudo   ALL=(ALL:ALL) ALL
```

Poistu editorista painamalla Ctrl + x, sitten Y ja Enter. Tämän jälkeen kirjaudu sisään hduser-käyttäjänä:

$ sudo su hduser

2.2. SSH KONFIGUROINTI

SSH:n saa asennettua Ubuntulle komennolla:

$ sudo apt-get install openssh-server

Syötä salasana ja valitse Y. Seuraavaksi luodaan avaimia, joiden avulla Hadoop-prosessit pystyvät keskustelemaan keskenään:

$ ssh-keygen

Jatka painamalla Enteriä niin kauan, kunnes palaat komentoriville:

```
+---[RSA 2048]----+
|       o..       |
|      . o .      |
|       o ....  .  |
|        +.+o  o  |
|       . Eo. oo  |
|        *oo.. .= |
|       o..+.+oo . |
|       o=.oB*. o +|
|       ++oo=+=+.+.|
+----[SHA256]-----+
hduser@HadoopSRV:~$
```

Seuraavaksi kopioidaan saadut avaimet Authorized_key-tiedostoon ja muokataan siellä käyttöoikeuksia. Avaimet sijaitsevat ~/.ssh/id_rsa.pub ja ne siirretään ~/.ssh/authorized_keys-polkuun.

$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

Annetaan ainoastaan hduser-käyttäjälle muokkausoikeudet kyseiseen kansioon.

$ chmod 700 ~/.ssh/authorized_keys

Seuraavaksi käynnistetään SSH-serveri:

$ sudo /etc/init.d/ssh restart

Ja testataan SSH:n toimivuutta:

$ ssh localhost

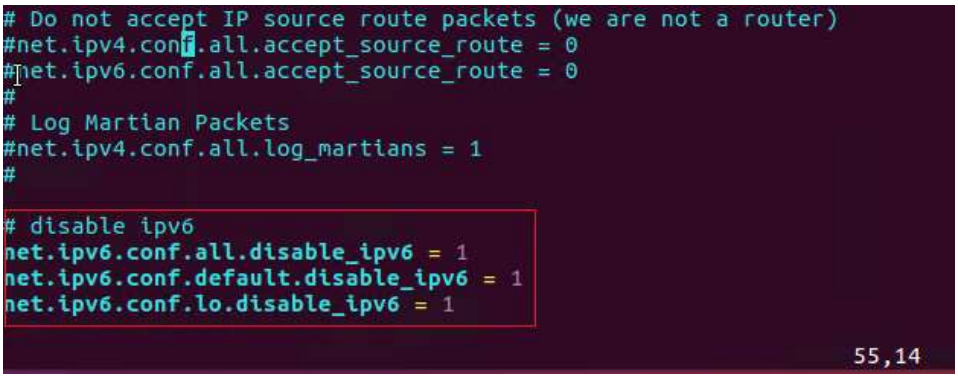
Vastaa ”yes” ja Enter.

2.3. IPV6 POIS KÄYTÖSTÄ

Hadoop ja IPv6 eivät toimi yhteen, joten poistetaan IPv6 käytöstä:

$ sudo vim /etc/sysctl.conf

Mene nuolinäppäimillä tiedoston loppuun, paina i-näppäintä ja kirjoita:



Tämän jälkeen paina Esc-näppäintä, kirjoita ":wq" ja Enter. Varmista, että IPv6 on pois päältä:

$ cat /proc/sys/net/ipv6/conf/all/disable_ipv6

Testin pitäisi antaa tulokseksi nolla. Tämän jälkeen käynnistä virtuaalikone uudelleen.

4 HADOOPIN ASENNUS (YHDEN SOLMUN KLUSTERI)

4.1. HADOOP-TIEDOSTON LATAUS JA PURKAMINEN

Avaa Mozilla Firefox Web Browser ja kopioi URL:
http://hadoop.apache.org/releases.html

Valitaan uusin versio ja klikataan binary-kohtaa. Tällöin avautuu uusi sivu, valitse ylin linkki, jonka pääte on .tar.gz.

Lataa tiedosto ja siirrä se työpöydälle. Työpöydältä tiedosto siirretään /usr/local/-polkuun:

$ cd Desktop

$ sudo mv ~/Desktop/hadoop-2.7.?.tar.gz /usr/local/

Muista käyttää lataamasi tiedoston versiota. Mene polkuun, johon tiedosto siirrettiin ja tarkista ls-komennolla, että tiedosto on siirtynyt:

$ cd /usr/local

Tämän jälkeen puretaan .tar.gz-tiedosto:

$ sudo tar -xvf hadoop-2.7.?.tar.gz

Kun toimenpide valmis poista pakattu tiedosto:

$ sudo rm hadoop-2.7.?.tar.gz

Seuraavaksi luodaan hadoop-2.7.?-kansiolle pikakuvake. Tämän avulla ei tarvitse jokaisella kerralla kirjoittaa versionumeroita konfigurointeja tehdessä:

$ sudo ln -s hadoop-2.7.? hadoop

4.2. HADOOP-KANSION OIKEUKSIEN LISÄÄMINEN

Seuraavaksi teemme hduser-käyttäjästä hadoop-kansion omistajan. Komennolla:

$ ls -ltr

näkee, että tällä hetkellä hadoop-kansion omistaa root-käyttäjä. Vaihdetaan kansion omistajaksi hduser, joka kuuluu hadoop-käyttäjäryhmään:

$ sudo chown -R hduser:hadoop hadoop-2.7.?

Jos katsotaan omistusoikeuksia uudelleen, nähdään, että hduser omistaa hadoop-2.7.?-kansion ja root-käyttäjä hadoop -> hadoop-2.7.3-kansion. Root-käyttäjä omistaa siis enää vain pikakuvakkeen. Vaihdetaan omistusoikeus hduser-käyttäjälle:

$ sudo chown -R hduser:hadoop hadoop

Tämän jälkeen annetaan täydet käyttöoikeudet kansioon:

$ sudo chmod 777 hadoop-2.7.?

4.3. HADOOP-ENV.SH-TIEDOSTON KONFIGUROINTI

Seuraavaksi konfiguroidaan hadoop-env.sh-tiedosto. Aukaistaan tiedosto:

$ sudo gedit /usr/local/hadoop/etc/hadoop/hadoop-env.sh

Mene tiedoston pohjalle ja kirjoita seuraavat komennot:

```
# A string representing this instance of hadoop. $USER by default.
export HADOOP_IDENT_STRING=$USER

export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true
export HADOOP_HOME_WARN_SUPPRESS="TRUE"
export JAVA_HOME=/usr/local/java/jdk1.8.0_111
```

Varmista, että käytät oman Java-version tietoja. Tallenna ja sulje tiedosto.

4.4. .BASHRC-TIEDOSTON KONFIGUROINTI SEKÄ DFS-HAKEMISTON LUONTI

Seuraavaksi muokataan .bashrc-tiedostoa. Avaa tiedosto komennolla:

$ gedit ~/.bashrc

Mene tiedoston pohjalle ja kopioi siihen alla oleva teksti:

 # Asetetaan Hadoop-muuttujat
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_PREFIX=/usr/local/hadoop
export HADOOP_MAPRED_HOME=${HADOOP_HOME}
export HADOOP_COMMON_HOME=${HADOOP_HOME}
export HADOOP_HDFS_HOME=${HADOOP_HOME}

export HADOOP_YARN_HOME=${HADOOP_HOME}
export HADOOP_CONF_DIR=${HADOOP_HOME}/etc/hadoop

 # Paikallinen reitti
export HADOOP_COMMON_LIB_NATIVE_DIR=${HADOOP_PREFIX}/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_PREFIX/lib"

 # Asetetaan JAVA_HOME
export JAVA_HOME=/usr/local/java/jdk1.8.0_???
#HUOM oman Javan versio!

#Lisätään Hadoop bin/ hakemisto PATH:lle
export PATH=$PATH:$HADOOP_HOME/bin:$PATH:$JAVA_HOME/bin:$HADOOP_HOME/sbin

Tarkista, että tiedot ovat oikein, tallenna ja sulje tiedosto. Tämän jälkeen sulje myös Terminaali ja avaa se uudelleen.

Seuraavaksi luodaan väliaikaishakemisto DFS:ää varten, annetaan hduser-käyttäjälle hakemiston omistajuus ja kaikille käyttöoikeudet kansioon:

$ sudo mkdir -p /app/hadoop/tmp
$ sudo chown -R hduser:hadoop /app/hadoop/tmp
$ sudo chmod -R 777 /app/hadoop/tmp

4.5. YARN-, CORE- SEKÄ MAPRED--SITE.XML KONFIGUROINTI

Yarn-site.xml-tiedoston konfigurointi:

$ gedit /usr/local/hadoop/etc/hadoop/yarn-site.xml

Tee tiedostossa seuraavat konfiguraatiot:

```
<configuration>
        <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>
</configuration>
```

Tämän jälkeen konfiguroidaan core-site.xml-tiedosto:

$ gedit /usr/local/hadoop/etc/hadoop/core-site.xml

Tee tiedostossa seuraavat konfiguraatiot:

```
<configuration>
        <property>
                <name>hadoop.tmp.dir</name>
                <value>/app/hadoop/tmp</value>
                <description>Hakemistojen perussijainti</description>
        </property>
        <property>
                <name>fs.default.name</name>
                <value>hdfs://localhost:9000</value>
                <description>Oletustietojärjestelman nimi</description>
        </property>
</configuration>
```

Seuraavaksi luodaan tiedostosta mapred-site.xml.template kopio tiedostoon mapred-site.xml:

$ cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml

Tämän jälkeen päästään konfiguroimaan mapred-site.xml-tiedosto:

$ gedit /usr/local/hadoop/etc/hadoop/mapred-site.xml

Tee tiedostossa seuraavat konfiguraatiot:

```
<configuration>
        <property>
                <name>mapreduce.framework.name</name>
                <value>yarn</value>
        </property>
        <property>
                <name>mapreduce.jobhistory.address</name>
                <value>localhost:10020</value>
        </property>
</configuration>
```

4.6. HDFS-HAKEMISTON LUONTI SEKÄ HDFS-SITE.XML-TIEDOSTON KONFIGUROINTI

Seuraavaksi luodaan hakemistoja, joihin Hadoop tallentaa työnsä. Annetaan hakemistoille omistaja ja oikeudet:

$ sudo mkdir -p /usr/local/hadoop/yarn_data/hdfs/namenode

$ sudo mkdir -p /usr/local/hadoop/yarn_data/hdfs/datanode

$ sudo chmod 777 /usr/local/hadoop/yarn_data/hdfs/namenode

$ sudo chmod 777 /usr/local/hadoop/yarn_data/hdfs/datanode

$ sudo chown -R hduser:hadoop /usr/local/hadoop/yarn_data/hdfs/namenode

$ sudo chown -R hduser:hadoop /usr/local/hadoop/yarn_data/hdfs/datanode

Tämän jälkeen konfiguroidaan hdfs-site.xml-tiedosto:

$ gedit /usr/local/hadoop/etc/hadoop/hdfs-site.xml

Tee tiedostossa seuraavat konfiguraatiot:

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

4.7. NAMENODEN FORMATOINTI JA HADOOPIN KÄYNNISTYS

Tämän jälkeen avaa uusi terminaali-ikkuna namenode-prosessin formatoimista varten:

$ hadoop namenode -format

Kun formatointi on suoritettu onnistuneesti, voidaan käynnistää juuri konfiguroitu yhden solmun Hadoop-klusteri:

- voit käynnistää prosessit erikseen
  - **$ start-dfs.sh**
  - **$ start-yarn.sh**
- tai kaikki kerralla
  - **$ start-all.sh**
- sammuta prosessit
  - **$ stop-all.sh**

”Do you want to continue connectiong?” kirjoita ”yes”, pelkkä y-kirjain ei riitä.

Kun prosessit on käynnistetty, varmista jps-komennolla niiden toimivuus:

```
hduser@HadoopSRV:~$ jps
3457 NameNode
3940 ResourceManager
3783 SecondaryNameNode
4056 NodeManager
4367 Jps
3599 DataNode
```

Käynnistä MapReduce-HistoryServer-prosessi komenolla:

$ mr-jobhistory-daemon.sh start historyserver

HistoryServer-prosessin avulla näet mitä MapReduce-töitä on tehty.

Seuraavaksi tarkistetaan, onko HDFS-hakemistoon luotu kotikansiota:

$ hadoop fs -ls

Jos terminaali ilmoittaa ”No such file or directory” luodaan se seuraavasti:

$ hdfs dfs -mkdir -p /user/hduser

Tämän jälkeen virheilmoitusta ei pitäisi enään tulla.

Selaimella pääset näkemään NameNoden, DataNoden ja ResourceManagerin tilan:

NameNode:	http://localhost:50070
ResourceManager:	http://localhost:8088
History Server:	http://localhost:19888

5 HADOOPIN ASENNUS (MONEN SOLMUN KLUSTERI)

Nämä ohjeet on tehty yhdelle master-koneelle ja kahdelle slave-koneelle. Samoilla ohjeilla on mahdollista kuitenkin tehdä myös suurempi klusteri, koska periaate pysyy samana.

Tämä ohje on jatkoa yhden solmun klusterin-ohjeesta. Aikaisemmin asennettu kone on kloonattu kolmeksi koneeksi.
Kun kolme virtuaalikonetta on valmiina, tarkistetaan, että ne löytävät toisensa. Ensimmäiseksi katsotaan jokaisesta koneesta IP-osoite komennolla:

$ ifconfig

Tämän jälkeen ping + ip-osoite-komennolla varmistetaan, että saat yhteyden jokaiseen koneeseen. Terminaali lopettaa ”pingaamisen” Ctrl + C-näppäinyhdistelmällä.

5.1. HOSTNAMEN VAIHTO SEKÄ HOSTS-YHTEYKSIEN LUONTI

Tämän jälkeen vaihdetaan jokaisen koneen nimi (hostname):

$ sudo gedit /etc/hostname

Poista edellinen nimi ja vaihda se koneen mukaan joko master, slave1 tai slave2. Tämän jälkeen päivitetään jokaisen koneen hosts-tiedosto uusilla nimillä sekä IP-osoitteilla:

$ sudo gedit /etc/hosts

Poista tiedostosta vanha koneen nimi ja IP-osoite:

```
127.0.0.1       localhost
127.0.1.1       HadoopSRV
```

Lopputuloksen pitäisi alla olevan mallin mukainen. X-kirjainten paikalla on jokaisen koneen oma IP-osoite:

```
127.0.0.1       localhost

xxx.xxx.xxx.xxx master
xxx.xxx.xxx.xxx slave1
xxx.xxx.xxx.xxx slave2

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Tallenna ja sulje tiedostot. Tämän jälkeen käynnistä jokainen virtuaalikone uudelleen, jotta muutokset tulevat voimaan. Uudelleen käynnistyksen jälkeen varmista koneen hostname Terminaalista. Seuraavaksi kokeillaan ping-komentoa hostnamen avulla jokaisesta koneesta:

$ ping master

$ ping slave1

$ ping slave2

5.2. SSH-YHTEYKSIEN KONFIGUROINTI

Seuraavaksi konfiguroidaan SSH-yhteys niin, että ottaessa yhteyttä yhdeltä koneelta toiselle, ei tarvitse käyttää salasanoja. Konfiguroidaan master-koneelta:

$ ssh master #vastaa yes

$ exit

$ ssh slave1 #vastaa yes

$ exit

$ ssh slave2 #vastaa yes

$ exit

Toista sama kaikilla koneilla. Aloita komennot, sillä koneella millä olet!

5.3. KONFIGURAATIO-TIEDOSTOJEN MUUTOKSET

Kun SSH-yhteydet ovat kunnossa aloitetaan konfiguraatioiden teko. Ensimmäisenä päivitetään core-site.xml-tiedosto jokaisessa koneessa:

$ sudo gedit /usr/local/hadoop/etc/hadoop/core-site.xml

Tiedostossa poista seuraavaksi ”hadoop.tmp.dir”-property ja muuta ”fs.default.name”-propertyn localhost masteriksi:

```
<configuration>
        <property>
                <name>fs.default.name</name>
                <value>hdfs://master:9000</value>
                <description>Oletustietojärjestelman nimi</description>
        </property>
</configuration>
```

Seuraavaksi päivitetään hdfs-site.xml-tiedosto jokaisessa koneessa:

$ sudo gedit /usr/local/hadoop/etc/hadoop/hdfs-site.xml

Tiedoston konfiguroinnit eroavat onko kyseessä master- vai slave-kone:

```
<configuration>
        <property>
                <name>dfs.replication</name>
                <value>2</value> Vaihda replikointi 1 -> 2
        </property>
        <property>
                <name>dfs.namenode.name.dir</name> Pidä master-koneella, poista slave-koneista
                <value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
        </property>
        <property>
                <name>dfs.datanode.name.dir</name> Pidä slave-koneilla, poista master-koneesta
                <value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
        </property>
</configuration>
```

Muista myös muuttaa tiedostojen polut. Koska NameNode-prosessi pyörii vain master-solmussa, ei sitä tarvita slave-koneella. Toisaalta DataNode-prosessit pyörivät vain slave-solmuissa, joten se otetaan pois master-koneelta.

Tämän jälkeen konfiguroidaan yarn-site.xml-tiedosto, joka tukee klusteria. Konfiguroinnit tehdään jokaisessa koneessa:

$ sudo gedit /usr/local/hadoop/etc/hadoop/yarn-site.xml

Lisää konfiguraatiot edellisten perään:

```
<configuration>
        <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>
        <property>
                <name>yarn.resourcemanager.resource-tracker.address</name>
                <value>master:8025</value>
        </property>
        <property>
                <name>yarn.resourcemanager.scheduler.address</name>
                <value>master:8030</value>
        </property>
        <property>
                <name>yarn.resourcemanager.address</name>
                <value>master:8050</value>
        </property>
</configuration>
```

Tämän jälkeen päivitetään mapred-site.xml-tiedosto kaikissa koneissa:

$ sudo gedit /usr/local/hadoop/etc/hadoop/mapred-site.xml

Vaihda HistoryServer-prosessin osoitteeksi master:

```
<configuration>
        <property>
                <name>mapreduce.framework.name</name>
                <value>yarn</value>
        </property>
        <property>
                <name>mapreduce.jobhistory.address</name>
                <value>master:10020</value>
        </property>
</configuration>
```

5.4. ANNETAAN SOLMUILLE MASTER- JA SLAVE-ROOLIT

Seuraava konfiguraatio tehdään ainoastaan master-koneella. Päivitetään masters- ja slaves-tiedostoihin oikeat solmut:

$ sudo gedit /usr/local/hadoop/etc/hadoop/slaves

Mikäli tiedostossa lukee ”localhost”, poista se ja kirjoita tilalle allekkain slave1 ja slave2, jonka jälkeen tallenna ja sulje tiedosto. Seuraavaksi master-tiedoston päivitys:

$ sudo gedit /usr/local/hadoop/etc/hadoop/masters

Master-tiedostoon kirjoita master, tallenna ja sulje. Nämä tiedostot auttavat suorittamaan oikeat prosessit oikeassa solmussa.

5.5. LUODAAN UUSI NAMENODE-HAKEMISTO

Seuraavaksi ainoastaan master-koneella poistetaan NameNode-hakemisto ja luodaan uusi oikeilla käyttöoikeuksilla:

$ sudo rm -rf /usr/local/hadoop_tmp
$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
$ sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/
$ sudo chmod 777 /usr/local/hadoop_tmp/hdfs/namenode

5.6. LUODAAN UUSI DATANODE-HEKEMISTO

Tämän jälkeen luodaan ainoastaan slave-koneissa DataNode-hakemisto uudelleen:

$ sudo rm -rf /usr/local/hadoop_tmp
$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
$ sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/
$ sudo chmod 777 /usr/local/hadoop_tmp/hdfs/datanode

5.7. NAMENODEN FORMATOINTI JA HADOOPIN KÄYNNISTYS

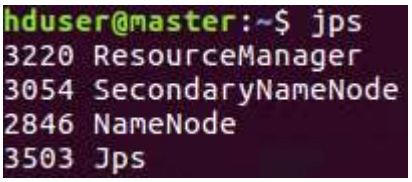
Seuraavaksi formatoi NameNode master-koneella:

$ hdfs namenode -format

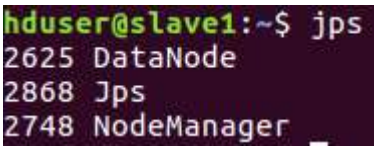
Formatoinnin jälkeen voidaan käynnistää Hadoop-prosessit. Komennot syötetään ainoastaan master-koneelle, joka käynnistää SSH:n avulla tarvittavat prosessit slave-koneissa:

$ start-dfs.sh
$ start-yarn.sh

Kun prosessit ovat käynnistyneet jps-komennolla saadaan tarkastettua master-koneen prosessit, joita on kolme:



Ja samalla komennolla slave-koneelta prosessit, joita on kaksi:



Mikäli kaikki prosessit käynnistyivät onnistuneesti, ne voi nähdä listattuna:

http://master:8088/cluster/nodes
http://master:50070

Solmujen määrää voi myös tutkia komennolla:

$ hdfs dfsadmin -report

5.8. SOLMUN LISÄÄMINEN KLUSTERIIN

Klusterin koon kasvattaminen onnistuu helposti lisäämällä slave-koneita. Kopioimalla olemassa oleva slave-solmu, saadaan valmiiksi konfiguroitu kone, johon joutuu tekemään vain muutamia muutoksia. Ensimmäisenä pitää muistaa vaihtaa uuden koneen hostname sekä katsoa IP-osoite. Tämän jälkeen joudutaan lisäämään uuden koneen tiedot kaikkiin jo olemassa olevien koneiden hosts-tiedostoihin, konfiguroimaan SSH-yhteydet sekä lisäämään uusi kone master-solmun slaves-tiedostoon.

6 HADOOPIN VIANETSINTÄ

Mikäli prosessit eivät lähteneet käyntiin käy konfiguraatiotiedostot uudelleen läpi kirjoitusvirheiden varalta. Jos tiedostot olivat oikein siirry polkuun:

$ cd /usr/local/hadoop/logs

$ ls

Täältä löydät Hadoopin logi-tiedostot. Esimerkiksi jos NameNode-prosessi ei käynnistynyt, etsi listasta kohta hadoop-hduser-namenode-master.log, kopioi se ja anna komento:

$ sudo more hadoop-hduser-namenode-master.log