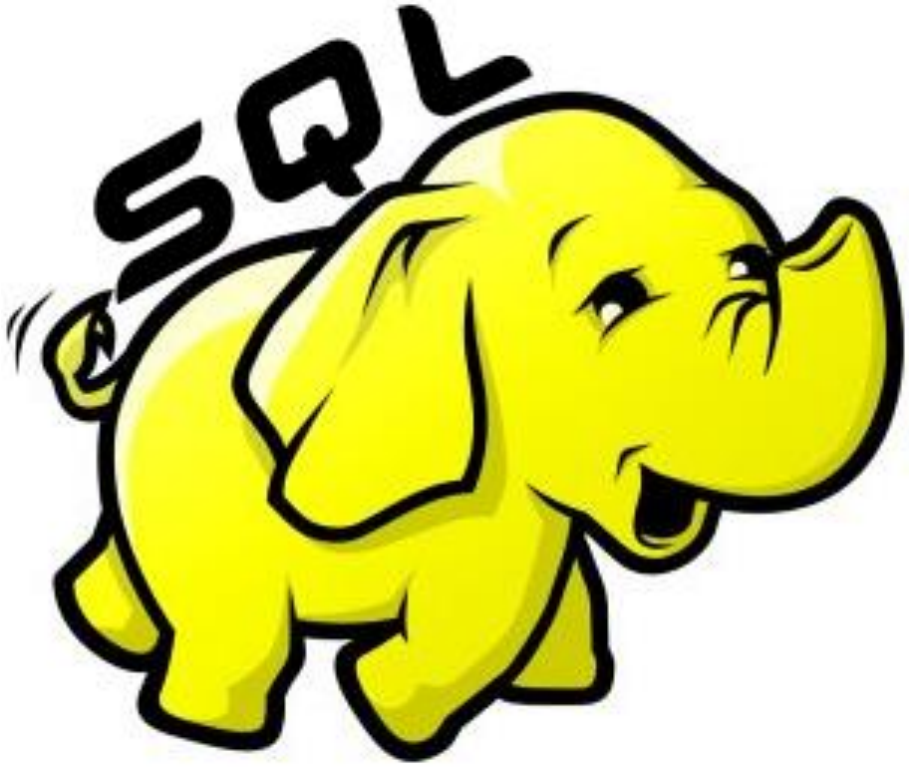
Afstudeerverslag

SQL-on-Hadoop



Titel	Afstudeerverslag
Project/Onderwerp	SQL-on-Hadoop
Auteur	Ruben Zorgman
Studentnummer	11017481
Studie	Informatica
School	Haagse Hogeschool
Eerste examinator:	O. Zor
Tweede examinator:	A. Nederend
Opdrachtgever:	J. Gorter
Afstudeerbegeleidster:	P. Hijl
Technisch Begeleider:	H. Peek
Business Unit manager:	H. Brands
Plaats	Zoetermeer
Datum	05-06-2015
Bedrijf	Info Support B.V.

Referaat

Zorgman R.J.E, Afstudeerverslag – SQL-on-Hadoop, Veenendaal, Info Support BV, 05-06-2015.

Dit afstudeerverslag is geschreven in het kader van het afstuderen voor de opleiding Informatica aan de Haagse Hogeschool te Zoetermeer. De auteur schrijft in dit verslag over zijn werkzaamheden tijdens het afstuderen. Hierbij wordt op beschrijvende en evaluerende wijze ingegaan op de werkzaamheden die in de periode 9 februari tot en met 5 juni 2015 zijn uitgevoerd voor Info Support BV te Veenendaal.

Descriptoren

- SQL-on-Hadoop
- Big Data
- Cloudera Impala
- Pakketselectie
- Onderzoek
- Ad-hoc queries
- Amazon EMR

Voorwoord

Mijn naam is Ruben Zorgman, vierdejaars student HBO Informatica aan de Haagse Hogeschool te Zoetermeer. Het verslag dat voor u ligt is geschreven naar aanleiding van mijn afstudeeropdracht bij Info Support. Dit verslag is bedoeld om de examinatoren Okan Zor, Arno Nederend en de op dit moment nog onbekende extern gecommitteerde inzicht te geven in de activiteiten die de afgelopen periode zijn uitgevoerd tijdens het afstuderen.

In dit voorwoord wil ik graag vanuit Info Support Jesse Gorter, Hylke Peek, Pascalle Hijl en Henk Brands hartelijk bedanken voor de mogelijkheid om bij Info Support af te studeren en de uitstekende begeleiding en bruikbare kritiek tijdens mijn afstudeerperiode. Tijdens deze periode heb ik veel kennis opgedaan en heb ik Info Support leren kennen als een zeer gedreven organisatie met gemotiveerde werknemers om het beste uit jezelf te halen.

Vanuit de opleiding wil ik Okan Zor en Arno Nederend bedanken voor de begeleiding en de nuttige feedback op mijn afstudeerverslag en voor de input tijdens het definiëren van mijn afstudeeropdracht.

Tot slot wil ik Maarten Koene, Bart Bijl, Hanjo de Wit, Allard Soeters en Tom Keim bedanken voor de prettige werksfeer in Zoetermeer.

Ruben Zorgman
Zoetermeer, 05-06-2015

Inhoudsopgave

1. Inleiding

In dit afstudeerverslag wordt beschreven wat er gedurende de afgelopen 17 weken is uitgevoerd. In deze periode is er een afstudeeropdracht voor Info Support uitgevoerd.

Het doel van dit afstudeerverslag is om inzicht te geven aan beide examinatoren en de extern gecommitteerde in de activiteiten en werkzaamheden die tijdens deze afstudeerperiode zijn uitgevoerd. Op deze manier kan er een oordeel worden gegeven over de afstudeeropdracht.

In hoofdstuk *'2. Organisatie*' wordt beschreven hoe de organisatie eruit ziet. Hierbij wordt het bedrijfsprofiel van Info Support samen met de klanten waarvoor projecten worden uitgevoerd beschreven. Hoofdstuk *'3. Opdrachtomschrijving*' wordt de opdracht behandeld. Hier komen de aanleiding, probleemstelling, doelstelling en het resultaat aan bod. Vervolgens wordt in hoofdstuk *'4. Huidige situatie*' de huidige situatie beschreven binnen Info Support. In hoofdstuk *'5. Projectaanpak*' worden de verschillende methodieken die tijdens dit project worden gebruikt behandeld. Ook wordt hier de fasering van dit project beschreven. In hoofdstuk *'6. Onderzoeksontwerp*' wordt ingegaan op de verschillende onderdelen tijdens het onderzoek. Hier wordt beschreven hoe het literatuuronderzoek, interview en experiment worden aangepakt. In hoofdstuk *'7. Requirements*' wordt beschreven hoe de requirements voor de pakketselectie zijn geïnventariseerd. Vervolgens wordt in hoofdstuk *'8. Longlist*' de longlist van de pakketselectie beschreven. Hierin wordt de aanpak, samen met de selectiematrix en het resultaat behandeld. In hoofdstuk *'9. Gegevensconversie*' wordt de gegevensconversie behandeld. Hier wordt beschreven welke dataset wordt geconverteerd en hoe dit is aangepakt. Vervolgens wordt in hoofdstuk *'10. Testen*' beschreven op welke manier de gegevensconversie is getest. In dit hoofdstuk worden ook de kwaliteitsattributen beschreven. In hoofdstuk *'11. Experiment & shortlist*' wordt beschreven hoe het experiment is opgezet. Vervolgens worden hier ook de resultaten en de definitieve keuze beschreven. In hoofdstuk *´12. Demo´* wordt de demo behandeld voor het geselecteerde pakket. Hier is een functioneel en technisch ontwerp voor geschreven. In hoofdstuk *´13. Project afsluiting´* wordt het project afgesloten volgens PRINCE2. In hoofdstuk *´14. Evaluatie´* worden de procesevaluatie, productevaluatie en de beroepstaken behandeld. In bijlage N '*Afkortingenlijst*' is de lijst te vinden met alle afkortingen die worden gebruikt in dit afstudeerverslag.

2. Organisatie

Info Support is een IT-bedrijf met meer dan 400 medewerkers. Het is in 1986 opgericht, met het hoofdkantoor in Veenendaal. Info Support focust zich op het ontwikkelen van maatwerksoftware, Business Intelligence en integratieoplossingen. Verder houdt Info Support zich bezig met het beheer van applicaties en wordt er veel aandacht besteed aan kennis delen. Voor het delen van kennis heeft Info Support een eigen ‘Kenniscentrum’, waar veel verschillende soorten trainingen kunnen worden gevolgd. Daarnaast heeft Info Support een kantoor in België en zijn er traningslocaties in Veenendaal en Utrecht.

Binnen Info Support wordt er veel gebruik gemaakt van technologieën van Microsoft. Hierbij valt te denken aan C#, .NET, maar ook SQL Server stack. Tegenwoordig wordt er ook veel ontwikkeld met Java. Bij de ontwikkeling van software wordt gebruik gemaakt van de, door Info Support ontwikkelde, ‘Endeavour’ ontwikkelstraat.

2.1 Bedrijfsprofiel

Info Support heeft een missie met een aantal kernwaarden. Deze kernwaarden worden hieronder uitgebeeld.

Afbeelding [1]: Soliditeit [1]

Afbeelding [2]: Integriteit [1]

Afbeelding [3]: Vakmanschap [1]

Afbeelding [4]: Passie [1]

Soliditeit komt tot uiting doordat er wordt gestreefd naar kwalitatief hoogwaardige oplossingen. Integriteit komt tot uiting door het nakomen van afspraken en door niet met onverwachte vervelende verassingen aan te komen. Vakmanschap komt tot uiting door de hoge technische kennis en de passie waarmee wordt gewerkt. De passie komt duidelijk naar voren doordat iedereen enthousiast is en dit ook laat blijken.

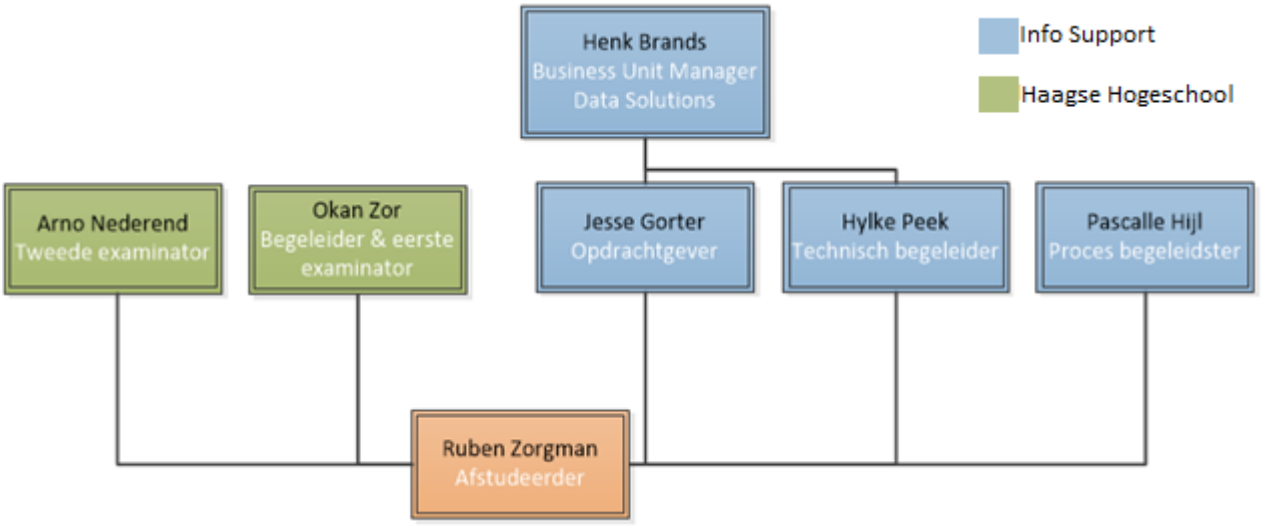
2.2 Klantprofiel

Veel van de opdrachtgevers van Info Support behoren tot de top 500 organisaties in Nederland en België. Info Support opereert niet in één sector. Zo wordt er in de sector overheid, financiën, handel & industrie en zorg & verzekeringen gewerkt. Om een indruk te krijgen voor wat voor soort klanten Info Support opdrachten uitvoert, is in afbeelding [5] een overzicht te zien van de klanten van Info Support.

Afbeelding [5]: Klanten van Info Support [2].

2.3 Organogram

Tijdens het afstuderen maak ik onderdeel uit van de Data Solutions Unit. Voorheen was dit de Business Intelligence Unit. Samen met mij zijn er nog twintig andere afstudeerders in dezelfde periode begonnen. In afbeelding [6] is het organogram te zien. In bijlage B '*PID*' hoofdstuk '*5. Projectmanagementteamstructuur'* worden de verschillende rollen beschreven.



Afbeelding [6]: Organogram.

3. Opdrachtomschrijving

In dit hoofdstuk wordt de opdracht beschreven die tijdens het afstuderen wordt uitgevoerd. Hier wordt ingegaan op de aanleiding, probleemstelling, doelstelling en het resultaat van de afstudeeropdracht.

3.1 Aanleiding

Datawarehousing was tot op heden voornamelijk gericht op het integreren van verschillende gestructureerde bronnen om vervolgens hier analyses op mogelijk te maken. Met de opkomst van Big Data is het mogelijk om te werken met grote hoeveelheden ongestructureerde, maar ook gestructureerde data.

Info Support wil Hadoop graag inzetten in Business Intelligence projecten. Hierbij worden zowel gestructureerde als ongestructureerde bronnen op aangesloten en worden analyses hierop uitgevoerd. Apache Hadoop is een open-source framework en wordt gebruikt om grote hoeveelheden data op te slaan op één of meerdere clusters.

3.2 Probleemstelling

De tekortkoming van Hadoop is dat er niet snel ad-hoc queries kunnen worden gedraaid. Hadoop maakt gebruik van MapReduce, een framework dat een grote taak in kleinere taken opsplitst. Het concept voor dit framework is door Google bedacht en beschreven. Vervolgens heeft Hadoop hier een eigen implementatie van gemaakt. Wanneer een MapReduce taak wordt gestart wordt er bijvoorbeeld eerst bepaald welke taken worden opgesplitst. Vervolgens wordt er een map aangemaakt met deze taken en nog veel meer. Na deze stappen wordt de taak pas verwerkt en uitgevoerd. Dit neemt tijd in beslag, wat vervelend is voor de eindgebruikers, omdat zij vaak spontaan en snel vragen over hun data beantwoord willen hebben.

Op dit moment zijn er diverse initiatieven om dit op te lossen zoals Stinger Initiative. Met deze, of een andere oplossing, zou Hadoop ook in een relatief snelle tijd ad-hoc queries kunnen draaien over hele grote datasets. Het gebruik van SQL om ad-hoc queries uit te voeren is veel toegankelijker dan wanneer er MapReduce taken moeten worden geprogrammeerd in Java.

3.3 Doelstelling

Tijdens het afstuderen is de doelstelling dat er voor Info Support een onderzoek wordt uitgevoerd. In dit onderzoek wordt onderzocht welk pakket het meest geschikt is om ad-hoc queries uit te voeren binnen Hadoop.

3.4 Resultaat

Wanneer de opdracht is afgerond, is er een theoretisch onderzoek uitgevoerd naar het meest geschikte pakket om ad-hoc queries uit te voeren binnen Hadoop. De resultaten van het onderzoek en de pakketselectie worden in een adviesrapport verwerkt.

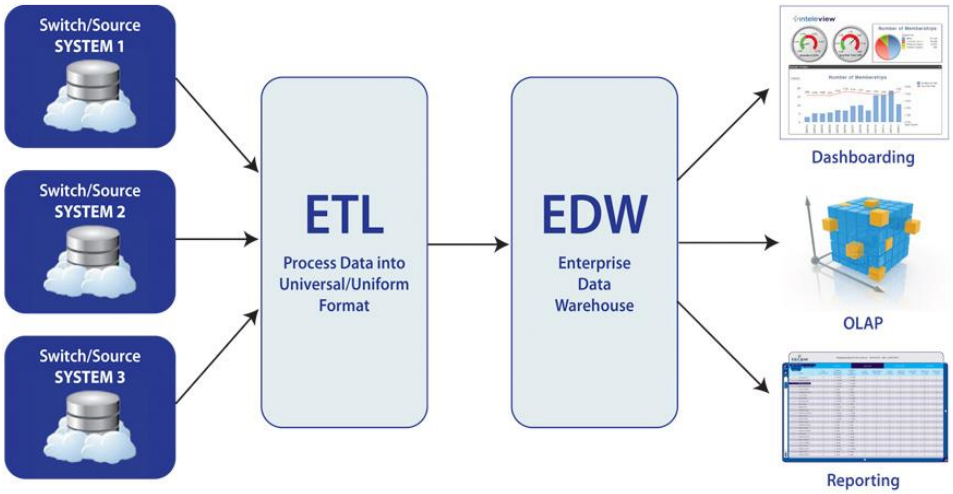
Na het onderzoek wordt er een demo gebouwd voor de gekozen oplossing. Deze demo toont rapportages met behulp van data uit Hadoop. Hierbij moet rekening worden gehouden met het presenteren van de data op een gebruiksvriendelijke manier aan de eindgebruiker en moet er een data conversie worden gedaan om de data in te laden. Daarnaast moet worden aangetoond dat de gegevensconversie van voldoende kwaliteit is. Dit wordt gedaan met behulp van het testrapport.

3.5 Activiteiten & werkzaamheden

1. Plan van aanpak schrijven (tussen de 3 en 5 dagen).
2. Opzetten onderzoek ontwerp (tussen de 2 en 4 dagen).
3. Uitvoeren literatuuronderzoek (tussen de 4 en 6 dagen).
4. Opstellen selectiecriteria pakketselectie (tussen de 4 en 6 dagen).
5. Opstellen longlist (tussen de 3 en 5 dagen).
6. Opstellen shortlist (tussen de 3 en 5 dagen).
7. Omgevingen inrichten voor de initiatieven op de shortlist (tussen de 9 en 11 dagen).
8. Uitvoeren conversie om data in te laden in gekozen initiatief (tussen de 5 en 7 dagen).
9. Bouwen demo (tussen de 10 en 12 dagen).
10. Testrapport opstellen en aantonen dat de rapportages op de data uit het initiatief correct zijn (tussen de 6 en 8 dagen).
11. Afstudeerdossier (15 dagen).

4. Huidige situatie

Binnen Info Support wordt er nog nauwelijks gewerkt met Hadoop. Er zijn daarnaast ook weinig mensen met kennis en ervaring over Hadoop. De Data Solutions unit, voorheen Business Intelligence unit, binnen Info Support houdt zich vooral bezig met het bouwen van traditionele Business Intelligence oplossingen. Veel van deze oplossingen komen neer op de stappen die in afbeelding [7] staan beschreven.



Afbeelding [7]: Overzicht Business Intelligence oplossing [3].

Zoals te zien in afbeelding [7] zijn er verschillende bronsystemen waar de data beschikbaar is. Een bronsysteem kan bijvoorbeeld een klanten database zijn, maar ook een Excel overzicht met verkochte producten. Deze data wordt door middel van ETL uit de bronsystemen opgehaald, aangepast en opgeslagen in een EDW. Tijdens het ETL-proces worden ook de business rules toegepast. Alle data is nu beschikbaar op één plek. Vervolgens wordt deze data gebruikt om verschillende soorten rapportages te genereren. Zo kan er gebruik worden gemaakt van dashboards, OLAP cubes en standaard rapportages.

De dashboards geven een eenvoudig overzicht met verschillende visuele indicatoren hoeveel winst of omzet er wordt behaald. Met OLAP kan de gebruiker op een interactieve manier zelf een overzicht creëren van bijvoorbeeld het aantal verkopen in een bepaald land. Deze manier van rapportages maken wordt ook wel ad-hoc analyse genoemd. Het draait namelijk om een zeer specifieke business vraag, die een gebruiker ter plekke beantwoordt wilt hebben. De standaard rapportages worden door de Business Intelligence specialisten gemaakt en vervolgens beschikbaar gesteld aan de gebruikers. Hier zijn alleen maar overzichten beschikbaar.

Sinds de naamsverandering van de Data Solutions unit, wordt er ook meer gefocust op nieuwe technologieën. Omdat Hadoop ook een redelijk nieuwe technologie is, wordt ook hier naar gekeken hoe dit binnen Business Intelligence projecten bij klanten kan toegepast worden. Wanneer er wordt gekeken op welke manier Hadoop binnen afbeelding [7] past, kan het als extra bron systeem dienen. Op deze manier is het ook mogelijk om rapportages en ad-hoc analyses uit te voeren op de data in Hadoop. Hadoop wordt binnen Info Support dus niet zozeer beschouwd als een vervanging voor het traditionele Data Warehouse, maar eerder als een toevoeging.

5. Projectaanpak

In dit hoofdstuk wordt beschreven welke handelingen er zijn uitgevoerd voor de start van het project. Hierbij wordt ingegaan op de gebruikte methoden en technieken, de risico's die tijdens het project kunnen ontstaan en hoe deze worden afgevangen, de fasering en de planning.

5.1 Methodieken

Tijdens het uitvoeren van het project is er gewerkt volgens een aantal methodieken. Deze worden hieronder beschreven.

5.1.1 Projectmanagement; PRINCE2

Om het project op een gestructureerde manier aan te pakken en te sturen, is er gekeken naar het gebruik van een projectmanagement methodiek. Een methodiek geeft richtlijnen hoe een project gestructureerd kan worden, bijvoorbeeld de fasering, maar ook hoe een project gestuurd kan worden. Om tot een goede keuze te komen, is er eerst gekeken over welke kennis ikzelf beschikte. Met PRINCE2 had ik al ervaring, vanwege het feit dat dit is behandeld tijdens de opleiding. Vervolgens is er op internet gekeken naar andere projectmanagement methodieken. Een van de eerste resultaten was een artikel van de Twynstra Gudde kennisbank, waarbij een aantal projectmanagement methodieken worden vergeleken met elkaar.

1. TGPM: Twynstra Gudde Project Management.
2. PRINCE2: PRojects IN Controlled Environments 2.
3. PMC: Project Management Consultant.
4. PMBOK: Project Management Body of Knowledge

ASPECT	TGPM	PRINCE 2	PMC	PMBOK
INHOUD IN VOLGORDE VAN (EXTERN) IMAGO	faseren, beheersen, beslissen, samen-werken en afstemmen	beheersen, beslissen, afstemmen, samen-werken en faseren	samenwerken, afstemmen, faseren, beslissen en beheersen	managen, beslissen, samenwerken en faseren
GEBRUIK VAN FORMATS EN STANDAARDEN	hulp bij het maken van standaarden en maat-werkformats	uitgebreide set van uniforme formats en standaarden	naruk op cultuur en in-tenties, minder op formats en standaarden	uitgebreide set technieken
WERKINGSGEBIED	alle soorten echte projecten	vooral ict-projecten	vooral projecten voor samenwerking	vooral technische projecten
CENTRALE RICHTING	resultaatgericht managen	managen van projectprocessen	bevorderen van samenwerking	managen van projectprocessen
NADRUK BIJ NVOERING	adviseren, trainen, leren, meedoen, voor-doen	voorschrijven, trainen en certificeren	helpen, opstarten en leren	voorschrijven, trainen en certificeren

Afbeelding [8]: Overzicht verschillende projectmanagement methodieken [4].

Aan de hand van afbeelding [8] is te zien dat er veel overlap zit in de verschillende projectmanagement methodieken. De principes lijken vaak op elkaar: faseren, managen of beheersen en beslissen. Er is wel een duidelijk

verschil merkbaar in het soort projecten waar de verschillende methoden worden toegepast. PRINCE2 wordt vooral binnen ICT-projecten gebruikt, terwijl de andere projecten niet specifiek zijn gericht op ICT-projecten.
Een andere mogelijkheid voor projectmanagement methode had PMBOK kunnen zijn. Deze wordt in afbeelding [8] ook PMBOK beschreven. Dit is naast PRINCE2 een veel gebruikte projectmanagement methodiek. PRINCE2 kan worden gezien als een 'best practice' [5], een methode die zich heeft bewezen in de praktijk. PMBOK kan worden gezien als 'common practice' [5], een methode die vaak wordt gebruikt, maar dit hoeft niet te betekenen dat deze methode zich in de praktijk heeft bewezen. Een ander groot verschil is dat PMBOK kan worden beschouwd als een handleiding, terwijl PRINCE2 echt een methode is, met duidelijke processen en stappen die moeten worden gevolgd [6].

De volgende aspecten hebben ertoe geleid dat er voor PRINCE2 is gekozen als projectmethodiek:

1. Er is veel informatie beschikbaar over PRINCE2. Hierdoor kan er goed worden uitgezocht welke onderdelen er nodig zijn binnen dit project.
2. PRINCE2 is een veel gebruikte projectmanagement methodiek binnen Nederland en Europa [5, 6].
3. Zelf heb ik al ervaring met PRINCE2.
4. PRINCE2 is non-proprietary. Dit houdt in dat er geen handelsrechten of patenten aan zijn verbonden.

Wel zit er een nadeel verbonden aan het gebruik van PRINCE2. Zo zorgt het implementeren van PRINCE2 in een kleiner project voor extra werk. Dit komt voornamelijk vanwege de extra managende stappen die moeten worden ondernomen.

PRINCE2 vindt zijn oorsprong in het Verenigd Koninkrijk. Zoals hierboven al beschreven heeft het een sterke focus op processen. PRINCE2 hanteert de volgende processen:

1. Starting up a Project (SU).
2. Initiating a Project (IP).
3. Directing a Project (DP).
4. Controlling a Project (CS).
5. Managing a Stage Boundary (SB).
6. Managing Product Delivery (MP).
7. Closing a Project (CP).

In hoofdstuk '*5.3 Fasering'* wordt uitgebreider beschreven hoe de processen van PRINCE2 zijn toegepast in dit project.

5.1.2 Onderzoek: 'Wat is onderzoek'

Het uitvoeren van het onderzoek is gedaan aan de hand van het boek '*Wat is onderzoek?'*, van Nel Verhoeven [58]. In dit boek wordt beschreven welke vormen van onderzoek er beschikbaar zijn en hoe dit aangepakt moet worden. Het boek is vooral populair op het HBO, omdat onderzoek een steeds belangrijkere plaats inneemt binnen opleidingen. Binnen de opleiding Informatica wordt dit boek ook gebruikt. Vandaar dat ervoor gekozen is om tijdens dit project dit boek te gebruiken.

In dit boek wordt ook de '*Big 6'* analysemethode beschreven. Deze methode wordt in het project tijdens het literatuuronderzoek gebruikt, om te bepalen of een bron geschikt is om informatie vandaan te halen. Hierin worden een zestal stappen beschreven die hierbij helpen. '*Big 6'* is de enige bekende methode die dit behandeld. In hoofdstuk '*6.2.1 Ontwerp'* wordt beschreven welke stappen dit zijn.

5.1.3 Pakketselectie; KPMG

Om de pakketselectie op een gestructureerde manier te laten verlopen, wordt er gebruik gemaakt van een bestaande pakketselectie methode. Voor het selecteren van een pakketselectie methode is er gekeken op internet. De volgende drie methoden kwamen hierbij naar voren:

1. KPMG pakketselectie.
2. Indora Software en Leverancierselectie.
3. IT-Eye pakketselectie.

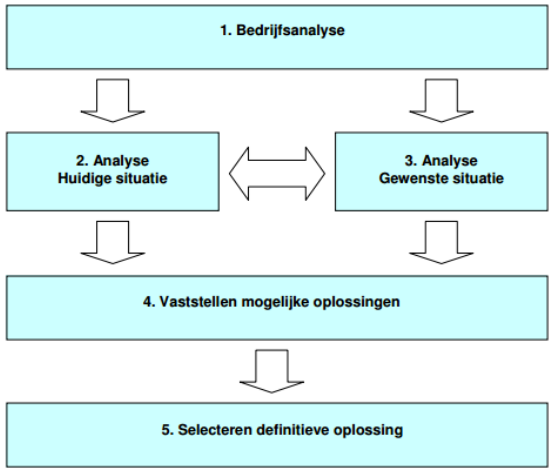
Alle methoden hebben overlap. Zo moeten er requirements, of eisen en wensen vooraf worden opgesteld. Daarnaast wordt er gebruikt gemaakt van 'Longlist' en een 'Shortlist', of iets vergelijkbaars. Over IT-Eye pakketselectie was zeer weinig informatie beschikbaar waarin staat vermeld op welke manier de methode toegepast moet worden. Daarom is er verder gekeken naar KPMG en Indora.

KPMG beschrijft duidelijk welke stappen genomen moeten worden en heeft een heldere en beknopte structuur:

Afbeelding [9]: Pakketselectie volgens KPMG [7].

De stappen in afbeelding [9] kunnen in principe allemaal worden toegepast. Afhankelijk of het pakket dat uit de pakketselectie komt commercieel is of open source, kan ook de laatste stap, contracteren worden toegepast. Wanneer het pakket open source is, vervalt deze laatste stap.

Indora bevat meer onderdelen die tijdens de pakketselectie moeten worden uitgevoerd:



Afbeelding [10]:"Pakketselectie volgens Indora [8].

Stap 5, in afbeelding [10], bestaat uit drie onderdelen:

1. Longlist opstellen, inclusief opstellen knock-out criteria.
2. Shortlist opstellen.
3. Contracteren.

Wanneer er gebruik wordt gemaakt van Indora worden niet alle onderdelen, zoals de 'Bedrijfsanalyse' en 'Vaststellen mogelijke oplossingen' gebruikt. Wel heeft Indora een zeer uitgebreide 'Checklist selectiecriteria'. In deze checklist zitten tientallen criteria beschreven waar rekening mee kan worden gehouden tijdens een pakketselectie. De KPMG methode bevat geen vooraf gedefinieerde lijst met 'KeyCriteria' die als input kan worden gebruikt voor de longlist. Daarom is ervoor gekozen om gebruik te maken van KPMG als methode, vanwege de heldere en beknopte structuur, in combinatie met de checklist van Indora.

In hoofdstuk '*8. Longlist*' wordt beschreven hoe de longlist volgens KPMG is toegepast.

5.1.4 Testen; TMAP Next

Voor het testen van de gegevensconversie is er gekeken of er een methode beschikbaar is. Deze methode moet in ieder geval beschrijven welke kwaliteitscriteria er getest moeten worden. Bij voorkeur ook welke soorten testen hiervoor moeten worden uitgevoerd. Hier was weinig informatie over te vinden. Wel was er een document beschikbaar, van TMAP Next: "*Overzicht toegepaste testvormen*" [33]. Dit document beschrijft welke kwaliteitsattributen getest moeten worden tijdens een gegevensconversietest. Deze kwaliteitsattributen zijn:

1. Volledigheid
2. Juistheid

Buiten TMAP Next zijn er verder geen methoden gevonden die een gegevensconversietest beschrijven. Wel waren er een aantal blogs die beschreven hoe een gegevensconversietest uitgevoerd moest worden, maar dit waren geen testmethoden.

TMAP Next is ontwikkeld door Sogeti en wordt beschouwd als een standaard binnen het testen [9]. Het voordeel van testen volgens een methode als TMAP Next is dat er op een gestructureerde en bewezen manier kan worden getest.

5.2 Risicomanagement

Nr.	Risico omschrijving				
	Oorzaak	Maatregel	P/S*	Wie	Wanneer
1	De afstudeerder komt tijdens de uitvoer van het project erachter dat de kennis van bepaalde technieken en of methoden niet voldoende is.				
	Ontbrekende kennis	Bestuderen van informatie en volgen van tutorials.	S	Afstudeerder	Onderzoek
	Ontbrekende kennis	Volgen van trainingen bij Info Support	P	Info Support	Onderzoek
2	Tijdens het inventariseren van de criteria zijn niet alle wensen van de opdrachtgever meegenomen of veranderen de criteria.				
	Ontbrekende criteria	Inventarisatie sessie goed voorbereiden en doorvragen tijdens de sessie.	P	Afstudeerder	Inventarisatie criteria
	Wijzigende criteria	De opdrachtgever laten verifiëren dat er geen nieuwe criteria bij komen.	P	Afstudeerder	Inventarisatie criteria
3	Het gekozen pakket blijkt niet te voldoen aan de wensen van de opdrachtgever, omdat de pakketselectie niet goed is uitgevoerd.				
	Ontbrekende informatie tijdens pakketselectie	Meerdere bronnen gebruiken voor pakketselectie.	P	Afstudeerder	Pakketselectie
	Subjectieve keuze afstudeerder	De pakketselectie moet objectief worden uitgevoerd aan de hand van de opgestelde criteria.	P	Afstudeerder	Pakketselectie
4	De deadline van een bepaalde fase, of de deadline van het project zelf worden niet behaald.				
	Niet halen van deadlines	Goed van te voren plannen wat er per fase nodig is en controleren of de globale planning nog klopt.	P	Afstudeerder	Gehele project
5	Het uitvoeren van de tests is niet goed uitgevoerd, waardoor er nog mogelijke fouten zijn.				
	Incorrecte tests	Van te voren bepalen hoe er wordt getest en welk doel de testen hebben.	P	Afstudeerder	Testen

Tabel [1]: Overzicht risico's tijdens het afstudeerproject.
*P/S = Preventief of Schadebeperkend

5.3 Fasering

Tijdens het selecteren van de projectmanagement methodiek was al kort beschreven welke zeven fasen er binnen PRINCE2 zijn. Deze fasen zijn op de volgende manier in het project toegepast:

5.3.1 Starting up a Project(SU)

Met deze fase wordt begonnen wanneer er een idee of vraag is om een project te starten. Het is een voorbereidende fase, waarbij in dit project eerst het afstudeerplan is opgesteld en is goedgekeurd door de opdrachtgever en school. Aan het einde van deze fase wordt het afstudeerplan opgeleverd. Nadat het afstudeerplan is goedgekeurd kan verder worden gegaan met de volgende fase.

5.3.2 Initiating a Project (IP)

De tweede fase draait om het opstellen van het Project Initiatie Document (PID). Het PID wordt gemaakt op basis van het goedgekeurde afstudeerplan en geldt als een overeenkomst tussen de afstudeerder en de opdrachtgever. In het PID worden onder andere de business case, de producten, de kwaliteitseisen en project beheersing beschreven. Aan het einde van deze fase wordt het PID opgeleverd. Wanneer het PID is goedgekeurd kan worden begonnen met het onderzoek.

5.3.3 Onderzoek

De onderzoeksfase omvat een groot gedeelte van het project. Omdat er meerdere onderdelen tijdens de onderzoeksfase worden uitgevoerd en de pakketselectie ook parallel loopt met de onderzoeksfase, is ervoor gekozen om een diagram te maken waarin dit visueel wordt getoond. Dit wordt gedaan in afbeelding [11]. Alle documenten en producten die worden opgeleverd tijdens of na het onderzoek zijn in oranje aangegeven. Er zit ook een chronologische volgorde in het diagram.

Onderzoeksontwerp. Als eerste wordt er een onderzoeksontwerp gemaakt. Hierin wordt beschreven hoe het onderzoek wordt uitgevoerd en wat er wordt onderzocht. In afbeelding [11] is het onderzoeksontwerp gekoppeld aan de onderzoeksfase die is weergegeven in het groen. Het onderzoeksontwerp wordt beschreven in bijlage C '*Onderzoeksdocument*' hoofdstuk *'3. Onderzoeksontwerp*' en hoofdstuk *'6. Onderzoek*'.

Literatuuronderzoek. Het literatuuronderzoek wordt gedeeltelijk parallel uitgevoerd met het opstellen van het onderzoeksontwerp. Tijdens het literatuuronderzoek wordt onderzocht hoe Hadoop werkt, wat de techniek achter Hadoop is en wat de knelpunten van Hadoop zijn. Het literatuuronderzoek is bedoeld als vooronderzoek, om kennis op te doen. Het literatuuronderzoek is gekoppeld aan de onderzoeksfase in afbeelding [11], weergegeven in het groen. De resultaten van het literatuuronderzoek worden beschreven in bijlage C '*Onderzoeksdocument*' hoofdstuk *'4. Resultaten literatuuronderzoek*'.

Inventarisatie requirements. Dit is de eerste stap tijdens de pakketselectie. Het is belangrijk dat er met de opdrachtgever wordt overlegd om alle requirements te inventariseren waar naar gekeken moet worden tijdens de pakketselectie. Ook moet tijdens deze inventarisatie een prioritering worden gegeven aan de selectiecriteria door de opdrachtgever. Aan het einde van dit onderdeel is er een rapport met requirements beschikbaar die ook zijn geprioriteerd. Deze requirements worden als input gebruikt voor de longlist. In afbeelding [11] is de inventarisatie van requirements gekoppeld aan de pakketselectie fase die is weergegeven in het blauw.

-**Longlist**. Nadat de requirements zijn geïnventariseerd wordt er verder gegaan met de longlist. De longlist moet het aantal beschikbare pakketten terugbrengen naar een lijst van pakketten die het beste voldoen aan de selectiecriteria. Dit is de tweede stap van de pakketselectie. Het resultaat van de longlist dient als input voor het experiment en de shortlist. De longlist is in afbeelding [11] gekoppeld aan de pakketselectie fase die is weergegeven in het blauw.

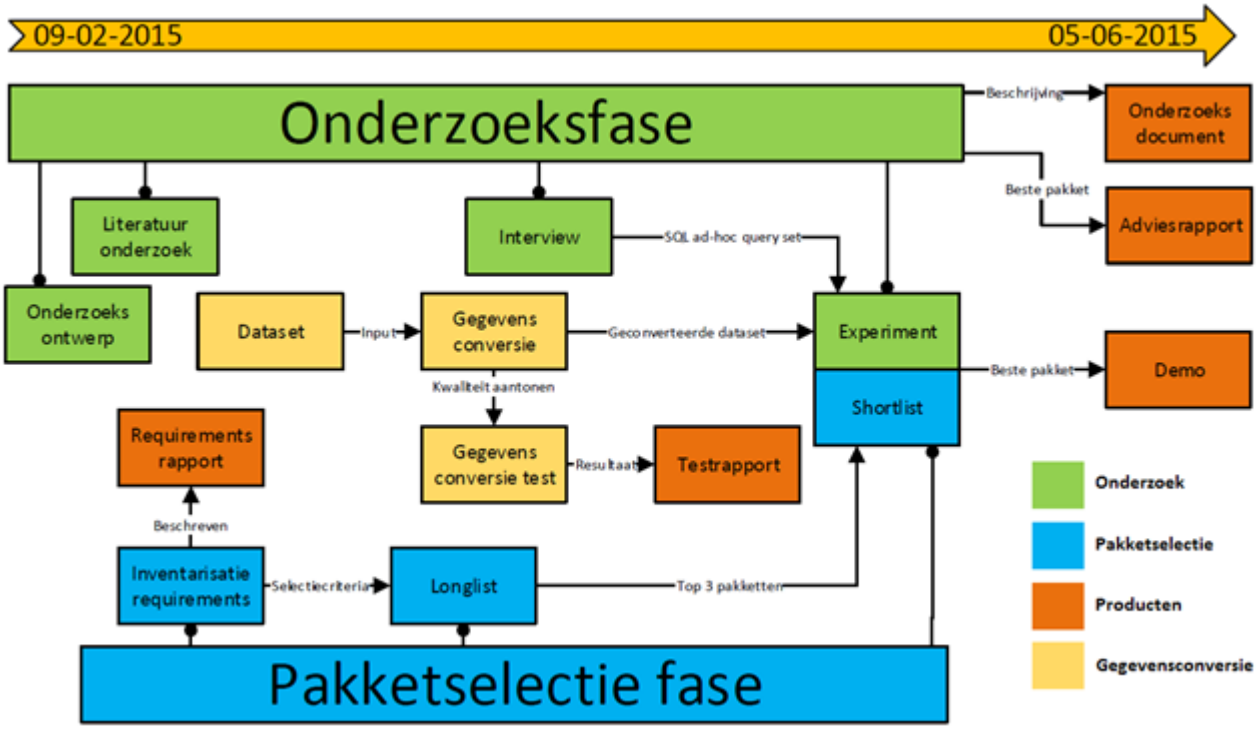
-Interview. Voordat er verder wordt gegaan met het experiment en shortlist, moet er eerst een ad-hoc query set worden gedefinieerd. Deze set aan ad-hoc queries wordt gedefinieerd met behulp van interviews en dient als input voor het experiment. In afbeelding [11] is het interview gekoppeld aan de onderzoeksfase die is weergegeven in het groen. De resultaten van het interview worden beschreven in bijlage C *'Onderzoeksdocument*' hoofdstuk *'5. Resultaten interview*' en in hoofdstuk *'11.1 Ad-hoc query set*'.

-**Gegevensconversie.** Naast een ad-hoc query set is er ook een dataset nodig om de ad-hoc queries op uit te voeren. Deze dataset moet eerst een conversie ondergaan voordat deze gebruikt kan worden. Vervolgens wordt deze dataset getest. Dit wordt gedaan met behulp van TMAP Next. Hierbij wordt gekeken naar de kwaliteitsattributen ´juistheid´ en ´volledigheid´ van de data. Na het opstellen en uitvoeren van de testen is er een testrapport beschikbaar waarin staat beschreven hoe de gegevensconversie is getest en of er voldaan is aan de kwaliteitsattributen. Het resultaat van de gegevensconversie, de geconverteerde dataset, wordt gebruikt als input voor het experiment. De dataset, gegevensconversie en de test zijn in afbeelding [11] in het geel weergegeven.

-**Experiment & shortlist**. Zoals te zien in afbeelding [11] zijn het experiment en de shortlist samengevoegd. Hier is bewust voor gekozen. Het doel van het onderzoek, met behulp van het experiment, is een adviesrapport schrijven welk pakket het meest geschikt is om ad-hoc queries uit te voeren. Het doel van de pakketselectie, door middel van de shortlist, is het selecteren van het meest geschikte pakket om ad-hoc queries uit te voeren. Omdat er twee keer sprake is van hetzelfde doel, is ervoor gekozen om het experiment en de shortlist samen te voegen. Hierbij worden de pakketten op de shortlist gebruikt tijdens het experiment. Aan de hand van de resultaten van het experiment wordt er een adviesrapport geschreven.

Het experiment zal in een gecontroleerde omgeving plaatsvinden. Hier worden eerst queries uitgevoerd op Hadoop zonder pakket. Hier wordt de set van ad-hoc queries gebruikt. De queries worden uitgevoerd op de dataset die van te voren met behulp van de gegevensconversie is overgezet. De resultaten die hieruit komen worden als nulmeting gebruikt. Daarna zal dezelfde set queries worden uitgevoerd op de verschillende pakketten die op de shortlist staan beschreven. Wanneer het meest geschikte pakket een open source pakket is, hoeft de laatste stap van de pakketselectie, het contracteren, niet te worden uitgevoerd.

Naast het adviesrapport wordt er ook een onderzoeksdocument geschreven, met hierin alle resultaten, van het literatuuronderzoek, het interview en het experiment. Voor het geselecteerde pakket wordt vervolgens een demo ontwikkeld.



Afbeelding [11]: Overzicht indeling en fasering onderzoek.

5.3.4 Ontwikkeling demo

Voor het meest geschikte pakket dat uit het onderzoek komt, wordt een demo gemaakt. Deze demo moet aantonen dat het mogelijk is om rapportages uit te kunnen voeren op de data uit HDFS. Hiervoor worden verschillende visualisatie tools gebruikt. Voor de demo is het belangrijk dat er ad-hoc analyses kunnen worden uitgevoerd en moet er rekening mee worden gehouden dat de data op een gebruiksvriendelijke manier wordt getoond. Er worden hiervoor twee technieken gebruikt, namelijk Tableau en Microsoft Excel Powerpivot. De grote concurrent van Tableau is Qlikview. De reden dat er voor Tableau is gekozen, komt omdat het inladen van de data eenvoudiger gaat bij Tableau dan bij Qlikview. Microsoft Excel Powerpivot is gekozen om dit een van de meest gebruikte tools is om rapportages te maken.

5.3.5 Closing a Project (CP)

Om het project op een gestructureerde manier af te sluiten, wordt het laatste proces van Prince2, ‘Closing a Project’ gebruikt. Tijdens deze fase vindt de overdracht plaats van het projectresultaat. Dit houdt in dat het adviesrapport, wordt overgedragen aan Info Support. Bij deze overdracht hoort ook de demo. Ook worden alle overige documenten opgeleverd.

5.4 Planning

Onderdeel	Begin datum	Eind datum	Benodigd aantal uur
Project Initiatie Document	09-02-2015	18-02-2015	64
Opzetten onderzoeksontwerp	19-02-2015	24-02-2015	32
Uitvoeren literatuuronderzoek	25-02-2015	27-02-2015	24
Onderzoeksfase*	02-03-2015	10-04-2015	240
Opstellen requirements	02-03-2015	05-03-2015	32
Opstellen longlist	06-03-2015	11-03-2015	32
Opstellen shortlist	12-03-2015	16-03-2015	24
Omgeving inrichten voor pakketten op shortlist	17-03-2015	01-04-2015	88
Uitvoeren conversie	02-04-2015	08-04-2015	40
Uitvoeren experiment	09-04-2015	10-04-2015	16
Ontwikkeling demo	13-04-2015	29-04-2015	104
Opstellen testrapport	30-04-2015	08-05-2015	56
Uitloop	11-05-2015	19-05-2015	56
Afstudeerverslag	20-05-2015	05-06-2015	128

Tabel[2]: Overzicht initiële planning uit het Plan van Aanpak.
** Binnen de onderzoeksfase worden alle dikgedrukte onderdelen uitgevoerd.*

6. Onderzoek

Voordat de pakketselectie was begonnen, is er eerst een onderzoeksontwerp opgesteld en een literatuuronderzoek uitgevoerd. Dit wordt behandeld in dit hoofdstuk. Hier is voor gekozen, omdat er namelijk geen voorkennis was van Hadoop. Met behulp van de opgedane kennis kan de pakketselectie beter worden uitgevoerd. Het complete onderzoeksdocument is te vinden in bijlage C '*Onderzoeksdocument*'.

6.1 Probleemstelling

Voorafgaand aan het onderzoek, is er eerst een onderzoeksontwerp opgesteld. Dit is gedaan volgens de stappen die zijn beschreven in het boek '*Wat is onderzoek* [58]'. Het opstellen van een onderzoeksontwerp zorgt ervoor dat er niet direct wordt begonnen met het onderzoek uitvoeren, maar dat er eerst bedacht en beschreven wordt wat er met het onderzoek bereikt moet worden en hoe dit wordt gedaan. In deze paragraaf wordt de probleemstelling van het onderzoeksontwerp behandeld.

6.1.1 Doelstelling

Het doel van dit onderzoek is het opstellen van een adviesrapport waarin wordt beschreven welk pakket het meest geschikt is om ad-hoc queries uit te voeren binnen Hadoop voor Info Support BV.

Het resultaat van het onderzoek is voor de volgende twee groepen interessant:

1. Info Support BV kan de informatie uit het adviesrapport gebruiken om een keuze te maken of Hadoop binnen Business Intelligence projecten gebruikt gaat worden en welk pakket het meest geschikt is om ad-hoc SQL queries op Hadoop te kunnen uitvoeren.
2. De afstudeerder kan de informatie uit het vooronderzoek gebruiken bij de uitvoering van het experiment. De resultaten uit het experiment zijn bruikbaar voor een definitieve keuze van een pakket.

De bruikbaarheid van het onderzoek is hoofdzakelijk instrumenteel. Dit houdt in dat het adviesrapport kan bijdragen aan het uitstippelen van het beleid op het gebied van nieuwe technologieën die worden gebruikt binnen Business Intelligence projecten.

6.1.2 Scope

Het literatuuronderzoek zal zich beperken tot Hadoop en het ecosysteem rond Hadoop. Er wordt dus niet gekeken naar andere projecten of applicaties die MapReduce hebben geïmplementeerd. Hier is voor gekozen, omdat de opdrachtgever alleen voor Hadoop wil kijken of het mogelijk is om ad-hoc queries op uit te voeren.

Het experiment zal zich beperken tot een Hadoop omgeving in de cloud. Dit is vanwege de volgende redenen:

1. Er is een investering nodig in 'on-premises' hardware om een test omgeving op te zetten. On-premises betekent in het gebouw zelf, fysieke hardware.
2. De opdrachtgever heeft aangegeven dat de implementatie bij klanten vaker in de cloud zal gebeuren dan met 'on-premises' hardware.

Tijdens het experiment wordt geen ongestructureerde data ingeladen. Hadoop wordt vaak geassocieerd met het verwerken van ongestructureerde data. Dit komt doordat Hadoop hier namelijk voor geschikt is. Alleen is er in overleg met de opdrachtgever voor gekozen om een criteria als 'Welk pakket is het meest geschikt om op unstructured data queries uit te voeren?' niet te behandelen tijdens het experiment. Dit komt namelijk doordat het onderzoek zich focust op 'SQL-on-Hadoop' pakketten. Wanneer er gebruik wordt gemaakt van SQL (Structured Query Language) op ongestructureerde data is er nog een tussenstap nodig. Deze tussenstap bestaat uit het bruikbaar maken van de ongestructureerde data om SQL queries op uit te voeren. Dit wordt gedaan door middel van het definiëren van een schema, met de tabel naam, de verschillende attributen in de tabel en uit welk data type deze attributen bestaan. Op deze manier is er niet meer te spreken van ongestructureerde data, maar van gestructureerde data.

6.1.3 Hoofdvraag

Welke van de bestaande 'SQL-on-Hadoop' pakketten maakt Hadoop het meest geschikt om ad-hoc queries op uit te voeren?

6.1.4 Deelvragen

1. Wat is Hadoop?
    a. Wat is Big Data?
    b. Hoe werkt MapReduce?
    c. Hoe werkt HDFS?
2. Welke tekortkomingen heeft Hadoop?
    a. Waarom zijn ad-hoc queries langzamer in Hadoop dan in traditionele RDBMS's?
3. Hoe zitten de pakketten op de shortlist technisch in elkaar?
    a. Welke tekortkomingen hebben deze pakketten?
4. Welke structuur hebben ad-hoc queries?
5. Welk pakket heeft de snelste ad-hoc query performance?
    a. Welk pakket heeft gemiddeld de snelste ad-hoc query performance?
6. Welk pakket is het beste schaalbaar?

Bij het opstellen van de deelvragen is er voornamelijk gekeken hoe deze bijdragen aan het helpen beantwoorden van de hoofdvraag. De deelvragen zijn chronologisch opgesteld. De eerste drie deelvragen worden tijdens het literatuuronderzoek behandeld. Het resultaat van de vierde deelvraag dient als input voor het experiment. De vijfde en zesde deelvraag worden tijdens het experiment onderzocht.

De eerste deelvraag, inclusief de bijbehorende onderzoeksvragen zijn van belang om kennis op te doen over de werking van Hadoop. Zonder deze kennis is het aanzienlijk lastiger om het onderzoek uit te kunnen voeren. Verder is ervoor gekozen om te focussen op de belangrijkste aspecten van Hadoop: MapReduce en HDFS. Aangezien deze kennis tijdens het onderzoek ook nodig is.

De tweede deelvraag, inclusief onderzoeksvraag, is van belang voor het hele onderzoek. Tijdens het experiment wordt er onderzocht hoe ad-hoc queries snel op Hadoop uitgevoerd kunnen worden, met behulp van een pakket. Hierbij is het van belang om eerst te weten waarom ad-hoc queries in Hadoop langzaam zijn, wat de oorzaak hiervoor is.

De derde deelvraag, inclusief onderzoeksvraag wordt uitgevoerd tijdens het experiment en de shortlist. Er is bewust voor gekozen om deze deelvraag te behandelen voor de pakketten op de shortlist, omdat met deze pakketten het

experiment wordt uitgevoerd. Hierbij is het zeer handig om te begrijpen hoe deze pakketten technisch in elkaar zitten. Dit vereenvoudigd het gebruik van de pakketten namelijk en kan ertoe bijdragen dat eventuele problemen sneller worden opgelost. Wanneer er ernstige tekortkomingen naar boven komen, zou dit ook invloed kunnen hebben op de keuze voor het meest geschikte pakket.

De vierde deelvraag is een zeer belangrijk deelvraag. Het experiment, inclusief de resultaten zijn namelijk afhankelijk van een ad-hoc query set. Om deze ad-hoc query set te maken moet er worden onderzocht welke structuur deze ad-hoc queries hebben.

De vijfde en zesde deelvraag zijn belangrijk voor het selecteren van het meest geschikte pakket. De opdrachtgever had aangegeven dat hij het belangrijk vond dat de snelheid en schaalbaarheid onderzocht werd. Vandaar dat ervoor gekozen is om deze onderwerpen als deelvragen op te nemen.

6.2 Literatuuronderzoek

Binnen dit onderzoek wordt een aantal van de deelvragen beantwoord door middel van een literatuuronderzoek. De eerste vier deelvragen zijn gericht op het vooronderzoek, bedoeld om kennis op te doen.

De reden dat er voor literatuuronderzoek is gekozen om deze deelvragen te beantwoorden, is vanwege de hoeveelheid tijd die tijdens het afstuderen beschikbaar is. Het vinden van bronnen op internet en informatie uit boeken is veel eenvoudiger en is ook betrouwbaar. Zeker wanneer er verschillende bronnen worden gebruikt om dezelfde vraag te beantwoorden. De eerste vier deelvragen zouden gedeeltelijk of geheel kunnen worden beantwoord door middel van interviews met experts. Alleen voor deze interviews zijn er experts nodig op het gebied van Hadoop. Binnen Info Support BV zijn er geen Hadoop experts beschikbaar met diepgaande kennis. Daarom zouden er externe experts moeten worden gezocht. Hier moeten dan afspraken mee worden gemaakt. Daarnaast zou er voor een betrouwbaar resultaat meerdere experts nodig zijn. Het inplannen en uitvoeren van meerdere interviews neemt veel tijd in beslag.

6.2.1 Ontwerp

De volgende deel- en onderzoeksvragen worden gedeeltelijk of geheel beantwoord door middel van een literatuuronderzoek:

1. Wat is Hadoop?
    a. Wat is Big Data?
    b. Hoe werkt MapReduce?
    c. Hoe werkt HDFS?
2. Welke tekortkomingen heeft Hadoop?
    a. Waarom zijn ad-hoc queries langzamer in Hadoop dan in traditionele RDBMS's?
3. Welke bedrijven maken gebruik van Hadoop?
4. Hoe zitten de pakketten op de shortlist technisch in elkaar?
    a. Welke tekortkomingen hebben deze pakketten?

Bij het literatuuronderzoek wordt er gebruik gemaakt van de '*Big6*' zoekmethode. Deze methode voor het analyseren van bronnen wordt uitgebreider beschreven in bijlage C '*Onderzoeksdocument*' hoofdstuk '*3.1.1 Ontwerp*'.

Tijdens de pakketselectie zal er literatuuronderzoek plaatsvinden om de volgende deelvraag en onderzoeksvraag te beantwoorden:

1. *'Hoe zitten de pakketten op de shortlist technisch in elkaar?'*
2. *'Welke tekortkomingen hebben deze pakketten?'*

De onderzoeksvraag *'Welke tekortkomingen hebben deze pakketten?'* kan ook gedeeltelijk worden beantwoord door middel van observaties tijdens het experiment voor de pakketselectie. Op het moment dat er tijdens het experiment bepaalde tekortkomingen naar voren komen, zal dit worden gerapporteerd.

6.2.2 Betrouwbaarheid bronnen

Bij literatuuronderzoek is het belangrijk dat de informatie uit de gebruikte bronnen betrouwbaar is. Om een goed literatuuronderzoek uit te kunnen voeren worden de volgende punten in acht genomen bij het beoordelen van een bron:

1. Wie is de auteur en is hij/zij nog actief op dit vakgebied?
2. Worden feiten/aannames onderbouwd door middel van bronverwijzingen?
3. Wanneer is de bron geschreven?

De bronnen die worden gebruikt binnen het literatuuronderzoek zullen voornamelijk voortkomen uit secundaire literatuur en grijze literatuur.

6.3 Interview

Voordat het experiment kan worden uitgevoerd wordt er eerst een interview gehouden met verschillende deskundigen binnen Info Support. Deze deskundigen zijn Business Intelligence experts. Het enige doel van dit interview is om informatie te verzamelen die kan worden gebruikt bij het beantwoorden van de volgende deelvraag:

1. *Welke structuur hebben ad-hoc queries?*

Het belangrijkste topic dat aan bod komt tijdens dit interview is:

1. Structuur van de ad-hoc queries.

Met behulp van de informatie uit het interview kan er worden bepaald wat voor structuur ad-hoc queries hebben en welke SQL-condities hierin worden gebruikt. Tijdens het interview wordt ook gevraagd of het mogelijk is om ad-hoc queries te analyseren uit projecten bij klanten, waar Info Support zit. Op deze manier kan er worden bekeken welke ad-hoc queries binnen Info Support worden gebruikt.

Naast het houden van interviews, was het niet gelukt om een betrouwbaar overzicht te vinden met daarin de meest gebruikte ad-hoc queries. Er was één blog beschikbaar, getiteld '*SQL queries: The top 10 most used*' [10], waarbij een aanname werd gedaan dat de queries uit het artikel 90% van alle benodigde queries zouden zijn. Er was geen bronvermelding vermeld naar het onderzoek hoe dit percentage tot stand was gekomen.

Wel zijn er verschillende benchmark scripts beschikbaar. Hierin worden business vragen nagebootst door middel van queries. Er is voor gekozen om de queries van de industrie standaard benchmark te analyseren [11, 12].

Deze queries worden vergeleken met de queries van Info Support, om te kijken of deze overeenkomen. Wanneer de structuur van de queries uit de twee verschillende bronnen onderling weinig verschillende, kan de structuur van deze queries worden gebruikt voor een eigen set aan ad-hoc queries.

6.4 Experiment

Het experiment zal worden uitgevoerd om de laatste twee deelvragen te kunnen beantwoorden.

1. Welk pakket heeft de snelste ad-hoc query performance?
    a. Welk pakket heeft gemiddeld de snelste ad-hoc query performance?
2. Welk pakket is het beste schaalbaar?

6.4.1 Operationalisatie

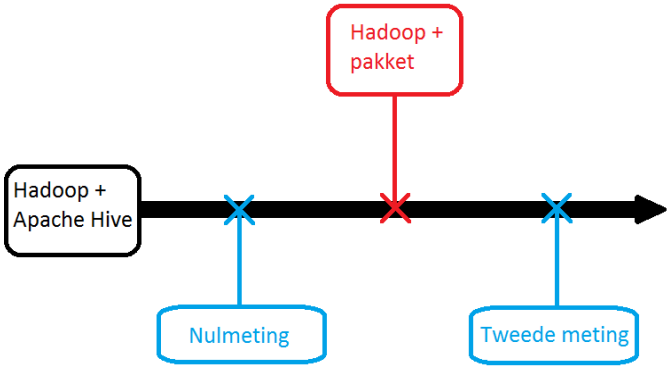
Om de begrippen die tijdens de deel- en onderzoeksvragen zijn behandeld uit te werken naar meetbare instrumenten wordt er gebruik gemaakt van operationalisatie. Tijdens het experiment zijn er twee begrippen die geoperationaliseerd moeten worden, namelijk '*snelste*' en '*schaalbaar*'. Om dit te realiseren is er gebruik gemaakt van de volgende operationalisatie:

Begrip	Dimensie	Indicator
Snelste	Tijd	Seconden
Schaalbaar	Tijd	Percentage

Tabel [3]: Operationalisatie van begrippen.

De schaalbaarheid wordt gemeten door het procentuele verschil te berekenen tussen twee metingen. Dit leek mij de enige mogelijkheid om de schaalbaarheid te meten. Tijdens het experiment zullen de resultaten van de verschillende pakketten worden gemeten in seconden. De afronding zal plaatsvinden op drie decimalen achter de komma, zodat ook het aantal milliseconden kan worden meegenomen.

6.4.2 Opzet experiment



Afbeelding [12]: Overzicht verschillende periode voor metingen.

Wanneer er een set aan ad-hoc queries is bepaald, kan de nulmeting uitgevoerd worden. Dit wordt gedaan op een cloud omgeving waar Hadoop geconfigureerd is. Deze configuratie moet een versie van Apache Hive bevatten die niet is 'versneld' met behulp van het Stinger Initiative. Hier is bewust voor gekozen, omdat Apache Hive de eerste

'SQL-on-Hadoop' oplossing was. Het doel van de nulmeting is om te bekijken hoeveel tijd er nodig is per ad-hoc query om uitgevoerd te worden. De resultaten uit de nulmeting zullen later worden gebruikt bij het vergelijken van de resultaten van de pakketten.

Vervolgens wordt Hadoop geconfigureerd met de verschillende pakketten. Per pakket zal dezelfde set aan ad-hoc queries worden uitgevoerd. Dit is de tweede meting zoals die staat beschreven in afbeelding [12]. Wanneer het experiment is uitgevoerd kan er worden onderzocht of er verschil is qua performance tussen Hadoop zonder pakket en Hadoop met pakket en de onderlinge verschillen per pakket.

6.4.3 Randvoorwaarden

De volgende randvoorwaarden zijn gesteld tijdens het experiment:

Dataset
Er is een representatieve dataset nodig met gestructureerde data om de ad-hoc queries op uit te kunnen voeren. Hiervoor wordt een aangepaste versie van de AdventureWorksDW2012 dataset gebruikt. Op deze manier kan er worden bekeken wat voor impact de grootte van de dataset heeft op de performance. Hoe de dataset tot stand is gekomen, staat beschreven in hoofdstuk '*9.1 Dataset*'.

Dataset in HDFS
De SQL-on-Hadoop pakketten kunnen geen queries uitvoeren op SQL Server. Om de dataset te kunnen gebruiken moet er een conversie worden uitgevoerd naar een HDFS, de opslag structuur die gebruikt wordt binnen Hadoop. Dit moet gedaan worden met behulp van SQOOP, een tool om relationele SQL tabellen om te zetten naar HDFS en andersom. Meer informatie over de keuze van SQOOP staat beschreven in hoofdstuk '*9.2 SQOOP*'.

Cloud omgeving
De omgeving waar het experiment wordt uitgevoerd is een cloud omgeving. In de praktijk zal Hadoop ook in de cloud worden gebruikt bij klanten. Daarnaast is het eenvoudiger om een cloud omgeving op te zetten en eventueel uit te breiden, vergeleken met dedicated hardware. De opdrachtgever had geen voorkeur voor een bepaalde cloud omgeving. Daarom is er gekeken welke cloud omgeving de beste ondersteuning bied voor de installatie van de verschillende pakketten.

Zowel Microsoft Azure als Amazon AWS hebben beiden mogelijkheden om een Hadoop cluster in de cloud op te zetten. Microsoft Azure wordt door Info Support zelf gebruikt en ook bij verschillende klanten. Na overleg met de opdrachtgever is er besloten om de Hadoop cluster niet op Microsoft Azure, maar op Amazon AWS te configureren. De reden hiervoor was dat er geen installatie script of handleiding beschikbaar was voor Apache Drill op Microsoft Azure. Zowel Microsoft Azure als Amazon AWS hebben 'out-of-the-box' configuraties van Apache Hive en Cloudera Impala. Amazon AWS heeft een duidelijk installatie script, inclusief handleiding beschikbaar voor Apache Drill.

Hardware
De hardware moet bij de uitvoering van het experiment voor alle drie de pakketten identiek zijn. Omdat er vanuit de opdrachtgever een wens was om te kijken wat voor impact een verdubbeling van het aantal nodes op de performance had van de ad-hoc queries, is in tabel [4] een overzicht gemaakt van de verschillende hardware configuraties voor het experiment.

Run	Dataset	Aantal nodes
1	**10 Miljoen records**	2
2	**10 Miljoen records**	4
3	**10 Miljoen records**	8
4	**100 Miljoen records**	2
5	**100 Miljoen records**	4
6	**100 Miljoen records**	8

Tabel [4]: Overzicht hardware configuratie

Zoals in tabel [4] te zien is, zal per hardware configuratie worden getest op beide datasets. Dit is gedaan om te kijken wat voor impact een grotere dataset heeft op de query performance. De keuze voor twee, vier en acht nodes is in overleg met de opdrachtgever gedaan. Hieruit kwam naar voren dat er met twee nodes wordt begonnen en dit wordt vervolgens elke keer verdubbeld. Op deze manier kan de schaalbaarheid ook goed kan worden onderzocht. De reden dat er geen zestien nodes zijn gebruikt, komt vanwege de kosten die dit met zich meebrengt.

Cluster verwijderen

Nadat een run van het experiment is uitgevoerd, moet de cluster waarop het experiment is uitgevoerd worden verwijderd. Op deze manier is het niet mogelijk dat wanneer een nieuwe run van het experiment wordt uitgevoerd er bijvoorbeeld gecachede queries bewaard zijn gebleven van de vorige run.

6.4.4 Zuiver experiment

Het experiment is een zuiver experiment. Dit houdt in dat het in een speciaal gecreëerde situatie wordt uitgevoerd. Deze omgeving is Amazon Elastic MapReduce (EMR), waar Hadoop op is geïnstalleerd. Hier worden vervolgens de pakketten op geïnstalleerd, zonder eventuele extra software. Op deze manier is er zoveel mogelijk geprobeerd om invloeden van buitenaf te beperken en heeft alleen het pakket invloed op de resultaten waardoor er zo min mogelijk ‘derde’ variabelen aanwezig zijn.

Een andere mogelijkheid voor het experiment zou zijn dat de complete Hadoop distributie van een pakket wordt geïnstalleerd op Amazon EC2. De pakketten die bij een bepaalde Hadoop distributie horen staan hieronder beschreven:

1. Stinger Initiative - Hortonworks distributie.
2. Cloudera Impala – Cloudera distributie.
3. Apache Drill – MapR distributie.

De installatie van de Hadoop distributies, met bijbehorend pakket, op Amazon EC2 is even eenvoudig vergeleken met alleen de pakketten op Amazon EMR te installeren. De Hadoop distributies zijn namelijk ook beschikbaar op Amazon EC2. Het nadeel hiervan is dat de distributies zijn geoptimaliseerd voor het bijbehorende pakket. Zo heeft MapR een eigen File System, genaamd MapR-FS. Een voorbeeld van een optimalisatie in MapR-FS, is dat het in de programmeertaal C is geschreven [13]. In tegenstelling tot HDFS dat in Java is geschreven. Doordat de File System in een andere programmeertaal is ontwikkeld kan dit invloed hebben op de performance en dus op de resultaten van het experiment.

Tijdens het experiment is er sprake van tenminste één controlegroep en één experimentele groep, namelijk de ad-hoc query set die op de oude versie van Apache Hive wordt uitgevoerd en de ad-hoc query set die op de pakketten wordt uitgevoerd.

Daarnaast is ervoor gekozen om het experiment twee keer uit te voeren. Wanneer het experiment één keer zou worden uitgevoerd zouden de resultaten teveel op toeval kunnen berusten. Wanneer het experiment twee keer

wordt uitgevoerd en de resultaten van de eerste en tweede keer komen goed overeen met elkaar, is de kans aanzienlijk kleiner dat de resultaten na twee keer op toeval berusten. De reden dat het experiment niet meer dan twee keer wordt uitgevoerd is omdat de opdrachtgever akkoord ging met de resultaten wanneer het experiment twee keer werd uitgevoerd en de resultaten geen grote verschillen lieten zien. Hierbij is in overleg met de opdrachtgever bepaald dat wanneer de gemiddelde resultaten meer dan 10% afwijking hebben het experiment nog een keer moet worden uitgevoerd. Daarnaast neemt het opzetten, uitvoeren en rapporteren van één run van het experiment ongeveer anderhalve dag tot twee dagen in beslag en kost het ongeveer $100.

Omdat het experiment op een Amazon EC2 instantie wordt uitgevoerd, de instanties waar Amazon EMR op draait, kunnen er in zeldzame gevallen performance issues optreden [14]. De hardware van deze instanties wordt namelijk met andere gebruikers gedeeld, tenzij er sprake is van een 'dedicated instance' [15]. Om te bekijken of het tijdstip van de uitvoer invloed op de resultaten heeft, wordt het experiment de eerste keer in de ochtend uitgevoerd en de tweede keer in de middag. Wanneer de resultaten van de tweede uitvoer overeenkomen met de eerste uitvoer van het experiment, dan kan er beter worden aangetoond dat de resultaten niet op toeval berusten en dat het tijdstip van de uitvoer ook geen invloed heeft. Verder wordt hiermee aangetoond dat het experiment herhaalbaar is. Dit is een belangrijk eigenschap van een experiment volgens het handboek '*Hoe doe ik onderzoek*'.

6.5 Populatie

6.5.1 Interview

De populatie die wordt gebruikt voor het interview zijn Business Intelligence werknemers binnen Info Support. De reden dat er voor Business Intelligence werknemers is gekozen, is omdat zij veel ervaring hebben op het gebied van ad-hoc queries. Hier is bewust voor gekozen, omdat de opdracht voor Info Support wordt uitgevoerd. De enige kenmerk voor de populatie is dat ze geen 'junior' developer of consultant meer zijn. Dit is gedaan omdat 'junior' developers of consultants vaak minder ervaring hebben dan 'medior', of 'senior' developers/consultants.

6.5.2 Experiment

De populatie voor het experiment zijn ad-hoc SQL queries, want over deze groep moet een uitspraak worden gedaan. Omdat er oneindig veel combinaties mogelijk zijn voor ad-hoc SQL queries, zal de enige kenmerk van de populatie worden dat het ad-hoc SQL queries moeten zijn waarbij een 'SELECT' statement wordt uitgevoerd. Dit wordt ook wel de operationele populatie genoemd. 'CREATE', 'UPDATE', 'DELETE' en 'INSERT' vallen buiten de populatie. De oneindige combinatie van mogelijkheden komt doordat elke query net wat kan verschillen, doordat er een andere combinatie van condities en of functies wordt gebruikt.

6.6 Steekproef

6.6.1 Interview

Voor de steekproef van de interview, wordt verwacht dat er vijf werknemers nodig zijn. Deze moeten overeenkomen met de kenmerken van de populatie die in het vorige hoofdstuk zijn beschreven. Wanneer de benodigde kennis eerder is verworven, zullen er minder interviews worden gehouden. Dit kan gebeuren wanneer er

bij twee opeenvolgende interviews geen nieuwe informatie meer wordt verteld. Als dit niet het geval is, zullen er meer interviews worden gehouden.

6.6.2 Experiment

Doordat er een oneindige combinatie aan SQL queries is, zal er een selecte steekproef plaatsvinden, in de vorm van een *'doelgerichte'* steekproef. Dit houdt in dat de steekproef niet willekeurig gekozen is, maar dat de steekproef wordt uitgevoerd op basis van bepaalde kenmerken. Het kenmerk voor dit experiment is de frequentie van de verschillende SQL functies en commando's binnen de ad-hoc queries. Aan de hand van deze frequentie zullen er ad-hoc queries worden gemaakt voor het experiment. Deze zelf gedefinieerde ad-hoc queries moeten voldoen aan de kenmerken van de populatie die hierboven staan beschreven.

De steekproefomvang voor dit experiment zal bestaan uit eenentwintig queries. De verantwoording voor deze keuze is te vinden in hoofdstuk '*11.1 Ad-hoc query set'*.

6.7 Analysemethoden

6.7.1 Vooronderzoek

De resultaten uit het vooronderzoek zullen worden verwerkt in tekstvorm, zodat de deel – en onderzoeksvragen beantwoord kunnen worden.

6.7.2 Interview

Voor de verwerking van de resultaten uit de interviews worden de volgende stappen gehanteerd:

1. Tijdens de interviews worden de ad-hoc queries beschikbaar gesteld.
2. Deze ad-hoc queries zullen worden geanalyseerd aan de hand van de frequentie van de SQL functies en commando's. Dit wordt gedaan met behulp van http://www.writewords.org.uk/word_count.asp. De resultaten worden in een tabel verwerkt.
3. Er wordt een wordcloud gemaakt, voor de visuele ondersteuning, met behulp van http://worditout.com/word-cloud/make-a-new-one. Deze is gebaseerd op de resultaten van de frequentie.

6.7.3 Experiment

De resultaten die uit het experiment komen zullen worden verwerkt in een Excel overzicht. Hierbij wordt per query aangegeven hoeveel seconden of minuten het heeft geduurd om een resultaat terug te krijgen. Ook wordt er aangegeven met welke omvang van de dataset is gewerkt en welke hardware configuratie is gebruikt. Alle queries uit de ad-hoc query set worden uitgevoerd op de verschillende configuraties.
Wanneer alle resultaten zijn verwerkt worden er grafieken gemaakt van de data die in Excel staan. Hierbij worden er grafieken gegenereerd om de data te visualiseren, zodat er een betere vergelijking kan worden uitgevoerd. De volgende grafieken zullen worden gegenereerd:

1. Gemiddelde snelheid van alle queries samen per pakket.
2. Overzicht van de snelheid per pakket per query.
3. Schaalbaarheid per pakket.

6.8 Resultaten onderzoek

De resultaten van het literatuuronderzoek kunnen worden bekeken in bijlage C '*Onderzoeksdocument*' hoofdstuk '*4. Resultaten literatuuronderzoek*'. De resultaten van de interviews worden behandeld in hoofdstuk *'11.1 Ad-hoc query set*'. De resultaten van het experiment worden behandeld vanaf hoofdstuk *'11.4 Resultaten gemiddelde snelheid'*.

7. Requirements

Nadat het onderzoeksontwerp was opgesteld en het vooronderzoek was uitgevoerd, is er verder gegaan met het opstellen van de requirements voor de pakketselectie. Dit is de eerste stap volgens de KPMG pakketselectie methode. In de KPMG methode wordt dit beschreven als '*opstellen van eisen & wensen'*, in dit project wordt het inventariseren requirements genoemd. Dit is voornamelijk een naamswijziging en heeft niet veel impact op het proces. Het voordeel is wel dat de requirements SMART geformuleerd kunnen worden. In dit hoofdstuk worden de stakeholders en de aanpak behandeld.

7.1 Stakeholders

De belangrijkste stakeholder tijdens dit project is de opdrachtgever. Hij bepaalt uiteindelijk met behulp van de requirements waar het pakket aan moet voldoen. Daarnaast is hij het enige aanspreekpunt voor het verifiëren en inventariseren van de requirements. Binnen Info Support zelf zijn er verder geen stakeholders.

Naast de opdrachtgever ben ik, de afstudeerder, ook een stakeholder. Met behulp van de requirements wordt duidelijk waar het pakket aan moet voldoen en kan ik de pakketselectie op een goede manier worden volbracht.

7.2 Aanpak

Voordat er is begonnen aan het inventariseren van de requirements voor de pakketselectie, heb ik eerst op internet gekeken naar informatie over het selecteren van een 'SQL-on-Hadoop' oplossing. De opdrachtgever had van te voren aangegeven dat er een opzet opgestuurd moest worden met requirements. Met deze requirements kon rekening worden gehouden tijdens de sessie. Op deze manier is de kans kleiner dat er tijdens de requirements inventarisatie sessie belangrijke requirements worden vergeten.

Tijdens het zoeken naar informatie op internet had ik een artikel gevonden met een aantal belangrijke requirements die kunnen worden gebruikt tijdens de pakketselectie [16]. Naast deze requirements heb ik ook met behulp van de 'Checklist selectiecriteria', van INDORA een aantal requirements opgesteld die van toepassing zijn voor de pakketselectie [17].

Met behulp van de gevonden informatie is er een lijst met voorlopige requirements opgesteld, die voorafgaand aan de inventarisatie sessie is opgestuurd naar de opdrachtgever. Daarnaast is er ook een topic lijst opgestuurd met onderwerpen die nog niet concreet genoeg waren om als requirement te formuleren, maar wel belangrijk waren om niet te vergeten. Een voorbeeld van een voorlopige requirement was: '*JOINS moeten worden ondersteund*'. Een voorbeeld voor een onderwerp op de topiclijst was: '*Configuratie van de pakketten'*.

7.2.1 Eerste requirements inventarisatie sessie

Het doel van de eerste requirements inventarisatie sessie was om een lijst met requirements op te stellen voor de pakketselectie. De sessie bestond uit twee delen, het doorlopen van de opzet van de requirements en het toevoegen van ontbrekende requirements. De input voor het eerste gedeelte van de sessie was de opzet van de

requirements die al van te voren was opgestuurd. Hierbij is door de opdrachtgever aangegeven welke requirements van deze lijst voor hem relevant waren tijdens de pakketselectie en welke niet.

Nadat de opzet van de requirements was doorgelopen is er verder gegaan met het tweede gedeelte van de sessie: het toevoegen van ontbrekende requirements. Hierbij is vooral dieper op ingegaan welke requirements nog ontbraken. Door middel van vragen te stellen aan de hand van de topic lijst, die vooraf was opgestuurd, zijn er nog een aantal requirements toegevoegd. Zo was het onderwerp '*Configuratie van de pakketten*' dat op de topic lijst stond uiteindelijk uitgewerkt in twee verschillende requirements, namelijk de mogelijkheid voor cloud en on-premise configuratie.

Nadat de requirements sessie was afgelopen zijn alle requirements tot dan toe Specifiek, Meetbaar, Acceptabel, Realiseerbaar en Tijdsgebonden (SMART) geformuleerd. Op deze manier zijn de requirements ook beter bruikbaar tijdens de pakketselectie, doordat er concreet kan worden gekeken naar wat er precies nodig is. Hierbij is het aspect 'Meetbaar' het belangrijkste, omdat er uiteindelijk scores worden gegeven tijdens de pakketselectie. Deze lijst met SMART requirements zijn opgestuurd naar de opdrachtgever, voor extra controle.

De voorlopige requirement '*JOINS moeten worden ondersteund*' was herschreven op de volgende manier:

ID	Omschrijving
PKSR17	Het pakket moet LEFT joins ondersteunen.
PKSR18	Het pakket moet RIGHT joins ondersteunen.
PKSR19	Het pakket moet INNER joins ondersteunen.
PKSR20	Het pakket moet FULL OUTER joins ondersteunen.

Tabel [5]: Overzicht JOIN requirements.

Het onderwerp '*Configuratie van de pakketten*' dat bijvoorbeeld op de topic lijst stond, was herschreven op de volgende manier:

ID	Omschrijving
PKSR21	Het pakket mag in de cloud configureerbaar zijn.
PKSR22	Het pakket mag on-premise configureerbaar zijn.

Tabel [6]: Overzicht configuratie requirements.

7.2.2 Tweede requirements sessie

Omdat het vaak niet lukt om alle requirements in één keer te inventariseren is er een tweede requirements sessie gehouden. Vaak komen er nog een aantal requirements bij waar in de eerste sessie niet aan was gedacht, bijvoorbeeld door nieuwe ingevingen. Het doel van de tweede requirements sessie bestond uit drie onderdelen:

1. Kleine aanpassingen maken aan de requirements uit de eerste sessie.
2. Toevoegen van ontbrekende requirements.
3. Het prioriteren van alle requirements.

Als eerste werd er tijdens de tweede sessie behandeld of de opdrachtgever akkoord ging met de requirements uit de eerste sessie. Hierbij waren een paar aanpassingen aangedragen door de opdrachtgever. Vervolgens is er gevraagd of er nog requirements ontbraken. Hierbij kwamen nog twee extra requirements naar voren. Deze zijn ter plaatse SMART geformuleerd, zodat er verder kon worden gegaan met de prioritering.

De prioritering is gedaan aan de hand van de MoSCoW methode. Deze methode wordt veel gebruikt binnen software ontwikkelprojecten, maar is ook toepasbaar voor projecten met requirements die buiten software ontwikkelprojecten vallen. Daarnaast is MoSCoW bekend als methode om requirements te prioriteren.

De afkorting MoSCoW staat voor: Must have, Should have, Could have en Will not have.

Tijdens de prioritering zijn alle requirements doorgelopen. Hierbij zijn eerst de 'Must have' requirements gedefinieerd. Omdat het er al snel op leek dat bijna alle requirements een 'Must have' werden is er kritischer gekeken naar de toewijzing van een 'Must have'. Dit is vooral gedaan door de volgende twee vragen te stellen:

1. "Welke reden is er om deze requirement als 'Must have' te prioriteren?"
2. "Wat zou het gevolg zijn voor de resultaten van de pakketselectie als deze requirement geen 'Must have' zou zijn?"

Dankzij deze vragen werd er beter nagedacht over de toekenning van de verschillende prioriteiten. Zolang er een goed argument was vanuit de opdrachtgever om een requirement een bepaalde prioritering te geven is deze prioritering toegekend. De bovenstaande twee vragen zijn vervolgens ook gesteld voor de requirements die een 'Should have' en 'Could have' prioritering kregen.
Nadat de prioritering was afgerond is alles nog voor een extra goedkeuring naar de opdrachtgever gestuurd. Nadat die akkoord was, is deze fase afgerond en kon er verder worden gegaan met de volgende fase: de pakketselectie.

De requirements voor de ondersteuning van JOINS hebben de volgende prioritering gekregen:

ID	Beschrijving	Prioritering
PKSR17	Het pakket moet LEFT joins ondersteunen.	M
PKSR18	Het pakket moet RIGHT joins ondersteunen.	M
PKSR19	Het pakket moet INNER joins ondersteunen.	M
PKSR20	Het pakket moet FULL OUTER joins ondersteunen.	M

Tabel [7]: Overzicht JOIN requirements.

De requirements voor de configuratie hebben de volgende prioritering gekregen:

ID	Beschrijving	Prioritering
PKSR21	Het pakket mag in de cloud configureerbaar zijn.	S
PKSR22	Het pakket mag on-premise configureerbaar zijn.	S

Tabel [8]: Overzicht configuratie requirements.

Een compleet overzicht van alle requirements, inclusief prioritering, is te vinden in bijlage G *'Requirementsrapport'*.
De requirements worden ook in hoofdstuk *'8.3 Selectiecriteria'* behandeld.

8. Longlist

Nadat de requirements voor de pakketselectie waren geïnventariseerd, kon er een begin worden gemaakt aan het tweede onderdeel van de pakketselectie, het opstellen van de longlist. In dit hoofdstuk wordt beschreven op welke manier de longlist is aangepakt, welke selectiecriteria zijn gebruikt en wat het resultaat van de longlist was.

8.1 Aanpak

Het doel van de longlist was om de pakketten op verschillende criteria te vergelijken met elkaar. Deze is aan de hand van de KPMG methode opgesteld.

Voor de longlist was er in Excel een selectiematrix opgesteld, met daarin de verschillende pakketten en criteria. Meer informatie over de totstandkoming van de criteria en de weging is te lezen in hoofdstuk '*8.2 Selectiecriteria*'. Op deze manier kan er op een eenvoudige manier worden gekeken hoe de pakketten scoren op de verschillende criteria en welke onderlinge verschillen er per pakket zijn.

Vervolgens zijn de verschillende criteria per pakket opgezocht in de officiële documentatie van de pakketten. Criteria die niet direct via de documentatie konden worden beantwoord zijn achterhaald met behulp van Google. Meer informatie over de resultaten is te lezen in hoofdstuk '*8.3 selectiematrix*'.

8.1.1 Marktonderzoek

Voordat er begonnen was met het opstellen van de longlist, was er eerst gekeken welke 'SQL-on-Hadoop' pakketten allemaal beschikbaar zijn. Hier werd al snel duidelijk dat dit aantal behoorlijk groot was. Er zijn volgens de '*Hadoop ecosystem table*' twaalf open-source pakketten [18]. Daarnaast zijn er nog zeven bedrijven die commerciële 'SQL-on-Hadoop' pakketten aanbieden [19]. In tabel [9] staan al deze pakketten beschreven.

Open-source:	Commercieel:
Apache Hive	HP Vertica
Apache HCatalog	Pivotal HAWQ
Trafodion	Oracle Big Data SQL
Apache Drill	IBM Big SQL 3.0
Cloudera Impala	Jethro Data
Facebook Presto	Hadapt
Datasalt Splout SQL	Actian Vortex
Apache Tajo	
Apache Phoenix	
Apache MRQL	
Apache Kylin	
Apache Spark SQL	

Tabel [9]: Overzicht SQL-on-Hadoop pakketten.

Zoals in tabel [9] te zien is, zijn er in totaal negentien 'SQL-on-Hadoop' pakketten beschikbaar. Dit aantal is misschien al toegenomen sinds het schrijven van de pakketselectie. Dit komt doordat vier jaar geleden alleen

Apache Hive beschikbaar was [20] en er zijn nu negentien beschikbare pakketten van commerciële bedrijven en de open-source community.

Omdat negentien pakketten een te grote hoeveelheid is om tijdens de longlist te onderzoeken is wordt het aantal pakketten teruggebracht. Dit wordt ook aangegeven door meerdere blogs en artikelen over pakketselecties [21, 22, 23]. Deze artikelen beschrijven dat er maximaal acht tot tien pakketten op de longlist mogen [21, 22, 23]. In overleg met de opdrachtgever is ervoor gekozen om het grote aanbod te filteren met behulp van de volgende drie requirements die voortkwamen uit de requirements sessie:

1. **Het pakket moet een SQL of SQL-like taal ondersteunen**. Op deze manier kunnen gebruikers die SQL beheersen direct aan de slag met rapportages en ad-hoc queries uitvoeren op de data die in Hadoop is opgeslagen. Voor SQL-like talen is er minimale training nodig om dit te kunnen gebruiken, vanwege de vele overeenkomsten met SQL. Pakketten waarbij scripting talen worden gebruikt, zoals Apache Pig vallen op deze manier af.
2. **Het pakket moet direct bovenop Hadoop worden gebruikt**. Pakketten waar een tussenlaag voor moet worden geïnstalleerd, zoals pakketten die HBase bijvoorbeeld nodig hebben om te functioneren, worden niet meegenomen in de pakketselectie.
3. **Het pakket moet open-source zijn**. In tegenstelling tot de bovenstaande twee requirements is deze requirement gedeeltelijk doorgevoerd. De opdrachtgever wilde namelijk wel weten of er een groot verschil zat tussen de commerciële pakketten en open-source pakketten. De opdrachtgever gaf aan dat er een grote voorkeur is voor open-source. Maar wanneer het blijkt dat een commercieel pakket het beste scoort op de longlist, zou dit pakket ondanks dat het een commercieel pakket is, toch kunnen worden meegenomen naar de shortlist. Daarom is er een beperkt aantal commerciële pakketten meegenomen op de longlist.

De commerciële pakketten die op de longlist terecht zijn gekomen, waren vooral gebaseerd op andere artikelen waarin ‘SQL-on-Hadoop’ pakketten werden vergeleken.
De open source pakketten die niet zijn meegenomen op de longlist, zijn afgevallen doordat ze niet voldeden aan de eerste twee requirements. De belangrijkste filter was dat het pakket niet direct boven Hadoop kon worden gebruikt. Door de requirements die hierboven beschreven staan is er een longlist ontstaan met negen pakketten. Deze pakketten worden beschreven in de volgende paragraaf.

8.2 Selectiecriteria

Een aantal van de requirements zijn tijdens het opstellen van de selectiecriteria voor de longlist samengevoegd. Hier is voor gekozen omdat ze onder hetzelfde onderwerp vallen. Een voorbeeld hiervan is de ondersteuning voor JOIN statements. Hieronder is een overzicht beschreven met alle selectiecriteria die tijdens de longlist worden gebruikt. Deze zijn gekoppeld aan de requirement(s), zodat het duidelijk is waarop deze selectiecriteria zijn gebaseerd.

Selectiecriteria:	Requirement(s):
Open source	-PKSR1: Het pakket moet open source zijn.
ANSI SQL	- PKSR2: Het pakket moet ondersteuning bieden aan ANSI SQL of een SQL-like taal.
Ondersteuning DDL	- PKSR4: Het pakket moet het Data Definition Language commando "CREATE" ondersteunen. - PKSR5: Het pakket moet het Data Definition Language commando "ALTER" ondersteunen. - PKSR6: Het pakket moet het Data Definition Language commando "DROP" ondersteunen.
Ondersteuning DML	- PKSR7: Het pakket moet het Data Manipulation Language commando "SELECT" ondersteunen. - PKSR8: Het pakket moet het Data Manipulation Language commando "INSERT" ondersteunen. - PKSR9: Het pakket moet het Data Manipulation Language commando "UPDATE" ondersteunen. - PKSR10: Het pakket moet het Data Manipulation Language commando "DELETE" ondersteunen.
Ondersteuning aggregates	- PKSR11: Het pakket moet de aggregate functie "AVG" ondersteunen. - PKSR12: Het pakket moet de aggregate functie "COUNT" ondersteunen. - PKSR13: Het pakket moet de aggregate functie "MAX" ondersteunen. - PKSR14: Het pakket moet de aggregate functie "MIN" ondersteunen. - PKSR15: Het pakket moet de aggregate functie "SUM" ondersteunen.
Ondersteuning subqueries	- PKSR16: Het pakket moet subqueries ondersteunen.
Ondersteuning joins	- PKSR17: Het pakket moet LEFT joins ondersteunen. - PKSR18: Het pakket moet RIGHT joins ondersteunen. - PKSR19: Het pakket moet INNER joins ondersteunen. - PKSR20: Het pakket moet FULL OUTER joins ondersteunen.
Cloud configuratie	- PKSR21: Het pakket mag in de cloud configureerbaar zijn.
On-premise configuratie	- PKSR22: Het pakket mag on-premise configureerbaar zijn.
Beschikbare documentatie	- PKSR23: Het pakket moet een gebruikershandleiding hebben. - PKSR24: Het pakket moet overzicht met ondersteunde datatypes hebben.

	- PKSR25: Het pakket moet een installatiehandleiding hebben.
Security	- PKSR26: Het pakket moet authentication ondersteunen. - PKSR27: Het pakket moet authorization ondersteunen. - PKSR28: Het pakket moet data encryptie ondersteunen. - PKSR29: Het pakket moet auditing ondersteunen.
Data federation	- PKSR30: Het pakket moet data federation ondersteunen voor drie of meer data sources.
Actieve community	- PKSR31: Het open source pakket mag een top-level Apache project zijn. - PKSR32: Het open source pakket mag twintig of meer contributors op Github hebben. - PKSR33: Het open source pakket mag honderd of meer stars op Github hebben. - PKSR34: Het commerciële pakket mag een forum hebben met support - PKSR 35: Het commerciële pakket mag een support helpdesk hebben.
Stable release	- PKSR36: Het pakket mag een 1.0.0 release of hoger zijn.
Ondersteuning file formats	- PKSR37: Het pakket mag twee of meer file formats ondersteunen.
Meerdere gebruikers	- PKSR38: Het pakket mag queries van verschillende gebruikers tegelijkertijd kunnen ondersteunen.
Hadoop distributie	- PKSR39: Het pakket mag bij een open source Hadoop distributie zitten.

Tabel [10]: Selectiecriteria met bijbehorende requirements.

8.3 Selectiematrix

Om te kunnen bepalen welke pakketten op de longlist het beste zijn, is er een punten verdeling toegekend aan de verschillende selectiecriteria. Omdat sommige pakketten bepaalde selectiecriteria niet helemaal ondersteunen, maar gedeeltelijk, is in overleg met de opdrachtgever ervoor gekozen om een onderscheid te maken in de puntenverdeling. Wanneer een pakket een bepaald selectiecriteria gedeeltelijk ondersteunt zal dit pakket niet het volledige aantal punten krijgen, maar een gedeelte van de punten. In tabel [11] is beschreven hoe de punten verdeling eruitziet en wanneer deze wordt toegekend.

Beschrijving	Puntenverdeling
Het pakket ondersteunt de selectiecriteria goed.	3 Punten
Het pakket ondersteunt de selectiecriteria voldoende.	2 Punten
Het pakket ondersteunt de selectiecriteria onvoldoende.	1 Punt
Het pakket ondersteunt de selectiecriteria niet.	0 Punten

Tabel [11]: Punten verdeling voor de selectiecriteria.

Volgens de KPMG pakketselectie methode moeten er zogenaamde ‘KeyCriteria’ worden gedefinieerd voor de Longlist. Omdat tijdens de prioritering van de requirements een aantal requirements een ‘Must have’ hebben gekregen kunnen deze ‘Must have’ requirements hiervoor gebruikt worden. Deze requirements zijn namelijk belangrijker dan de andere requirements. In overleg met de opdrachtgever is er afgesproken dat de

puntenverdeling in tabel [11] wordt verdubbeld voor KeyCriteria. Dit komt dan neer op 6 punten voor goed, 4 punten voor voldoende, 2 punten voor onvoldoende, 0 punten voor niet ondersteund. Wanneer het puntenaantal zou worden verdrievoudigd, hebben de minder belangrijke requirements weinig invloed meer op de eindscore, vandaar dat er voor een verdubbeling is gekozen. De KeyCriteria staan hieronder beschreven:

1. Open source.
2. ANSI SQL.
3. Ondersteuning DDL.
4. Ondersteuning DML.
5. Ondersteuning aggregates.
6. Ondersteuning subqueries.
7. Ondersteuning joins.
8. Beschikbare documentatie.
9. Security.
10. Data federation.

In de selectiematrix zijn deze KeyCriteria dikgedrukt. De selectiematrix is te zien in afbeelding [13]. Verder wordt in bijlage H '*Pakketselectie'* hoofdstuk *'3.2 Selectiematrix*' uitgebreid beschreven per selectiecriteria hoe de puntenverdeling tot stand is gekomen. De beschrijving per puntenverdeling is gebaseerd op de puntenverdeling in tabel [11].

Selectiecriteria	Apache Drill	Stinger Initiative	Cloudera Impala	Facebook Presto	Apache Tajo	Apache Kylin	Oracle Big Data SQL	Pivotal HAWQ	Jethro Data
Open source	6	6	6	6	6	6	0	0	0
ANSI SQL	6	4	4	6	6	6	6	6	6
Ondersteuning DDL	4	6	6	6	4	?	6	6	6
Ondersteuning DML	2	6	2	2	2	?	2	6	2
Ondersteuning aggregates	6	6	6	6	2	?	6	6	6
Ondersteuning subqueries	6	6	6	6	6	?	6	6	6
Ondersteuning joins	6	6	6	6	6	?	6	6	6
Cloud configuratie	3	3	3	3	3	3	3	3	3
On-premise configuratie	3	3	3	3	3	3	3	3	3
Beschikbare documentatie	6	6	6	6	6	2	6	6	6
Security	6	6	6	2	2	2	6	2	2
Data federation	6	2	4	6	6	6	6	6	2
Actieve community	3	3	3	3	2	2	3	3	3
Stable release	2	3	3	2	2	2	3	3	1
Ondersteuning file formats	3	3	3	2	3	?	3	2	3
Meerdere gebruikers	3	3	3	3	3	?	3	3	3
Hadoop distributie	3	3	3	3	3	0	3	0	0
Totaal	74	75	73	71	65	32	68	67	58

Afbeelding [13]: Selectiecriteria longlist.

In de selectiematrix is te zien dat voor Apache Kylin een aantal selectiecriteria een"?" hebben gekregen. Voor deze selectiecriteria was het niet mogelijk om uit te zoeken of deze werden ondersteund in Apache Kylin. Dit kwam door zeer geringe documentatie.

8.4 Resultaat

Aan de hand van de totale score per pakket in de selectiematrix kan er worden bepaald hoe goed de pakketten hebben gescoord op de verschillende selectiecriteria. Geen enkel pakket heeft het maximale aantal punten van 81 gehaald. Daarom is er per pakket uitgebreider beschreven welke tekortkomingen elk pakket heeft. Dit is voor de selectiecriteria gedaan waar niet het volledige aantal punten is toegekend. Dit staat beschreven in bijlage H '*Pakketselectie'* hoofdstuk *'3.3 Tekortkomingen per pakket'*. Dankzij deze informatie en de totale score uit de selectiematrix kan er een weloverwogen keuze worden gemaakt door de opdrachtgever.

Zoals te zien in afbeelding [13] zitten de scores van de vier beste open-source pakketten redelijk dicht bij elkaar. Wanneer er wordt gekeken naar de commerciële pakketten zijn er twee pakketten die goed hebben gescoord op alle selectiecriteria, behalve op de open-source selectiecriteria. Wanneer de open-source selectiecriteria niet zou worden meegenomen, zouden deze pakketten even hoog scoren als Apache Drill en Cloudera Impala. De reden dat de open-source criteria toch behouden blijft, is vanwege de grote voorkeur van de opdrachtgever voor open-source pakketten.

Vervolgens is er in overleg met de opdrachtgever voor gekozen om de drie hoogst scorende pakketten te behandelen in de shortlist. De volgende drie pakketten zijn geselecteerd die worden behandeld op de shortlist:

1. Stinger Initiative
2. Apache Drill
3. Cloudera Impala

Stinger Initiative is het project dat wordt geleid door Hortonworks om Apache Hive te versnellen.

Wel is het belangrijk om er rekening mee te houden dat de resultaten van de longlist gebaseerd zijn op informatie die is opgezocht tussen 09-03-2015 en 13-03-2015. Omdat de meeste pakketten nog verder worden ontwikkeld kan het zijn dat wanneer de pakketselectie op een later moment zou worden uitgevoerd er andere resultaten uit kunnen komen. Tijdens de longlist is de huidige versie die toen beschikbaar was gebruikt om de verschillende selectiecriteria te onderzoeken voor de pakketten.

Daarnaast is het ook belangrijk om er rekening mee te houden dat de pakketten die niet op de shortlist zijn gekomen wellicht sneller kunnen zijn met het uitvoeren van queries, omdat dit pas tijdens de shortlist wordt behandeld.

9. Gegevensconversie

Nadat de longlist was afgerond en de drie beste pakketten bekend waren, kon er verder worden gegaan met de shortlist en het experiment. Om het experiment uit te kunnen voeren is er een dataset nodig. Voordat deze dataset kan worden gebruikt moet er eerst een gegevensconversie worden uitgevoerd. In dit hoofdstuk wordt deze gegevensconversie behandeld. Het resultaat van deze gegevens conversie is een relationele dataset die beschikbaar is in HDFS. Met behulp van deze dataset kunnen de ad-hoc queries worden uitgevoerd tijdens het experiment en de shortlist, om zo te bepalen welk pakket het meest geschikt is voor SQL-on-Hadoop.

9.1 Dataset

Omdat er gebruik wordt gemaakt van een bestaande dataset, bevat deze naast tabellen met data, ook verschillende views, triggers, stored procedures en user-defined functions. Voor de scope van het onderzoek wordt alleen gefocust op de data en worden de views, triggers, stored proces en user-defined functions niet meegenomen. Daarnaast zou het ook alleen maar mogelijk zijn om Views in te laden in Apache Hive [24].

De dataset die wordt gebruikt bij de conversie is van AdventureWorksDW2012. Dit is een Datawarehouse dataset en bestaat uit zeven fact tabellen en zestien dimensie tabellen. Naast de Datawarehouse variant is er ook de traditionele OLTP AdventureWorks2012 database. De reden dat er voor de Datawarehouse variant is gekozen is in overleg gedaan met de opdrachtgever. De opdrachtgever gaf aan dat de opdracht namelijk gericht was op Business Intelligence en Datawarehouse omgevingen. Hierbij werd de AdventureWorksDW2012 dataset als voorbeeld aangedragen. De structuur van deze dataset is gedenormaliseerd en wordt ook wel een ‘snowflake’ schema, of ‘star’ schema genoemd. Deze structuur bevat veel minder tabellen die aan elkaar moeten worden gekoppeld vergeleken met een genormaliseerde dataset. Hierdoor zijn er minder JOIN statements nodig wanneer een query wordt uitgevoerd. Het gevolg hiervan zou kunnen zijn dat er andere resultaten uit het experiment komen dan wanneer er een genormaliseerde dataset wordt gebruikt.

Er is bewust voor gekozen om niet de volledige dataset te gebruiken. In plaats daarvan worden de twee grootste fact tabellen gebruikt, met de negen bijbehorende dimensie tabellen. Op deze manier hoeft de dataset maar voor twee fact tabellen te worden uitgebreid. Er hoeven dan geen miljoenen rijen gegenereerd te worden voor elke fact tabel. Hierdoor is er minder ruimte nodig om de dataset op te slaan. Daarnaast zijn deze twee fact tabellen met de meeste dimensie tabellen verbonden. Hierdoor zijn er veel filter criteria mogelijk voor de ad-hoc queries. In afbeelding [14] is in een database diagram te zien hoe de fact tabellen zijn verbonden met de dimensies.

De volgende tabellen komen voor in de AdventureWorksDW2012 dataset waar de ad-hoc queries op worden uitgevoerd:

1. DimCustomer
2. DimDate
3. DimEmployee
4. DimGeography
5. DimProduct
6. DimProductCategory
7. DimProductSubcategory
8. DimReseller
9. DimSalesTerritory

10. FactInternetSales
11. FactResellerSales

Voor een overzicht met alle kolommen per tabel en bijbehorende data types, kan worden gekeken in bijlage I '*Datadictionary*'.

9.1.1 Aanpak

De dataset is vervolgens uitgebreid met de tool: '*SQL Data Generator*'. Deze tool heeft uitgebreide mogelijkheden om te worden geconfigureerd, waaronder minimale of maximale waarden voor kolommen, automatische relatie herkenning tussen tabellen en regular expressions. Verder was voor deze tool een dertig dagen trial beschikbaar, waarmee een onbeperkt aantal rijen kon worden gegenereerd. Veel van de andere data generatie tools die een trial hadden konden maar een beperkt aantal rijen genereren. Daarnaast had '*SQL Data Generator*' de meeste configuratie mogelijkheden vergeleken met gratis data generatie tools. Vanwege deze redenen is deze tool gebruikt. Dankzij deze uitgebreide configuratie kan er data worden gegenereerd die redelijk overeenkomt met de bestaande data.

De oorspronkelijke grootte van beide fact tabellen was ongeveer 60.000 rijen. Deze zijn uitgebreid naar een dataset van tien miljoen rijen per fact tabel en een dataset van honderd miljoen rijen per fact tabel. Daarnaast hebben de volgende dimensie tabellen één miljoen rijen erbij gekregen:

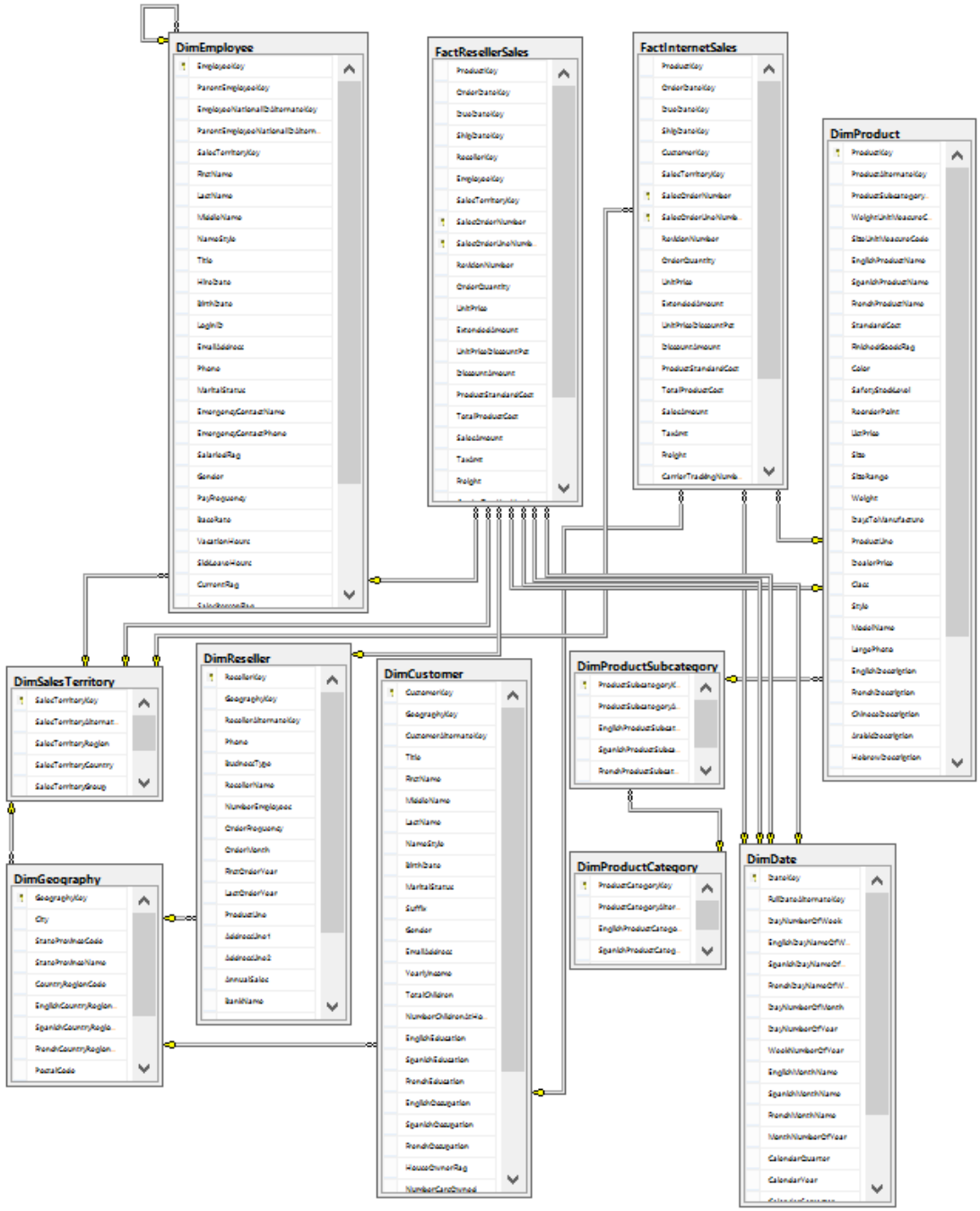
1. DimCustomer
2. DimEmployee

Op deze manier zijn er ook dimensie tabellen met een miljoen rijen en is dit niet alleen voor de fact tabellen gedaan. De reden dat er geen dataset van meer dan honderd miljoen rijen is gegenereerd, is vanwege de tijd die het genereren in beslag nam. Tijdens het genereren van de data zijn er elke keer tien miljoen rijen toegevoegd. Vervolgens werden de indexen van de tabellen opnieuw opgebouwd. Naarmate er meer data werd gegenereerd, ging het proces steeds langzamer. Soms is het proces ook handmatig gestopt, omdat de computer werd afgesloten. Vandaar dat er niet precies honderd miljoen rijen zijn gegenereerd.

Bij het genereren van de dataset van tien miljoen rijen zijn er tien miljoen rijen toegevoegd aan de bestaande dataset. Vandaar dat deze ook niet precies tien miljoen rijen bevatten. Het exacte aantal rijen is te vinden in hoofdstuk '*10.2 Resultaat*'.

Omdat het genereren van de data lang duurde, is in overleg met de opdrachtgever besloten dat voor het experiment een dataset van tien miljoen en honderd miljoen rijen per fact tabel genoeg is.

9.1.2 Database diagram



Afbeelding [14]: Database diagram AdventureWorksDW2012.

9.2 Sqoop

Om de relationele structuur van de dataset te converteren naar HDFS structuur moet er een conversie worden uitgevoerd. Deze conversie kan handmatig worden uitgevoerd, of met behulp van een tool. Om de dataset handmatig in te laden in HDFS zou er een tool gebouwd moeten worden die een connectie met de Amazon EMR cluster opzet en een connectie met de MS SQL Server database instantie opzet. Na het opzetten van de connectie zou de data moeten worden geconverteerd en in HDFS worden gezet. Het zelf bouwen van zo'n soort tool neemt veel tijd in beslag. Daarom is ervoor gekozen om een bestaande tool te selecteren.

Om data in te laden in HDFS zijn er twee tools beschikbaar:

1. Apache Sqoop
2. Apache Flume

Apache Sqoop focust zich op het importeren van data uit een relationele data uit onder andere relationele databases naar HDFS. Hierbij wordt de relationele structuur van de data behouden [25]. Apache Flume focust zich op het importeren van log data uit verschillende bronnen naar HDFS [26]. Het is dus niet specifiek ontwikkeld om relationele data te converteren naar HDFS. Daarom is ervoor gekozen om gebruik te maken van Apache Sqoop.

9.2.1 Installatie & configuratie SQOOP

In tegenstelling tot de pakketten op Amazon EMR, was SQOOP niet 'out-of-the-box' beschikbaar. Voor de installatie is SQOOP 1.4.5 gebruikt. Omdat SQOOP een verbinding met een Microsoft SQL Server 2014 instantie op Amazon maakt, is de Microsoft JDBC Driver 4.1 toegevoegd aan SQOOP. Vervolgens is er getest of er een connectie kon worden opgezet met de Microsoft SQL Server database. Toen deze connectie succesvol was is er verder gegaan met het uitvoeren van de gegevensconversie.

9.2.2 Mapping

Niet elk data type wordt ondersteund in Apache Hive of Apache Sqoop. Daarom is in tabel [12] een overzicht te zien welke data types dit zijn.

Datatypes	Microsoft SQL Server 2014	Apache Sqoop	Apache Hive
TINYINT	V	V	V
SMALLINT	V	V	V
INT	V	V	V
BIGINT	V	V	V
FLOAT	V	V	V
DOUBLE	V	V	V
DECIMAL	V	V	V
TIMESTAMP	V	V	V
DATE	V	V	V
STRING	V	V	V
VARCHAR	V	V	V
CHAR	V	V	V
BOOLEAN	X	V	V
BINARY	V	V	V
BIT	V	X	X
MONEY	V	V	X
VARBINARY	V	X	V

Tabel [12]: Overzicht ondersteunde data types per pakket.

In tabel [12] is te zien dat de data types BIT, MONEY en VARBINARY niet overal worden ondersteund. De niet ondersteunde data types zijn met een rood kruis gemarkeerd. Daarom is ervoor gekozen om deze tijdens de conversie aan te passen naar de volgende data types:

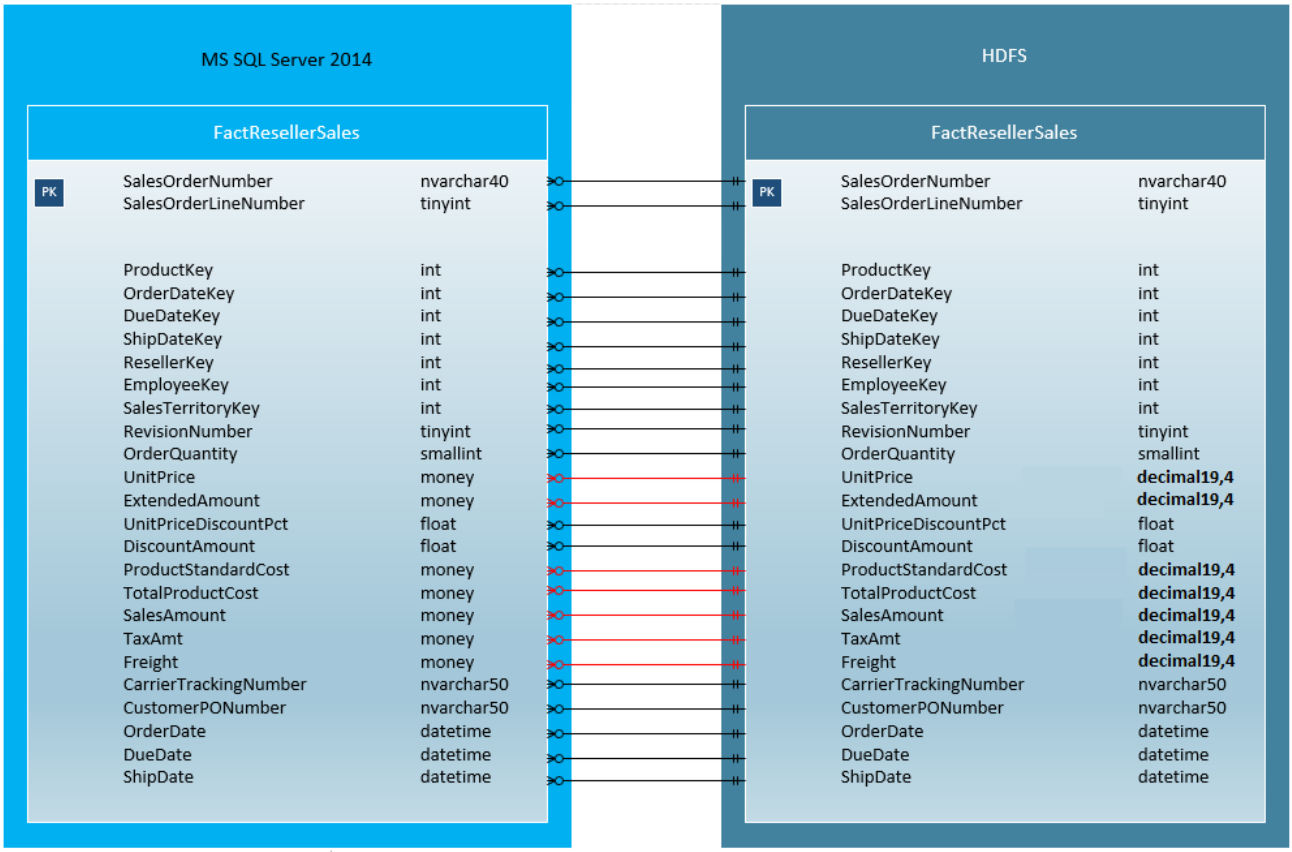
1. BIT -> BOOLEAN
2. MONEY -> DECIMAL(19,4)
3. VARBINARY -> VARCHAR(MAX)

De informatie welke data types worden ondersteund in Apache Hive is met behulp van de 'Language Manual' achterhaald [27]. De ondersteuning voor VARBINARY in Apache Sqoop wordt in de volgende paragraaf meer over verteld.

Omdat de conversie de structuur van bepaalde data aanpast, wat leidt tot data die groter wordt qua opslag, is er gekeken wat voor invloed dit precies heeft. Hieronder worden deze invloeden beschreven:

1. BOOLEAN: geeft aan of iets TRUE of FALSE is, door middel van een TINYINT [28]. Dit neemt qua opslag ruimte meer in dan alleen een BIT [29].
2. DECIMAL(19,4): slaat data accurater op dan MONEY data type, maar gebruikt wel meer opslag ruimte hiervoor [30].
3. VARCHAR(MAX): heeft een grotere impact op de performance en de opslag vergeleken met VARBINARY [31].

Vervolgens zijn met behulp van de datadictionaries de mappings gemaakt voor de conversie. Deze mappings zijn terug te vinden in bijlage L '*Technisch Ontwerp'* hoofdstuk '*5.2 Mapping*'. In afbeelding [15] is een voorbeeld te zien van de mapping FactResellerSales.

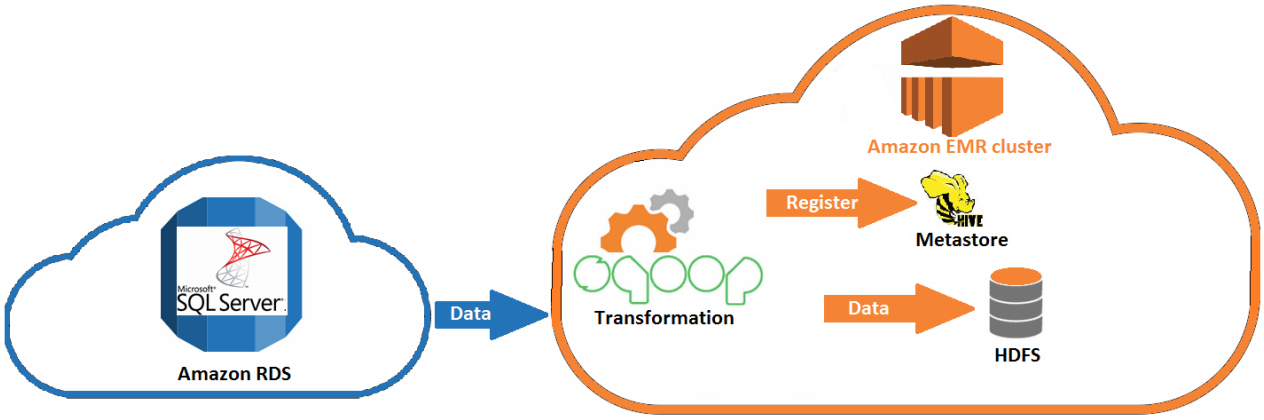


Afbeelding [15]: Voorbeeld mapping.

Attributen die tijdens de conversie worden aangepast zijn dikgedrukt en hebben een rode lijn.

9.2.3 Uitvoer conversie

Na het maken van de mappings is de conversie uitgevoerd. Hiervoor is de lokaal gegenereerde dataset naar een Amazon RDS (Relational Database Service) instantie geüpload. Vervolgens is de conversie uitgevoerd. In afbeelding [16] is hier een overzicht van gemaakt.



Afbeelding [16]: Overzicht conversie.

Zoals in afbeelding [16] te zien is, vindt de conversie compleet in de cloud plaats. De blauwe cloud stelt de Amazon RDS instantie voor. De oranje cloud stelt de Amazon EMR cluster voor. Apache Sqoop draait op een instantie van een Amazon EMR cluster en importeert de data vanuit een Microsoft SQL Server database. Een voorwaarde voor de uitvoer van deze conversie is dat Apache Hive, inclusief de Hive Metastore al geïnstalleerd moeten zijn op de Hadoop cluster. De reden hiervoor wordt later in deze paragraaf behandeld.

Apache Sqoop verandert tijdens de conversie de data types BIT en MONEY automatisch naar data types die in Apache Hive worden ondersteund. Dit werd tijdens de uitvoer van de conversie duidelijk. De uitzondering hierop is VARBINARY. Deze moet handmatig met een query worden verandert. Vanwege deze reden is de standaard ‘import’ functie van Apache Sqoop niet gebruikt, omdat er drie tabellen zijn met een VARBINARY kolom.
Wanneer er wel gebruik gemaakt werd van de standaard ‘import’ en er waren kolommen met VARBINARY data type aanwezig, ontstaan er foutmeldingen. Dit komt doordat Sqoop dit data type niet kan converteren, omdat deze data type nog niet wordt ondersteund [32].

Tijdens de conversie is ook de ‘—hive-import’ parameter meegegeven. Op deze manier werden de tabellen tijdens de conversie automatisch geregistreerd in de Hive Metastore. Dankzij de Hive Metastore kunnen direct queries worden uitgevoerd op de tabellen die in HDFS staan opgeslagen, omdat de metadata namelijk is opgeslagen op één plek. Cloudera Impala en Apache Drill maken om dezelfde reden gebruik van de Hive Metastore.

Als de tabellen niet zouden worden geregistreerd bij de Hive Metastore, zouden de tabellen handmatig moeten worden geregistreerd. Hiervoor worden dan zogenaamde ‘external tables’voor aangemaakt. Het maken van deze tabellen heeft veel weg van het maken van tabellen in SQL. Er moet worden aangegeven welke attributen en datatypen de tabel bevat en wat de naam is van de tabel. Het gebruik van external tables geeft extra mogelijkheden, qua configuratie en gebruik van namen. Alleen tijdens deze conversie is dat niet nodig.

10. Testen

Nadat de gegevensconversie is uitgevoerd moest deze worden getest. In dit hoofdstuk wordt de gegevensconversie test behandeld. Hierbij wordt ingegaan op welke manier de testen zijn aangepakt en wat voor resultaten de testen hebben opgeleverd.

10.1 Aanpak

Omdat het teveel tijd in beslag neemt om alle rijen handmatig te controleren die zijn overgezet tijdens de conversie, is ervoor gekozen om het testen te automatiseren. Dit is gedaan volgens TMAP Next. TMAP Next heeft in een document beschreven, genaamd "*Overzicht toegepaste testvormen*" [33], welke kwaliteitsattributen van toepassing zijn voor een gegevensconversie test. Hieronder wordt beschreven welke kwaliteitsattributen en testen er zijn opgesteld en uitgevoerd om deze kwaliteitsattributen aan te tonen.

Kwaliteitsattributen: juistheid, volledigheid.

Om de kwaliteitsattributen aan te tonen zullen de volgende testen worden uitgevoerd:

1. Tellen van het aantal rijen per tabel.
2. Tellen van de numerieke waarden per kolom.
3. Tellen van de lengte van niet-numerieke waarden per kolom

Deze testen zijn in overleg met de opdrachtgever en technisch begeleider opgesteld en worden door de opdrachtgever als voldoende beschouwd om aan te tonen dat de gegevensconversie goed is uitgevoerd. De bovenstaande testen zijn gebaseerd op ervaring, van zowel mijzelf, de technisch begeleider als de opdrachtgever. Het was echter niet gelukt om een methode te vinden waarin concreet wordt beschreven welke testen er uitgevoerd moeten worden voor een gegevensconversie. Buiten TMAP Next waren er geen methoden gevonden om een gegevensconversie te testen.

Deze testen worden voor de volgende twee datasets uitgevoerd:

1. Eerste dataset; twee fact tabellen met ongeveer tien miljoen rijen en één miljoen rijen voor DimCustomer en DimEmployee.
2. Tweede dataset; twee fact tabellen met ongeveer honderd miljoen rijen en één miljoen rijen voor DimCustomer en DimEmployee.

Vanuit de opdrachtgever was het een wens om de gegevensconversie tests voor elke dataset opnieuw uit te voeren. De reden hiervoor was dat wanneer de conversie van de dataset wordt uitgevoerd er gegevens kunnen veranderen en/of verloren gaan. Daarom zijn alle tabellen van zowel de dataset van de tien miljoen en honderd miljoen rijen volledig getest.

10.1.1 Rijen tellen

Het doel van deze test is om te kijken of er tijdens de conversie bepaalde rijen niet zijn meegenomen. Dit is de eerste test om aan te tonen dat de data conversie volledig is. Dit is op de volgende manier uitgevoerd:
`SELECT COUNT(*) FROM 'tabel'.`
Het resultaat van deze test geeft aan hoeveel rijen de tabel bevat waarop deze wordt uitgevoerd. Deze test is uitgevoerd voor elke tabel in Microsoft SQL Server 2014 en elke tabel in HDFS.

10.1.2 Tellen van numerieke & niet-numerieke waarden

Het doel van deze test is om te kijken of alle waarden per attribuut correct zijn meegenomen tijdens de conversie. Op deze manier kan ook worden aangetoond dat de data die tijdens de conversie op een juiste manier is meegenomen, zonder dat de waarden handmatig bekeken moeten worden. Daarnaast helpt deze test ook bij het aantonen dat de data conversie volledig is uitgevoerd.

Voor de numerieke waarden, zoals INT, TINITY, of bijvoorbeeld DOUBLE zijn er SUM functies uitgevoerd. Deze functie is op deze manier gebruikt:
`SUM(YearlyIncome).`
Voor alle niet-numerieke waarden zoals bijvoorbeeld VARCHAR en CHAR is het minder eenvoudig om aan te tonen dat deze goed zijn meegenomen tijdens de conversie. Daarom is ervoor gekozen om de lengte van de waarde op te tellen. Hier worden de volgende SQL functies voor gebruikt:
`SUM(LEN(EmailAddress)).`
De bovenstaande beschreven tests zijn gebruikt bij elk attribuut voor elke tabel in zowel Microsoft SQL Server 2014, als elke tabel in HDFS.

10.2 Resultaat

Nadat de verschillende testen waren gemaakt en verwerkt in scripts, zijn deze uitgevoerd op de dataset in Microsoft SQL Server 2014 en de geconverteerde dataset in HDFS. De resultaten van de verschillende testen per dataset zijn verwerkt in een Excel document.

10.2.1 Foutieve resultaten

Tijdens de eerste uitvoer van het tellen van de numerieke en niet-numerieke waarden op de tabellen in HDFS, bleek dat sommige waarden niet overeenkwamen met de resultaten in Microsoft SQL Server 2014. Dit was het geval in bijna elke tabel voor meerdere attributen. Daarom is uitgezocht waardoor dit kwam. Hierbij werd al snel duidelijk dat alle afwijkende waarden een overeenkomst hadden, er zaten NULL waarden in.

Het blijkt dat tijdens de uitvoer van de conversie de NULL waarden zijn omgezet naar een string: “NULL”. Daarom is ervoor gekozen om tijdens de uitvoer van de test voor elke attribuut dat NULL kan zijn een CASE statement toe te voegen. Deze CASE statement vervangt de “NULL” string met een lege string. Op deze manier komen er geen foute optelling tijdens de SUM. Na deze aanpassing kwamen de resultaten van met elkaar overeen, voor de verschillende testen. Deze resultaten worden in de volgende paragrafen beschreven.

10.2.2 Rijen tellen

Tabel	MS SQL Server COUNT resultaat	HDFS COUNT resultaat
DimCustomer	1018484	1018484
DimDate	2191	2191
DimEmployee	1000296	1000296
DimGeography	655	655
DimProduct	606	606
DimProductCategory	4	4
DimProductSubcategory	37	37
DimReseller	701	701
DimSalesTerritory	11	11
FactInternetSales	10060398	10060398
FactResellerSales	10060855	10060855

Tabel [13]: Resultaat dataset tien miljoen rijen.

Tabel	MS SQL Server COUNT resultaat	HDFS COUNT resultaat
DimCustomer	1018484	1018484
DimDate	2191	2191
DimEmployee	1000296	1000296
DimGeography	655	655
DimProduct	606	606
DimProductCategory	4	4
DimProductSubcategory	37	37
DimReseller	701	701
DimSalesTerritory	11	11
FactInternetSales	101304282	101304282
FactResellerSales	100370882	100370882

Tabel [14]: Resultaat dataset honderd miljoen rijen.

10.2.3 Waarden tellen

Naast het tellen van de rijen zijn ook alle individuele waarden per tabel geteld. De resultaten hiervan zijn te vinden in bijlage J '*Testrapport*' hoofdstuk *'3.3 Tellen van numerieke en niet-numerieke waarden*'.

De kwaliteitsattributen *´Juistheid´* en *´Volledigheid´* kunnen dankzij de bovenstaande resultaten worden aangetoond dat deze van toepassing zijn voor de conversie. Dit is wel bereikt nadat de testen voor een tweede keer waren uitgevoerd. Dit kwam doordat de eerste uitvoer die foutieve resultaten bevatte.

11. Experiment & shortlist

Nadat de gegevensconversie was getest kon er verder worden gegaan met de shortlist en het experiment. Het doel van de shortlist volgens de KPMG pakketselectie methode is: 'het in detail vergelijken van de kwaliteit en de mogelijkheden van de overgebleven pakketten'. De opdrachtgever wilde de volgende aspecten behandeld hebben tijdens het experiment voor de shortlist:

1. De snelheid van het pakket per ad-hoc query.
2. De gemiddelde snelheid van het pakket.
3. De schaalbaarheid van het pakket.

Zoals al eerder is beschreven in hoofdstuk *'5.3.3 Onderzoek'* worden het experiment en de shortlist gecombineerd, omdat er namelijk sprake is van hetzelfde doel. In dit hoofdstuk wordt beschreven hoe de ad-hoc query set tot stand is gekomen, op welke manier het experiment is aangepakt en uitgevoerd en wat voor resultaten er uit zijn gekomen.

11.1 Ad-hoc query set

Tijdens de uitvoer van het experiment moeten er SQL queries uitgevoerd. Het is van belang dat er een duidelijke set aan ad-hoc queries wordt gedefinieerd. In deze paragraaf wordt beschreven hoe deze ad-hoc query set tot stand is gekomen.

11.1.1 Info Support queries

Om te analyseren wat voor structuur ad-hoc queries hebben, zijn er Business Intelligence collega's benaderd binnen Info Support voor een interview. Deze collega's beschikken over veel kennis van SQL queries, omdat zij er veel mee werken bij klanten van Info Support. In totaal zijn er drie collega's benaderd die queries hebben aangeleverd. Na de derde collega waren er 147 queries beschikbaar om te analyseren. Deze queries komen voort uit verschillende projecten die deze collega's bij klanten van Info Support hebben uitgevoerd. De reden dat er niet meer collega's zijn benaderd, is omdat de set aan queries groot genoeg was voor analyse. Naast deze set worden er ook nog 99 queries uit een andere bron geanalyseerd. Hier is meer over te lezen in hoofdstuk *'11.1.2 Bestaande benchmarks'*.

Na overleg met de technisch begeleider is ervoor gekozen om de queries te analyseren met behulp van een wordcloud, om een overzicht te krijgen welke SQL commando's en functies vaak voorkomen in die queries die zijn aangedragen door de drie collega's. Deze wordcloud is gegenereerd met een online tool op: http://worditout.com/. Op deze manier is er een visuele representatie van de structuur van deze ad-hoc queries. Dit wordt vervolgens nog verduidelijkt met behulp van een tabel. Hierin wordt beschreven hoe vaak een bepaald SQL commando, functie of clausule voorkomt, inclusief het percentage. Dankzij de percentages kunnen de verschillende sets ad-hoc queries eenvoudig met elkaar worden vergeleken.

Woord	Frequentie	Percentage	Woord	Frequentie	Percentage	Woord	Frequentie	Percentage
and	513	340%	left	33	22%	null	16	11%
as	203	136%	group	33	22%	else	16	11%
from	180	122%	sum	31	21%	is	15	10%
select	180	122%	like	29	20%	all	10	7%
where	175	119%	count	29	20%	outer	8	5%
on	140	95%	when	26	18%	not	8	5%
join	140	95%	then	26	18%	avg	8	5%
by	101	68%	inner	26	18%	desc	4	3%
order	68	38%	distinct	19	13%	round	2	1%
year	46	31%	max	18	12%	having	2	1%
in	43	29%	case	17	11%			
or	41	28%	end	17	11%			

Tabel [15]: Frequentie van SQL commando's of functies, geanalyseerd uit de SQL queries van Info Support.

Afbeelding [17]: Wordcloud van de meest voorkomende SQL commando's of functies uit de SQL queries van Info Support.

Uit de resultaten van de analyse van zijn de volgende punten die opvallen:

1. SELECT en FROM statements zijn verplichte syntax in een SQL query. De reden dat er meer van deze statements zijn dan het aantal queries, komt doordat deze statements ook worden gebruikt bij subqueries. Het aantal subqueries komt dan neer op 22%.
2. Naast subqueries wordt er gebuik gemaakt van JOIN statements. Er zijn 33 LEFT, 26 INNER en 8 OUTER JOIN statements. De overige JOIN statements worden standaard als INNER JOIN statements behandeld [59]. In totaal zijn er 107 INNER JOIN statements.
3. De CASE, WHEN, THEN, END statements horen bij elkaar. De reden dat er meer WHEN en THEN statements zijn, is doordat deze vaker kunnen worden gebruikt binnen één CASE statement. Elk CASE statement eindigt uiteindelijk met een END statement.
4. Het BY statement wordt gebruikt door GROUP BY en ORDER BY. Wanneer de frequentie van GROUP en ORDER bij elkaar wordt opgeteld, komt dit overeen met de frequentie van BY.

11.1.2 Bestaande benchmarks

Tijdens het onderzoek kwam ook naar voren dat er bestaande benchmarks zijn voor databases. Deze benchmarks bevatten verschillende sets aan ad-hoc queries die veel voorkomende vragen uit de business nabootsen, waaronder ook reporting queries. Hieronder is een overzicht van drie organisaties die zich bezighouden met deze benchmarks:

1. Transaction Processing Performance Council (TPC) wordt beschouwd als een van de meest gebruikte database benchmarks [34,35].
2. Standard Performance Evaluation Corporation (SPEC) is een andere organisatie voor benchmarks binnen databases [36].
3. The Securities Technology Analysis Center (STAC) is de jongste organisatie, waarbij vooral de focus ligt op de beveiligingsindustrie [37].

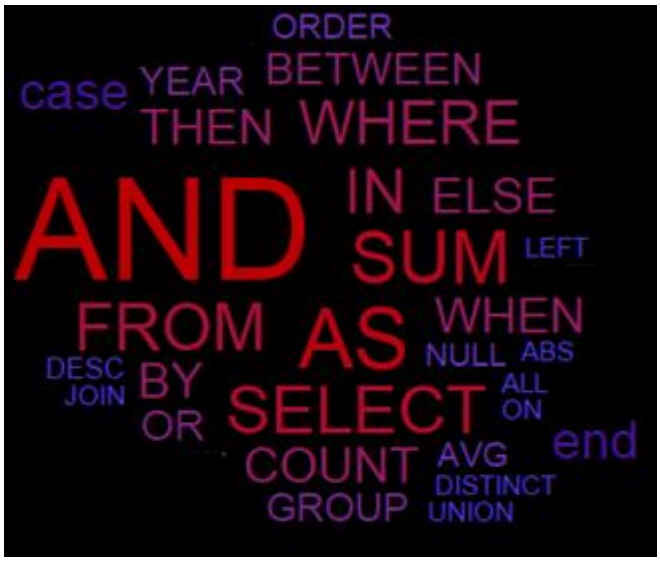
Het nadeel van de benchmarks van de bovenstaande organisaties is dat ze gefocust zijn op traditionele RDBMS pakketten. Het is niet eenvoudig om deze benchmarks te gebruiken voor SQL-on-Hadoop [34]. Daarnaast sluiten de benchmarks niet aan op de enorme workload die binnen SQL-on-Hadoop kan ontstaan [34]. Er is sinds kort wel een nieuwe benchmark van TPC beschikbaar, namelijk TPCx-HS [38]. Deze benchmark focust zich op de hardware en software performance die wordt gebruikt bij Big Data implementaties [38]. Alleen maakt deze benchmark geen gebruik van ad-hoc queries, maar van workload tests [39]. Daarnaast is deze benchmark tot nu toe alleen maar gebruikt door Cisco en Dell om hun hardware voor Big Data oplossingen mee te testen [39]. Daarom is ervoor gekozen om deze benchmark niet te gebruiken binnen voor het experiment.

Om bijvoorbeeld gebruik te maken van de TPC-DS benchmark, zou tijdens het experiment de set aan queries moeten worden aangepast om deze werkend te krijgen voor pakketten die gebruik maken van een SQL-like taal. Om dit te bereiken zou een gedeelte van de queries moeten worden herschreven. Uit een artikel van IBM bleek dat het herschrijven van de queries veel tijd in beslag neemt. De queries uit de TPC-DS benchmark werden herschreven om te werken met Cloudera Impala en Apache Hive. Het herschrijven nam vier weken per pakket in beslag [35]. Vanwege deze reden en de beschikbare tijd voor het afstuderen is er besloten om de bestaande RDBMS benchmarks niet te gebruiken.

Daarom is ervoor gekozen om de queries van de TPC-DS benchmark te analyseren met behulp van een wordcloud en de frequentie van de woorden in een tabel. Dit is op dezelfde manier aangepakt zoals beschreven in hoofdstuk '*11.1.1 Info Support queries*'. In tabel [16] is beschreven hoe vaak een bepaald SQL commando, of functie voorkomt. In totaal zijn er 99 queries geanalyseerd.

Woord	Frequentie	Percentage	Woord	Frequentie	Percentage	Woord	Frequentie	Percentage
and	560	566%	else	99	100%	null	33	33%
as	234	236%	case	99	100%	is	33	33%
sum	196	198%	end	99	100%	desc	17	17%
select	169	171%	count	98	99%	all	15	15%
from	169	171%	between	83	84%	union	14	14%
in	153	154%	or	71	72%	on	11	11%
where	143	144%	group	57	58%	join	11	11%
by	106	107%	year	57	58%	distinct	7	7%
when	99	100%	avg	51	52%	left	4	4%
then	99	100%	order	45	45%	abs	3	3%

Tabel [16]: Frequentie van SQL commando's of functies, geanalyseerd uit de TPC-DS benchmark queries.



Afbeelding [18]: Wordcloud van de meest voorkomende SQL commando's of functies uit de TPC-DS benchmark queries.

11.1.3 Info Support queries VS benchmark queries

De volgende grote verschillen zijn zichtbaar:

1. Het grootste verschil is dat de SUM functie bij de benchmark queries ongeveer twee keer per query wordt gebruikt. Bij de Info Support queries wordt de SUM functie maar één op de vijf keer per query gebruikt.
2. De AND operator komt bij de benchmark queries vaker voor vergeleken met de Info Support queries. Gemiddeld vijf keer per query, tegenover drie keer per query.
3. De benchmark queries hebben gemiddeld meer subqueries, 71%, vergeleken met de Info Support queries, 21%. In plaats daarvan hebben de Info Support queries wel meer JOIN statements, 95%, vergeleken met de benchmark queries, 11%.
4. Het laatste verschil is de hoeveelheid CASE statements die worden gebruikt, inclusief de bijbehorende statements. De benchmark queries bevatten gemiddeld één CASE statement voor elke query, bij de Info Support queries is dit gemiddeld één CASE statement voor elke tien queries.

11.1.4 Zelf gedefinieerde ad-hoc query set

Om de set van zelf gedefinieerde ad-hoc queries duidelijk te kunnen vergelijken met de benchmark en Info Support ad-hoc queries, zijn de resultaten van de Info Support queries en de benchmark queries samengevoegd. In totaal gaat het om 246 queries. Dit staat beschreven in tabel [17].

Woord	Frequentie	Percentage	Woord	Frequentie	Percentage	Woord	Frequentie	Percentage
and	1073	436%	case	116	47%	distinct	26	11%
as	437	178%	end	116	47%	inner	26	11%
select	349	142%	else	115	47%	all	25	10%
from	349	142%	year	114	46%	desc	21	9%
where	318	129%	order	113	46%	max	18	7%
sum	227	92%	or	112	46%	union	14	6%
by	207	84%	group	90	37%	outer	8	3%
in	197	80%	between	83	34%	not	8	3%
on	151	61%	avg	59	24%	abs	3	1%
join	151	61%	null	49	20%	round	2	1%
count	127	52%	is	48	19%	having	2	1%
when	125	51%	left	37	15%			
then	125	51%	like	29	12%			

Tabel [17]: Samengevoegde resultaten van Info Support SQL queries en benchmark queries.

Met behulp van de bovenstaande tabel is een eigen set aan ad-hoc queries gemaakt. Deze set bestaat uit twintig ad-hoc queries die overeenkomen met de kenmerken die hierboven staan beschreven. Er is bewust voor twintig queries gekozen, vanwege de verschillende combinaties van functies en statements. Hierbij zijn de volgende combinaties gemaakt:

1. Queries met vier, drie, twee en één JOIN statements.
2. LEFT JOIN statement in combinatie met INNER JOIN statement.
3. JOIN statements met aggregate functies gecombineerd.
4. Queries met drie, twee en één subqueries.
5. Subqueries met aggregate functies gecombineerd.
6. Subqueries in combinatie met een JOIN statement.
7. Subqueries met aggregate functies gecombineerd.
8. Queries waar expliciet geen resultaat is.

Bij het maken van deze queries was het niet eenvoudig om herhaling te voorkomen, doordat sommige queries op elkaar lijken qua syntax en uitkomst. Wanneer er bijvoorbeeld veertig queries zouden moeten worden gemaakt, zouden er meer queries zijn die bijna hetzelfde doen. Verder is het belangrijker dat de gebruikte functies en statements in de queries overeenkomen met de gebruikte functies en statements in de geanalyseerde queries, in plaats van het aantal queries dat is gemaakt.

Woord	Frequentie	Percentage	Woord	Frequentie	Percentage	Woord	Frequentie	Percentage
and	69	345%	case	9	45%	distinct	2	10%
as	35	175%	end	9	45%	inner	2	10%
select	31	155%	else	9	45%	all	2	10%
from	31	155%	year	9	45%	desc	2	10%
where	27	135%	order	9	45%	union	2	10%
sum	17	85%	or	9	45%	outer	2	10%
in	16	80%	between	7	35%	not	1	5%
by	14	70%	group	7	35%	abs	1	5%
on	14	70%	avg	4	20%	round	1	5%
join	14	70%	null	4	20%	having	1	5%
count	11	55%	is	4	5%	max	1	5%
when	9	45%	left	3	15%			
then	9	45%	like	2	10%			

Tabel [18]: Resultaten van de zelf gedefinieerde ad-hoc query set.

Zoals te zien in tabel [18] hebben de resultaten van de eigen ad-hoc query set weinig afwijkingen vergeleken met de samengevoegde Info Support en benchmark queries. Vervolgens is deze set aan queries nog een keer voorgelegd aan verschillende Business Intelligence collega's, om feedback op te krijgen. Hierbij was de feedback positief, waardoor deze set aan queries gebruikt kan worden tijdens het experiment.

11.1.5 Herziene ad-hoc query set

Tijdens het uitvoeren van de ad-hoc queries bij het experiment bleek dat Cloudera Impala geen subqueries ondersteunt. Dit staat uitgebreider beschreven in hoofdstuk *´11.2.4 Herschrijven ad-hoc queries per pakket'*. Omdat voor Cloudera Impala de queries die een subquery bevatte moesten worden herschreven, is ervoor gekozen subqueries uit de zelf gedefinieerde ad-hoc query set te halen. Dit was een eis van de opdrachtgever. De opdrachtgever vond anders dat het resultaat van het experiment niet goed gegarandeerd zou kunnen worden.

Verder was er nog op verzoek van de opdrachtgever een extra query toegevoegd. Dit brengt het totaal van de ad-hoc query set op eenentwintig queries. Deze query is zeer klein en haalt precies één record op, door middel van een WHERE conditie met een uniek ID. In bijlage E *'Ad-hoc queries*' is een overzicht te vinden met alle ad-hoc queries voor de verschillende pakketten.

11.2 Aanpak

11.2.1 Beschikbare versies

Op Amazon EMR, de Hadoop distributie van Amazon, zijn de volgende versies van de pakketten gebruikt:

1. Cloudera Impala: 1.2.4
2. Apache Drill: 0.8
3. Stinger Initiative: Apache Hive 0.13, Apache Tez 0.6
4. Apache Hadoop: 2.4

Voor Cloudera Impala en Apache Hive waren deze versies 'out-of-the-box' beschikbaar. Voor Apache Drill en Apache Tez waren er bootstrap scripts beschikbaar. Deze zijn aangepast, zodat er gebruik kon worden gemaakt van de

huidige versies. De nulmeting is uitgevoerd met Apache Hive 0.12. Deze versie heeft namelijk de verbeteringen van het Stinger Initiative nog niet.

De huidige versie van Cloudera Impala is 2.1.0. Vergeleken met de gebruikte 1.2.4 versie loopt Cloudera Impala zes versies achter op de huidige versie [40]. Voor Apache Drill is de huidige versie 0.8. Hier zit geen verschil in met de gebruikte versie [41]. Voor Stinger Initiative is de huidige versie van Apache Hive 1.0.1. Deze versie loopt drie versies achter op de huidige versie [42]. Voor Apache Tez is de huidige versie 0.6. Hier zit geen verschil in met de gebruikte versie [43].

De reden dat de huidige versie van Cloudera Impala niet was gebruikt, kwam vanwege het feit dat deze niet beschikbaar was op Amazon EMR. Een handmatige update gaf aan dat deze niet in de repository beschikbaar was. Een andere mogelijkheid zou zijn om Cloudera Impala handmatig te installeren en configureren op alle Hadoop nodes. Hier is niet voor gekozen vanwege de complexiteit die dit met zich meebrengt en wordt uitgebreider beschreven in de volgende paragraaf. Hetzelfde geld voor Apache Tez.

11.2.2 Lokale virtual machine

Tijdens het oefenen met de verschillende pakketten op een lokale virtual machine bleek al dat het handmatig opzetten van een Hadoop cluster veel tijd en kennis vergt. Het was gelukt om een Hadoop cluster met één node op te zetten. Hier moesten vervolgens nog de drie pakketten op geïnstalleerd en geconfigureerd worden. Om deze pakketten te installeren en configureren was er uitgebreide kennis nodig van de verschillende pakketten.

Na ongeveer twee weken was het gelukt om lokaal Apache Drill en Apache Hive draaiend te krijgen op een eigen Hadoop cluster met één node. De installatie gebeurde in Linux. Hier had zelf ik nog geen ervaring mee. Vervolgens moesten Cloudera Impala en Apache Tez nog geïnstalleerd en geconfigureerd worden. De installatie tot dusver was alleen uitgevoerd op één node. Wanneer er gebruik wordt gemaakt van meerdere nodes binnen een cluster moet de communicatie ook nog worden geregeld tussen deze nodes. Hier moet Zookeeper voor worden gebruikt. Daarnaast moeten de pakketten ook op de andere nodes worden geïnstalleerd en geconfigureerd worden en zijn er optimalisaties nodig voor het verdelen van het werk.

Tijdens de installatie en configuratie ben ik meerdere keren vastgelopen op errors, of dingen die ik niet wist. Verder duurde dit onderdeel langer dan was gepland. Het gevolg hiervan was dat ik ongeveer twee weken achterliep op de planning. Aan de hand van de planning is toen vervolgens met de opdrachtgever bekeken hoe dit opgelost kon worden. Er was anderhalve week uitloop ingepland. Verder is er toen ook besloten om te stoppen met het installeren en configureren op de lokale virtual machine en direct de cloud omgeving op te zetten. Het daadwerkelijk uitvoeren van het experiment moest uiteindelijk toch op een cloud omgeving worden gedaan.

11.2.3 Amazon EMR configuratie

Voor het uitvoeren van de ad-hoc query set op Amazon EMR is er gebruik gemaakt van de volgende soort server:

1. C3.xlarge on Linux.

Dit type server heeft 4 CPU cores en 7.5GB RAM geheugen en was aangeraden door de Amazon AWS helpdesk. Deze was benaderd, omdat ik niet precies wist wat voor type server er nodig was. Aan de hand van de business case die was voorgelegd is dit advies gegeven. Deze business case beschreef wat er met de Amazon EMR cluster bereikt moest worden; het vergelijken van meerdere ‘SQL-on-Hadoop’ pakketten.

Vervolgens is deze hardware configuratie voor elke uitvoer van het experiment gebruikt. Op deze manier heeft de hardware zo min mogelijk invloed op de resultaten van het experiment.

11.2.4 Herschrijven ad-hoc queries per pakket

Tijdens het testen of de ad-hoc query set die in Microsoft SQL Server 2014 was opgesteld ook konden worden uitgevoerd op de drie pakketten zijn er een aantal veranderingen doorgevoerd. Deze veranderingen staan hieronder beschreven:

Cloudera Impala
Cloudera Impala ondersteunt in versie 1.2.4 op Amazon EMR geen subqueries. Daarom zijn deze in overleg met de opdrachtgever herschreven naar INNER JOIN statements. Vanaf Cloudera Impala 2.0 worden subqueries wel ondersteund [44]. Voor elke ORDER BY is er een LIMIT clausule nodig. Deze zijn toegevoegd aan de ad-hoc queries voor Cloudera Impala. Vanaf Cloudera Impala 1.4 is de LIMIT clausule niet meer verplicht [45]. Om ervoor te zorgen dat dit geen invloed heeft op de resultaten, is de waarde voor de LIMIT clausule hoger gezet dan de waarde die uit de resultset zou voortkomen.

Apache drill
Apache Drill ondersteunt geen TINYINT als datatype[46]. Dit veroorzaakt errors wanneer in de WHERE en AND conditie een attribuut wordt gebruikt met het datatype TINYINT. Hetzelfde geldt wanneer er voor een TINYINT een CAST wordt gebruikt [47]. Dit is opgelost door de attributen die een TINYINT zijn in de WHERE en AND condities te vervangen met een ander datatype, namelijk een INT.

Apache Drill ondersteunt de YEAR() functie niet [48]. Dit is opgelost door YEAR() te vervangen voor een BETWEEN conditie waarbij de datum binnen de grenzen van een bepaald jaar moet vallen.

Tot slot ondersteunt Apache Drill geen UNION-ALL voor meer dan twee resultsets [49]. Dit is opgelost door de query te herschrijven waarbij er maar twee resultsets worden gecombineerd, in plaats van drie. Zoals al eerder beschreven zijn alle subqueries vervangen voor INNER JOIN statements.

Apache Hive
Apache Hive 0.13 ondersteunt maximaal één subquery per query [50]. Vanaf Hive 0.14 is het mogelijk om meerdere subqueries te gebruiken [51]. Zoals al eerder beschreven zijn alle subqueries vervangen voor INNER JOIN statements.

11.3 Uitvoer

Elk pakket heeft HDFS gebruikt om de data uit de dataset op te halen met behulp van de queries. Deze dataset was met behulp van SQOOP geconverteerd en ingeladen. Het uitvoeren van deze conversie is beschreven in hoofdstuk '*9.2.3 Uitvoer conversie*', net als het generen van de dataset. Nadat de dataset was ingeladen in HDFS moesten de tabellen nog geregistreerd worden in de Hive Metastore. Op deze manier is de metadata beschikbaar en kunnen alle pakketten hier gebruik van maken. Vanaf dit moment was het ook mogelijk om de ad-hoc queries uit te voeren.

Tijdens het uitvoeren van de queries op de verschillende pakketten werd al snel duidelijk dat query 16 op geen enkel pakket werd ondersteund. Dit kwam doordat er een SELECT werd uitgevoerd in een kolom. Deze vorm van subquery werd bij geen enkel pakket ondersteund.

Zoals al beschreven in het onderzoeksontwerp zijn er een aantal combinaties uitgevoerd met het aantal nodes en de grootte van de dataset. Dit is te vinden in tabel [19].

Run	Dataset	Aantal core nodes
1	10 Miljoen records	2
2	10 Miljoen records	4
3	10 Miljoen records	8
4	100 Miljoen records	2
5	100 Miljoen records	4
6	100 Miljoen records	8

Tabel [19]: Overzicht combinatie tussen aantal core nodes en dataset grootte.

Voor elke run was er een nieuwe cluster aangemaakt. Op deze manier is er geen mogelijkheid dat queries zijn gecached. Dit zou namelijk invloed kunnen hebben op de resultaten. In totaal zijn alle runs twee keer uitgevoerd. Dit komt neer op twaalf runs totaal. De reden dat alles twee keer is uitgevoerd, is om aan te tonen dat de resultaten niet op toeval berusten en dat het experiment herhaalbaar is. De resultaten van de uitvoer worden in de volgende paragraaf beschreven. Verder is er per query bijgehouden wat voor resultset er was bij de nulmeting. Vervolgens is per pakket ook bekeken of hetzelfde resultset uit de query is gekomen. Deze kwamen overeen met die van de nulmeting.

Alle queries zijn in de console, via een SSH verbinding, uitgevoerd. De queries zijn handmatig uitgevoerd in de console. Er was namelijk geen rekening gehouden met het maken van een script dat de queries automatisch kan uitvoeren en de resultaten kan opslaan. Voor Apache Hive was het ook mogelijk om via een web interface de ad-hoc queries uit te voeren. Maar om het experiment zo zuiver mogelijk te houden is ervoor gekozen om alle queries via dezelfde console interface uit te voeren.

11.3.1 Nulmeting; Apache Hive 0.12

Voor elke run is als eerste de nulmeting uitgevoerd. Deze is uitgevoerd met behulp van Apache Hive 0.12. Hierbij is de execution engine op ‘MR’ gezet. Dit is gedaan door middel van: ‘*set hive.execution.engine=MR*’ en is tevens de standaard execution engine. Dit houdt in dat de queries worden uitgevoerd op de MapReduce engine. Apache Hive maakt al automatisch gebruik van de eigen Hive Metastore. Het enige dat nog nodig was om de queries uit te kunnen voeren was aangeven in welke database de dataset was opgeslagen.

Wanneer een query in de nulmeting klaar was gaf de console aan hoeveel seconden en milliseconden erover is gedaan. Deze tijd was gebaseerd op het uitvoeren van de query en het ophalen van de resultset. De resultset werd vervolgens getoond in de console.

Nadat de nulmeting was uitgevoerd werd de cluster opnieuw opgestart. Op deze manier is de kans dat bepaalde queries gecached zijn niet meer van toepassing. Vervolgens zijn de ad-hoc queries uitgevoerd op Apache Drill, Cloudera Impala en Stinger Initiative. Om aan te tonen dat de resultaten niet worden beïnvloed doordat een bepaald pakket steeds als eerste wordt getest, is ervoor gekozen om per run een willekeurige volgorde aan te houden. Zo wordt elk pakket meerdere keren als eerste, tweede of derde getest.

11.3.2 Stinger Initiative

Om gebruik te maken van Stinger Initiative is de cluster opgestart met Apache Hive 0.13. Vervolgens is aangegeven in welke dataset de database was opgeslagen. Omdat Stinger Initiative gebruik maakt van Apache Tez als execution engine is dit aangegeven door middel van: ‘*set hive.execution.engine=tez*’. Voor de werking van Apache Tez kan er

worden gekeken in bijlage C '*Onderzoeksdocument*' hoofdstuk *'4.3 Hoe zitten de pakketten op de shortlist technisch in elkaar?'*. Vervolgens zijn de ad-hoc queries uitgevoerd.

11.3.3 Apache Drill

Het installeren en configureren van Apache Drill was niet 'out-of-the-box' beschikbaar. Dit is gedaan met behulp van een bootstrap script. Deze scripts worden uitgevoerd wanneer een Amazon EMR cluster wordt opgestart. Wanneer er meer dan één node is, moeten de overige nodes nog handmatig worden geactiveerd, zodat Apache Drill deze nodes kan gebruiken. Om de ad-hoc queries uit te kunnen voeren moet Apache Drill gebruik maken van de Hive Metastore. Daarnaast moet ook de correcte database worden gebruikt waar de dataset in is opgeslagen.

Omdat Apache Drill geïnstalleerd is in een 'clustered' omgeving, moeten de verschillende nodes met elkaar communiceren. Dit is geregeld door Apache Zookeeper te installeren tijdens de installatie van Apache Drill. In bijlage C '*Onderzoeksdocument*' hoofdstuk *'4.3 Hoe zitten de pakketten op de shortlist technisch in elkaar?'* wordt uitgebreider behandeld hoe de verschillende Apache Drill nodes met elkaar communiceren.

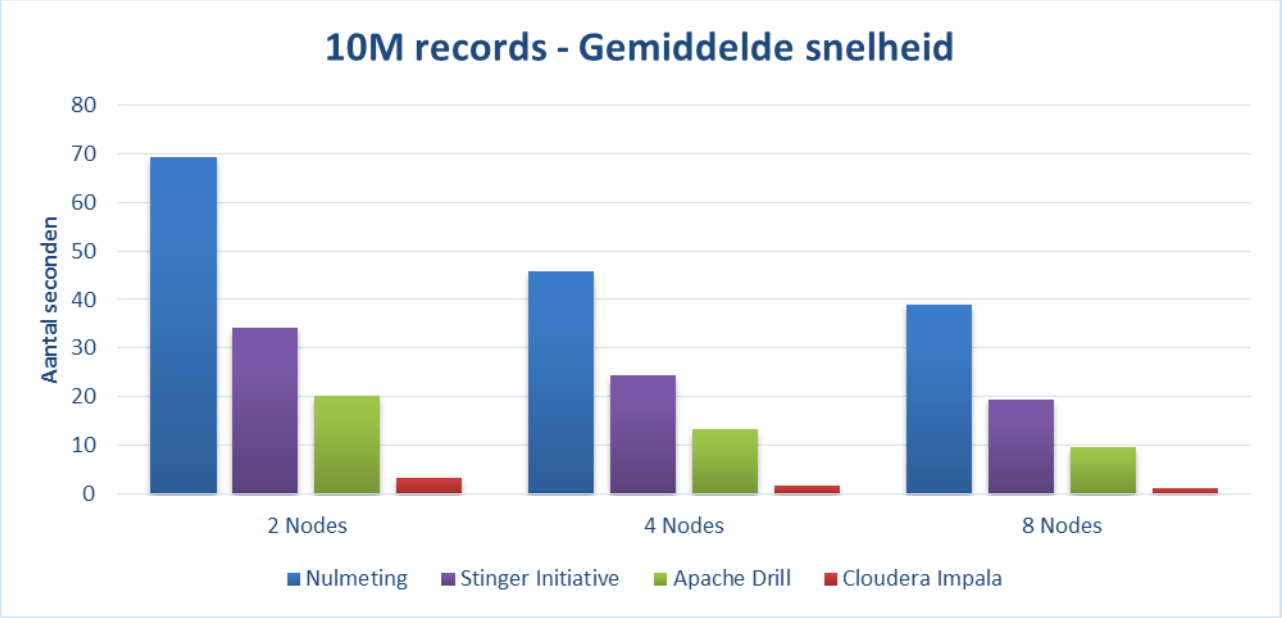
Tijdens de uitvoer van de ad-hoc queries op de dataset met honderd miljoen records was er een error bij query 9. Deze gaf aan dat er een 'out of memory failure' was. Dit gebeurde bij de run met twee nodes en met vier nodes. Bij acht nodes was dit niet het geval. Wanneer een query was uitgevoerd werd op de console zichtbaar hoeveel seconden en milliseconden de query in beslag nam om uitgevoerd te worden.

11.3.4 Cloudera Impala

Cloudera Impala was op Amazon EMR een 'out-of-the-box' configuratie. Om de Hive Metastore te gebruiken moest met behulp van het statement '*invalidate metadata'* de connectie met de Hive Metastore worden ververst. Nadat dit was gedaan konden de queries worden uitgevoerd. De tijd die in beslag nam werd na elke uitvoer vermeld in de console.

11.4 Resultaten gemiddelde snelheid

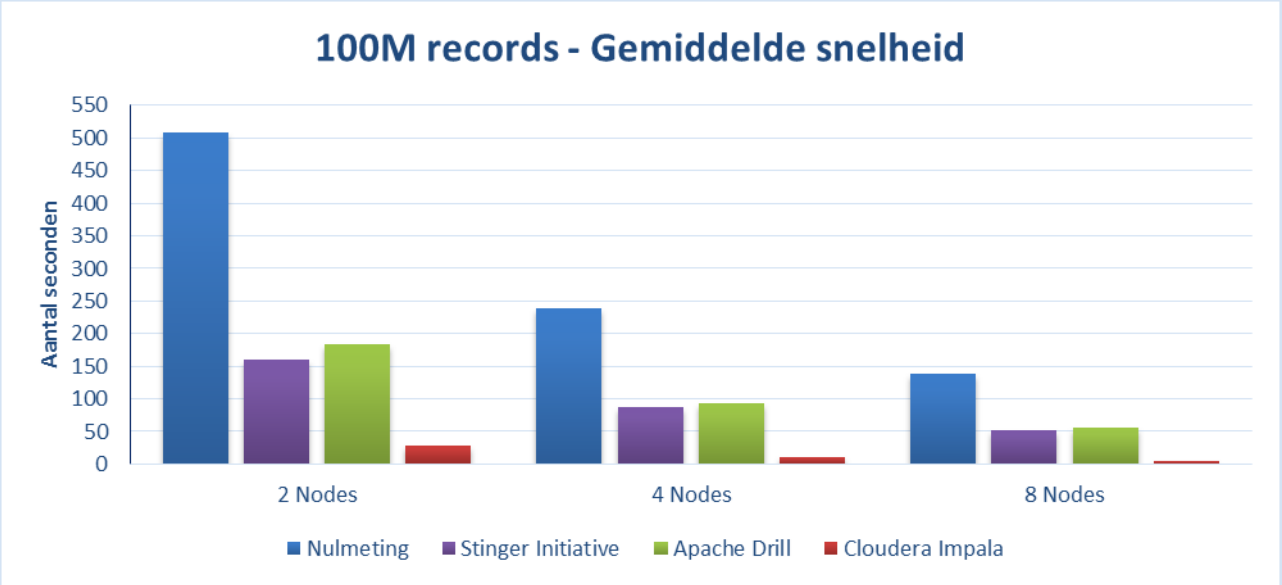
Om de deelvraag te kunnen beantwoorden welk pakket gemiddeld het snelste de queries uitvoert, is de gemiddelde snelheid van de 21 queries genomen. De resultaten hiervan staan hieronder beschreven. De 'rauwe' data van alle metingen zijn te vinden in bijlage F *'Experiment data*'.



Afbeelding [19]: Gemiddelde snelheid voor tien miljoen rijen.

In afbeelding [19] is te zien dat voor de uitvoer van het experiment met tien miljoen records Cloudera Impala veruit de snelste gemiddelde snelheid heeft. Dit geld voor twee, vier en acht nodes. Wanneer de resultaten worden vergeleken met de nulmeting, is Cloudera Impala op twee nodes 21 keer sneller. Op vier nodes 29 keer sneller en op acht nodes maar liefst 31 keer sneller.

Op de tweede plaats komt Apache Drill. Apache Drill is vergeleken met de nulmeting op twee nodes ongeveer 3.5 keer sneller. Op vier nodes ongeveer 4 keer sneller. Op acht nodes ongeveer 4 keer sneller. Op de derde plaats komt Stinger Initiative. Stinger Initiative is vergeleken met de nulmeting op twee nodes ongeveer 2 keer sneller. Op vier nodes ongeveer 2 keer sneller en op acht nodes ook ongeveer 2 keer sneller.



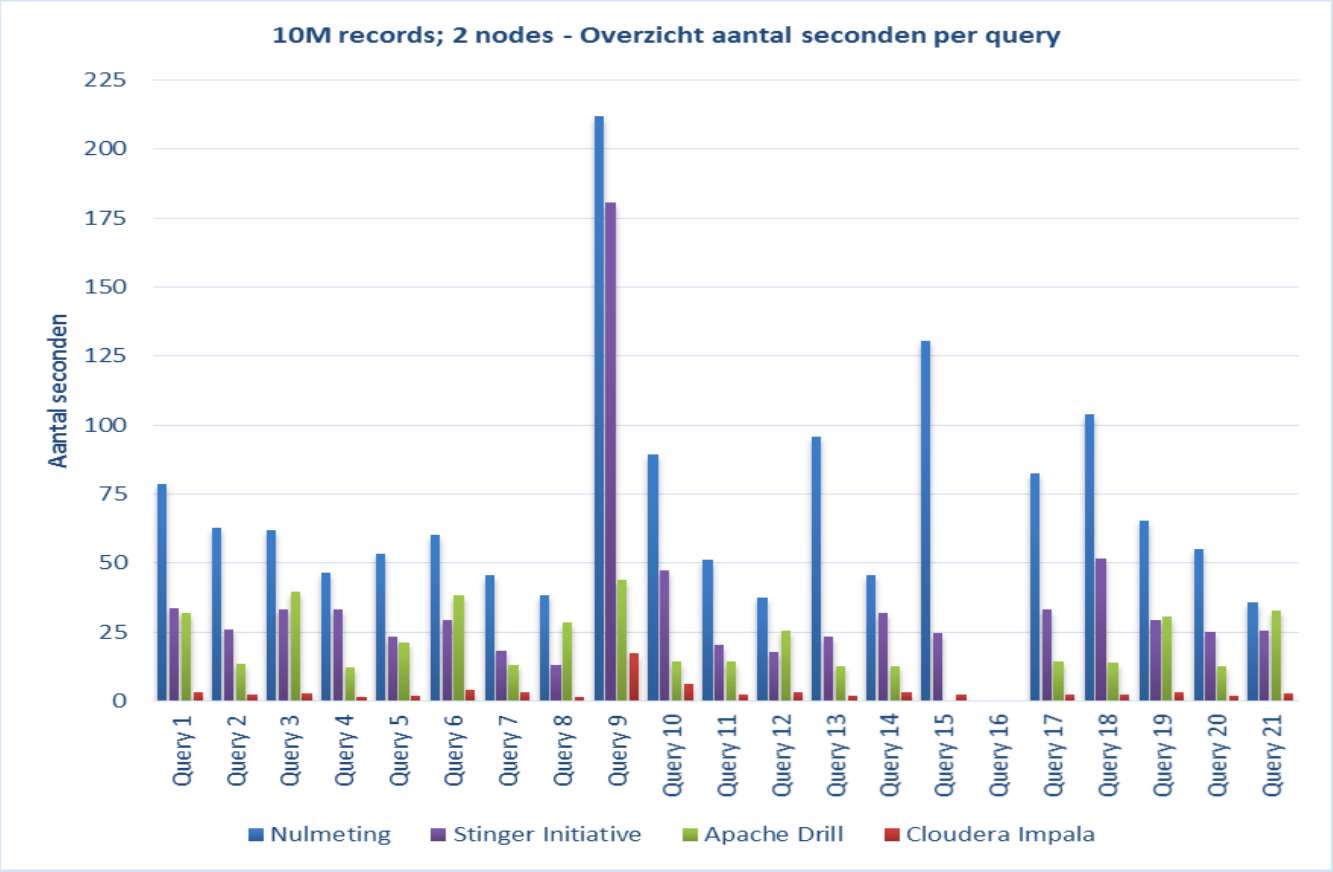
Afbeelding [20]: Gemiddelde snelheid voor honderd miljoen rijen.

In afbeelding [20] is de gemiddelde snelheid te zien van het experiment met honderd miljoen records. Ook hierbij heeft Cloudera Impala de snelste gemiddelde snelheid bij twee nodes, vier nodes en acht nodes. Wanneer ook hier de resultaten worden vergeleken met de nulmeting, is Cloudera Impala op twee nodes ongeveer 18 keer sneller. Op vier nodes ongeveer 24 keer sneller en op acht nodes ongeveer 27 keer sneller.

Een opvallend verschil met de meting op tien miljoen rijen is dat Stinger Initiative nu op de tweede plaats staat. Stinger Initiative is vergeleken met de nulmeting op twee nodes ongeveer 3 keer sneller. Op vier nodes ongeveer 3 keer sneller en op acht nodes ook ongeveer 3 keer sneller. Op de derde plaats komt Apache Drill. Apache Drill is vergeleken met de nulmeting op twee nodes ongeveer 3 keer sneller. Op vier nodes ongeveer 2.5 keer sneller en op acht nodes ook ongeveer 2.5 keer sneller.

11.5 Resultaten snelheid per query

Naast de gemiddelde snelheid is er ook gekeken naar de snelheid per query. Query 16 heeft in geen enkel overzicht resultaten. Bij Query 15 heeft Apache Drill geen resultaten. De resultaten staan hieronder beschreven:

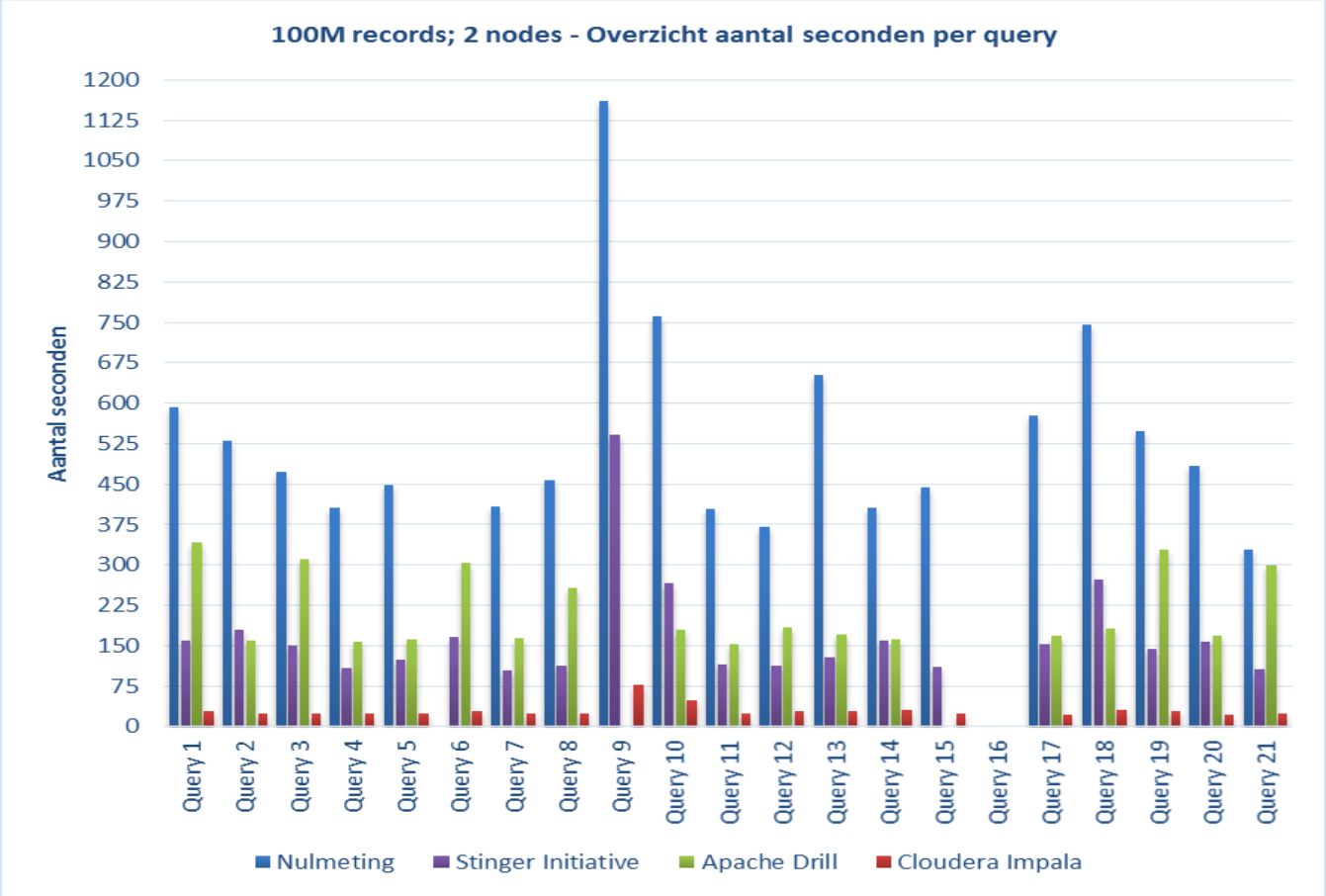


Afbeelding [21]: Overzicht resultaten op tien miljoen rijen met twee nodes.

In afbeelding [21] is te zien dat Cloudera Impala bij alle queries het snelste is. Zowel Stinger Initiative als Apache Drill zijn bij een aantal queries op tweede of derde plaats geëindigd.

Wanneer er wordt gekeken naar de verschillen tussen Apache Drill en Stinger Initiative vallen een aantal dingen op. Zo is Apache Drill over het algemeen sneller dan Stinger Initiative bij het experiment van tien miljoen records. Deze trend is ook te zien bij vier en acht nodes. De queries waar Stinger Initiative sneller bij was vergeleken met Apache Drill hebben veel met elkaar gemeen. Zo zijn er bijna geen aggregaat functies, alleen maar '*SELECT * '* statements. Verder hebben de meeste van deze queries één of meerdere JOIN statements. Apache Drill is dus bij queries met aggregaat functies sneller dan Stinger Initiative.

De resultaten van het experiment voor vier en acht nodes op tien miljoen records zijn te vinden in bijlage C '*Onderzoeksdocument*' hoofdstuk *'6.4 Snelheid per query*'.



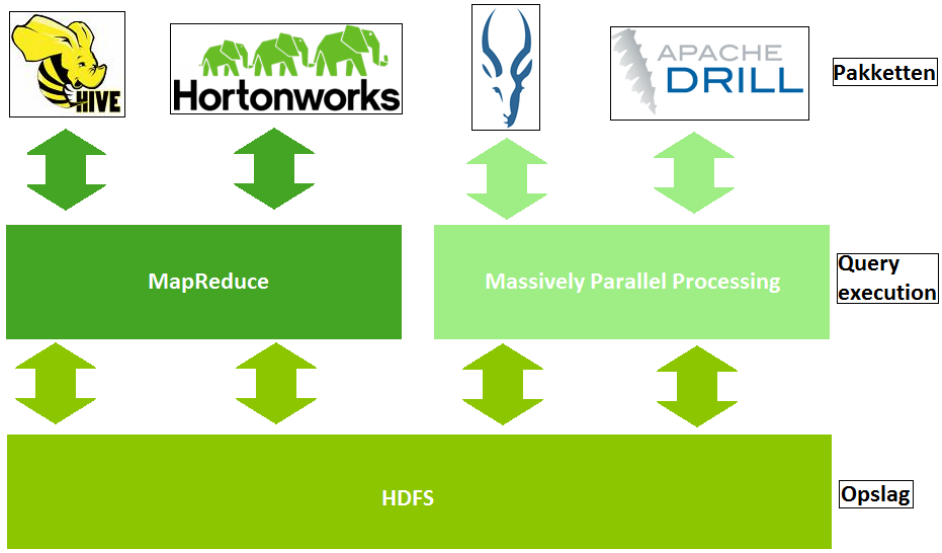
Afbeelding [22]: Overzicht resultaten op honderd miljoen rijen met twee nodes.

In afbeelding [22] is te zien dat Cloudera Impala het snelste is bij alle queries. Een opmerkelijk verschil tussen de meting met tien miljoen records en honderd miljoen records, is dat Stinger Initiative nu bij bijna alle queries sneller is dan Apache Drill. Hier wordt in hoofdstuk *11.5.3 Cloudera Impala vs Apache Drill vs Stinger Initiative*' verder op ingegaan. Deze trend is ook te zien bij de resultaten van het experiment voor vier en acht nodes. De resultaten van het experiment voor vier en acht nodes op honderd miljoen records zijn te vinden in bijlage C '*Onderzoeksdocument*' hoofdstuk *'6.4 Snelheid per query*'.

De reden voor de verschillen in resultaten komt doordat de pakketten gebruik maken van verschillende technieken. Tijdens het beantwoorden van de deelvraag '*Hoe zitten de pakketten op de shortlist technisch in elkaar?*' is er naar de technische werking van de pakketten gekeken.

Zowel Cloudera Impala als Apache Drill zijn geïnspireerd op Google Dremel en maken ook gebruik van de beschreven technologieën die zijn toegepast in Google Dremel [52, 53]. Google Dremel is een '*massively parallel query engine'*, afgekort tot MPP (Massively Parallel Processing) [54, 55]. Dit houdt in dat meerdere processoren zich bezighouden met het verwerken en uitvoeren van een bepaald onderdeel van de query. Het verwerken en uitvoeren wordt gecoördineerd en er komt een resultaat uit [54, 55]. Google Dremel zorgt ervoor dat er op datasets van honderden miljoenen tot miljarden rijen snel queries uitgevoerd kunnen worden. Hierbij valt te denken aan enkele seconden tot tientallen seconden, afhankelijk van de grote van de dataset. Dit is mogelijk door het gebruik van twee technologieën die ervoor zorgen dat queries zo snel kunnen worden uitgevoerd: Columnar Storage en Tree Architecture [54, 55].

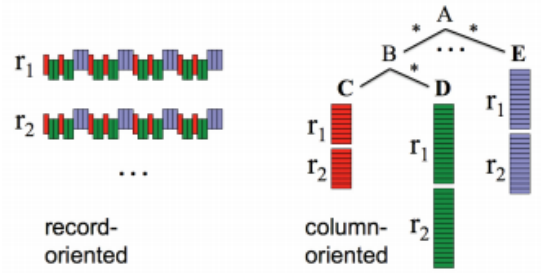
In afbeelding [23] is duidelijk te zien dat de query execution laag verschillend is bij de pakketten.



Afbeelding [23]: Lagenstructuur van de verschillende pakketten.

11.5.1 Columnar Storage

Columnar Storage houdt in dat data wordt opgeslagen in kolommen. Dit betekent dat een rij wordt opgesplitst in verschillende kolom waarden. Deze kolom waarden worden vervolgens op verschillende locaties op het opslag medium opgeslagen [54, 55]. In tegenstelling tot traditionele databases waar de hele rij wordt opgeslagen op één opslag medium [54, 55]. In afbeelding [24] wordt dit geïllustreerd.

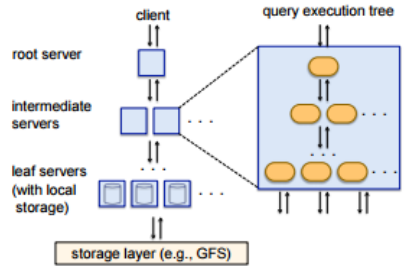


Afbeelding [24]: Columnar storage in Google Dremel [54].

Het voordeel van Columnar Storage is dat alleen de benodigde waarden worden gescand bij een query. Wanneer de volgende query wordt uitgevoerd: *SELECT TOP 10 ProductKey FROM FactResellerSales*, hoeft alleen nog de '*ProductKey*' kolom worden gescand. Hierdoor hoeft er veel minder data te worden gescand voor de query [55].

11.5.2 Tree Architecture

Wanneer een query wordt uitgevoerd, wordt dit gedaan met behulp van een Tree Architecture. In afbeelding [25] wordt dit geïllustreerd.



Afbeelding [25]: Tree Architecture in Google Dremel [54].

De Tree Architecture wordt gebruikt om snel een query te verdelen over verschillende servers. De 'Root server' coördineert alles. De 'Intermediate servers' zijn verantwoordelijk voor het samenvoegen van de resultaten uit de 'Leaf servers'. De 'Leaf servers' zijn verantwoordelijk voor het scannen en ophalen van de data. Dankzij de Tree Architecture is het mogelijk om met Google Dremel 'Massively Parallel Processing', MPP uit te voeren op queries [55].

11.5.3 Cloudera Impala vs Apache Drill vs Stinger Initiative

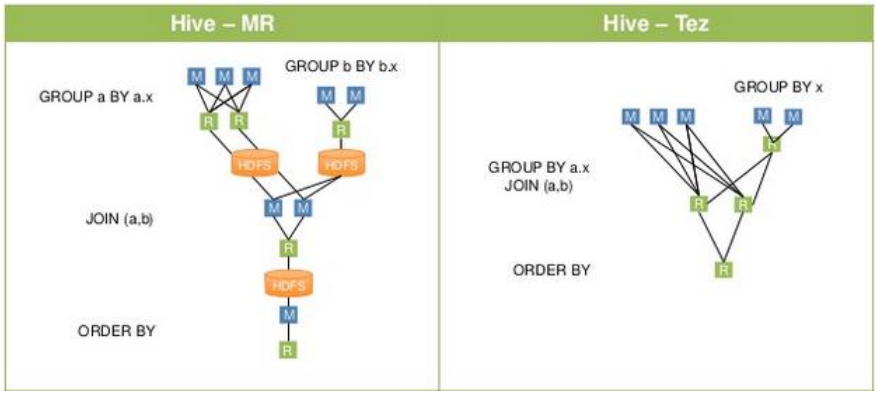
Zoals al eerder beschreven maken Cloudera Impala en Apache Drill gebruik van MPP. Stinger Initiative maakt gebruik van MapReduce. Zowel MPP als MapReduce kunnen worden gebruikt om enorme hoeveelheden data te verwerken. Toch zijn er verschillen tussen MPP en MapReduce. Om MapReduce te gebruiken moet er worden geprogrammeerd in Java. Bij MPP databases is dit in SQL. MapReduce kan ook bijvoorbeeld ongestructureerde data verwerken, dit is niet mogelijk bij MPP. MapReduce is ontwikkeld met als doel om '*batch processing*' uit te voeren over grote datasets. Google Dremel, met MPP, is ontwikkeld met als doel om snel analyses uit te kunnen voeren op grote datasets [54].

Qua performance is er ook een verschil. Zoals beschreven in bijlage C '*Onderzoeksdocument*' hoofdstuk *'4.2 Welke tekortkomingen heeft Hadoop?*', kan het opstarten van een MapReduce taak al tientallen seconden duren. Vervolgens moet deze taak nog worden uitgevoerd. Bij MPP is er geen MapReduce, dus is er ook geen sprake van deze vertraging.

Wanneer er wordt gekeken naar de resultaten is dit ook zichtbaar. De nulmeting is overal trager. Stinger Initiative maakt ook gebruik van MapReduce. Toch is Stinger Initiative sneller dan de nulmeting. Dit komt door een aantal optimalisaties, waarvan twee belangrijke hieronder worden beschreven:

1. Het opstarten van MapReduce jobs en taken is weggehaald. Dit kan een aantal tot tientallen seconden schelen per job of taak. Wanneer er meerdere jobs of taken moeten worden opgestart is er een groot verschil in de performance [57].

2. Bij Stinger Initiative hoeven tijdelijke resultaten niet meer te worden weggeschreven in HDFS. Dit wordt in afbeelding [26] geïllustreerd. Dit bespaart tijd tijdens het uitvoeren van een query, doordat er in het oude voorbeeld drie keer iets moet worden weggeschreven in HDFS.



Afbeelding [26]: Overzicht verschil tussen MapReduce en Tez als execution engine [57].

Cloudera Impala is vergeleken met de nulmeting en Stinger Initiative veel sneller. Dit komt, zoals eerder beschreven, doordat Cloudera Impala geen gebruik maakt van MapReduce, maar Massively Parallel Processing. Hoe MPP verder is uitgewerkt in Cloudera Impala en welke optimalisaties zijn toegevoegd wordt helaas nergens beschreven in de documentatie van Cloudera Impala.
Naast Cloudera Impala maakt Apache Drill ook gebruik van Massively Parallel Processing. Wanneer er wordt gekeken naar de resultaten met tien miljoen rijen is Apache Drill sneller dan de nulmeting, maar niet altijd sneller dan Stinger Initiative. Wanneer er wordt gekeken naar het soort queries waar Apache Drill sneller is dan Stinger initiative, is er een verband te zien. Zo bevatten veel van deze queries één of meerdere aggregaat functies, zoals SUM, COUNT of AVG. Dit komt doordat MPP zeer snel resultaten kan aggregeren, iets dat ook wordt beschreven in het Whitepaper van Google Dremel [54]. Verder wordt in dit Whitepaper beschreven dat MapReduce beter kan worden gebruikt voor grote JOIN operaties [54]. Tijdens het experiment komt dit gedeeltelijk terug. De queries waar Stinger Initiative sneller is dan Apache Drill bevatten over het algemeen één of meerdere JOIN operaties, zonder aggregaat functies.

Mede doordat de data gedenormaliseerd is, blijft het aantal mogelijke JOIN operaties in deze dataset redelijk beperkt. Wanneer er een genormaliseerde dataset zou zijn gebruikt zouden er veel meer JOIN operaties kunnen worden gebruikt. Op deze manier zou er nog beter kunnen worden onderzocht wat voor impact een grote hoeveelheid JOIN operaties zouden hebben op de performance.

De resultaten van het experiment met honderd miljoen rijen is anders dan die van tien miljoen records. Zo is Stinger Initiative bij bijna alle queries sneller dan Apache Drill. Een reden hiervoor zou kunnen zijn doordat MPP veel in-memory uitvoert [54]. Apache Drill beschrijft ook op de pagina over de architectuur dat de queries zoveel mogelijk in-memory worden uitgevoerd [56]. Wanneer dit niet mogelijk is wordt de harde schijf gebruikt voor ‘virtual memory’ [56]. Virtual memory is alleen veel langzamer vergeleken met RAM geheugen, omdat de lees en schrijf snelheid van een harde schijf veel lager ligt dan dat van RAM geheugen.

Een observatie tijdens het experiment was dat bij tien miljoen records op de vier en acht nodes, sommige nodes ‘idle’ waren. Deze nodes hadden nog hun complete RAM geheugen beschikbaar. Bij honderd miljoen records werd het RAM geheugen van alle nodes volledig gebruikt. Dit geeft aan dat bij tien miljoen records de queries vaker uitgevoerd konden worden in-memory, in tegenstelling bij honderd miljoen records. Wanneer er wordt gekeken naar de resultaten van vier en acht nodes bij het experiment met honderd miljoen rijen, is zichtbaar dat Apache Drill

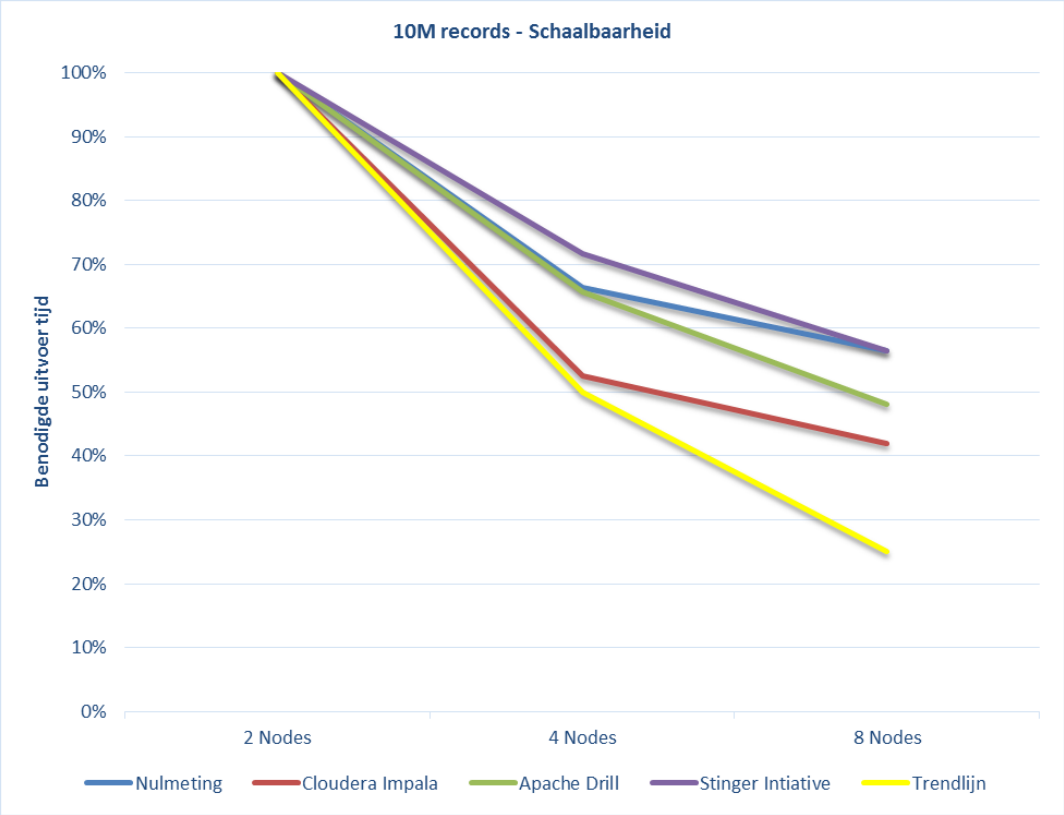
bij meer queries een snellere performance heeft dan Stinger Initiative. Dit geeft ook aan dat Apache Drill beter presteert wanneer er meer RAM geheugen beschikbaar is.

Wel blijft het onduidelijk waarom Cloudera Impala vergeleken met Apache Drill wel snel is bij queries met alleen JOIN operaties en waarom er een groot verschil is tussen de resultaten van Cloudera Impala en Apache Drill. Beide gebruiken namelijk dezelfde techniek. Ondanks dat ik dit graag had willen weten was het niet gelukt om dit te onderzoeken. Dit kwam mede doordat de documentatie de technische werking van deze pakketten niet uitgebreid behandelde.

11.6 Resultaten schaalbaarheid

De schaalbaarheid vond de opdrachtgever een belangrijk aspect dat onderzocht moest worden tijdens het experiment. Om dit te onderzoeken is er gekeken hoe de performance wordt beïnvloed wanneer er meer Hadoop nodes in een cluster beschikbaar zijn.

Allereerst is er gekeken naar de schaalbaarheid op de dataset met tien miljoen records. Deze grafiek staat hieronder beschreven:

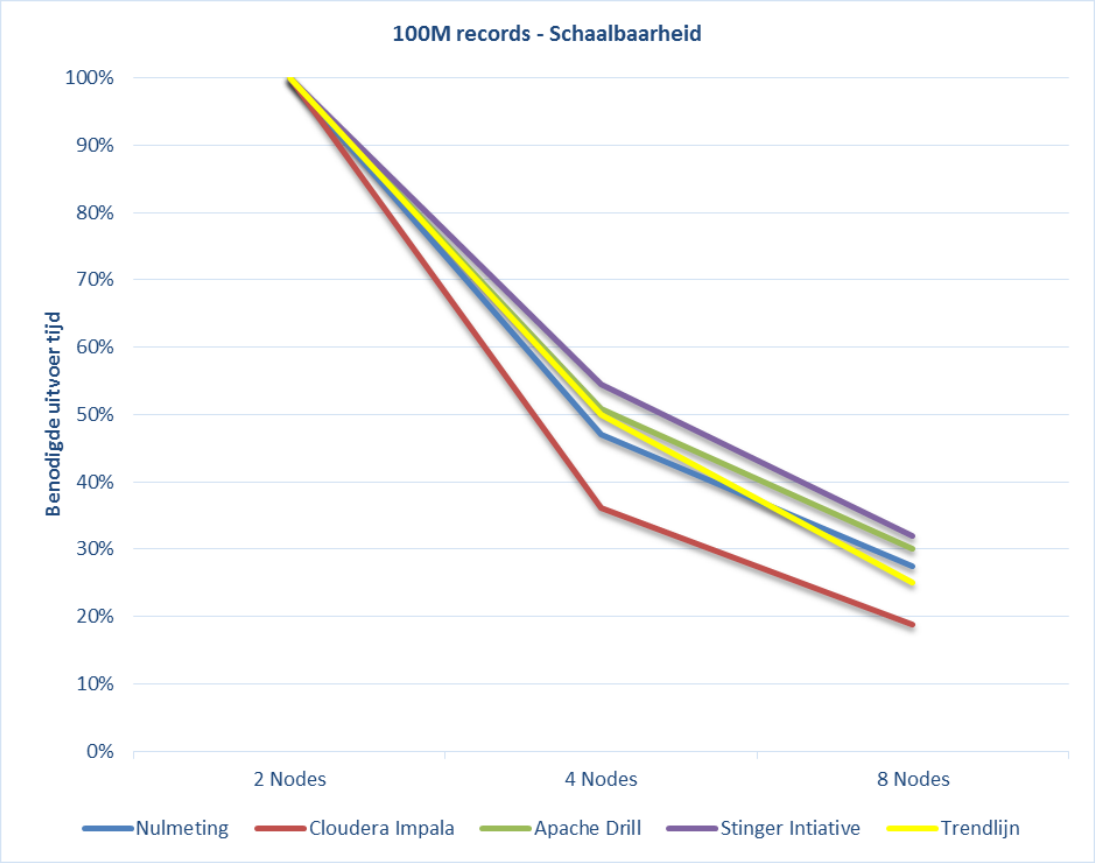


Afbeelding [27]: Schaalbaarheid tien miljoen rijen.

Zoals in afbeelding [27] te zien is, is de gele lijn in de grafiek de **trendlijn:** wanneer er een verdubbeling van het aantal nodes is, zou er in theorie een halvering van de benodigde uitvoer tijd moeten plaatsvinden. Als startpunt wordt twee nodes genomen, 100%. De trendlijn zit bij vier nodes op 50% van de benodigde tijd vergeleken met het startpunt. Wanneer er weer een verdubbeling plaatsvindt van het aantal nodes, acht nodes, zou er weer een halvering moeten plaatsvinden van de benodigde uitvoer tijd. De trendlijn zit dan bij acht nodes op 25%.

Cloudera Impala volgt de trendlijn tot vier nodes bijna perfect. De andere pakketten zitten verder boven de trendlijn. Wanneer er wordt gekeken bij acht nodes is te zien dat Cloudera Impala het dichtste in de buurt zit van de trendlijn vergeleken met de andere pakketten. Toch is er wel een groter verschil ontstaan tussen de trendlijn en de pakketten. Dit duidt erop dat de schaalbaarheid op tien miljoen rijen niet optimaal is. Zoals al eerder beschreven zou dit kunnen komen doordat de hoeveelheid data bij tien miljoen rijen te weinig was om alle nodes optimaal te benutten.

Vervolgens is er gekeken naar de schaalbaarheid op de dataset met honderd miljoen records. Deze grafiek staat hieronder beschreven:



Afbeelding [28]: Schaalbaarheid honderd miljoen records.

Zoals in afbeelding [28] te zien is, is de gele lijn in de grafiek de trendlijn. Ook hierbij is het startpunt twee nodes. Wanneer er wordt gekeken naar de trendlijn tot de vier nodes is er een duidelijk verschil zichtbaar. Apache Drill en Stinger Initiative zitten bijna op de trendlijn. De nulmeting en Cloudera Impala zitten zelfs onder de trendlijn. Cloudera Impala zit behoorlijk ver onder de trendlijn. Wanneer er wordt gekeken naar de trendlijn op acht nodes, zit de nulmeting er nu net boven. Samen met Apache Drill en Stinger Initiative zitten deze niet ver boven de trendlijn. Ook hierbij zit Cloudera Impala nog steeds onder de trendlijn.

11.7 Definitieve keuze

In de vorige paragrafen zijn de resultaten van het experiment behandeld. In deze paragraaf wordt met behulp van deze resultaten een conclusie geschreven die antwoord moet geven op de hoofdvraag:

Welke van de bestaande 'SQL-on-Hadoop' pakketten maakt Hadoop het meest geschikt om ad-hoc queries op uit te voeren?

Voordat de hoofdvraag wordt beantwoord, wordt er eerst ingegaan op de verschillende deelvragen.

11.7.1 Gemiddelde ad-hoc query performance

Aan de hand van de resultaten kan worden geconcludeerd dat Cloudera Impala gemiddeld de snelste uitvoer heeft voor ad-hoc queries. Dit geld zowel voor de dataset van tien miljoen records en honderd miljoen records.

Wat verder opvalt, is de verandering van de tweede en derde plaats. Bij tien miljoen rijen is Apache Drill tweede en Stinger Initiative derde. Bij honderd miljoen rijen is dit omgedraaid. Dit komt, zoals eerder beschreven, doordat Apache Drill bijna alles in-memory uitvoert [56]. Wanneer het uitvoeren van een query het maximaal beschikbare RAM geheugen in beslag neemt, wordt de harde schijf gebruikt voor 'virtual memory'[56]. Dit gaat alleen veel langzamer vergeleken met RAM geheugen, omdat de lees en schrijf snelheid van een harde schijf lager ligt dan dat van RAM geheugen.

11.7.2 Snelste ad-hoc query performance

Aan de hand van de resultaten kan er worden geconcludeerd dat Cloudera Impala het snelste pakket is. Dit was voor alle metingen met de dataset van tien miljoen en honderd miljoen records bij elke query het geval. Daarnaast is het ook interessant om te kijken welke queries het snelste of het langzaamste waren tijdens de nulmeting. Hieronder in de tabellen wordt dit behandeld:

Plaats	Query:
1	9
2	18
3	15, 10

Tabel [20]: Overzicht langzaamste queries.

Plaats	Query:
1	21
2	12, 4, 7
3	11

Tabel [21]: Overzicht snelste queries.

De langzaamste queries hebben de volgende eigenschappen gemeen:

1. Bevatten allemaal twee of meer aggregate functies, zoals SUM, AVG, MAX, COUNT, ABS.
2. Twee van de vijf queries hebben een JOIN of subquery.
3. Eén query bevat twee UNION ALL statements.
4. Vier van de vijf queries bevat een GROUP BY.

De snelste queries hebben de volgende eigenschappen gemeen:

1. Bijna geen aggregate functies, één query bevat maar één COUNT functie.
2. Twee van de vier queries bevatten een SELECT *.
3. Bevatten geen JOIN of subqueries.
4. Bevatten geen GROUP BY statements.

11.7.3 Schaalbaarheid

Aan de hand van de resultaten is duidelijk te concluderen dat Cloudera Impala het pakket is met de beste schaalbaarheid. Cloudera Impala volgt de trendlijn bij tien miljoen records tot vier nodes bijna perfect. Bij de andere pakketten is dit niet het geval. Bij de acht nodes zit geen enkel pakket op de trendlijn. Ook hierbij is Cloudera Impala wel het dichtste in de buurt van de trendlijn.

Bij de honderd miljoen records was Cloudera Impala wederom het pakket met de beste schaalbaarheid. Hier was zelfs het geval dat Cloudera Impala onder de trendlijn zat bij vier en acht nodes. Daarnaast geldt voor elk pakket dat de schaalbaarheid beter was wanneer er een query op een grotere dataset werd uitgevoerd. Dit was zichtbaar bij de dataset van honderd miljoen records.

11.7.4 Hoofdvraag

Nu de deelvragen zijn beantwoord kan er antwoord worden gegeven op de hoofdvraag.

Welke van de bestaande 'SQL-on-Hadoop' pakketten maakt Hadoop het meest geschikt om ad-hoc queries op uit te voeren?

Het meest geschikte ´SQL-on-Hadoop' pakket is: Cloudera Impala. Dit wordt bevestigd door de verschillende resultaten van het experiment. Hierbij heeft Cloudera Impala voor elk onderdeel het beste gescoord. Naast het experiment wordt er ook gekeken naar de technische beperkingen. Hierbij werd duidelijk dat Cloudera Impala geen subqueries ondersteunt en ook geen ORDER BY. Dit kwam doordat het experiment met een oudere versie van Cloudera Impala is uitgevoerd. In de nieuwere versies zijn deze problemen opgelost. Als er verder wordt gekeken naar de technische beperkingen heeft Apache Drill de meeste beperkingen, terwijl tijdens het experiment de meest recente versie is gebruikt.

12. Demo

Nadat het experiment is afgerond is het beste pakket bekend, namelijk Cloudera Impala. Cloudera Impala wordt vervolgens gebruikt tijdens de demo. De uitwerking van deze demo wordt beschreven in dit hoofdstuk.

De demo is eigenlijk soort Proof of Concept. Dit was door een miscommunicatie met de opdrachtgever door mij verandert in het afstudeerplan. Maar vanwege het feit dat deze naam in het afstudeerplan en PID stond is ervoor gekozen om de naam demo te behouden. In dit hoofdstuk wordt het doel van de demo beschreven. Verder wordt het functioneel en technisch ontwerp behandeld dat is gemaakt voor de demo.

12.1 Doel

Het doel van de demo is om te bekijken of het mogelijk is om als eindgebruiker op een gebruiksvriendelijke manier met de data te interacteren die is opgeslagen in HDFS. Via Cloudera Impala wordt deze data opgehaald uit HDFS. Het gebruiksvriendelijk interacteren met de data wordt gerealiseerd door middel van verschillende Business Intelligence visualisatie tools. Tijdens de demo wordt er gebruik gemaakt van Tableau en Microsoft Excel Powerpivot als visualisatie tool. De grote concurrent van Tableau is Qlikview. De reden dat er voor Tableau is gekozen, komt vanwege de ingebouwde connectie voor Cloudera Impala op Amazon EMR. Met behulp van deze connectie is het inladen van de data in Tableau zeer eenvoudig. Qlikview heeft geen ingebouwde ondersteuning voor Cloudera Impala op Amazon EMR.

Microsoft Excel Powerpivot is gekozen omdat de opdrachtgever dit graag wilde. De reden hiervoor was dat Excel een veel gebruikte tool is om rapportages te maken en analyses te doen op data. Daarnaast heeft Excel geen standaard connectie ingebouwd voor Cloudera Impala en moet dit worden geregeld via een ODBC connectie.

In principe is elke visualisatie tool mogelijk, zolang er in de tool ondersteuning is voor bijvoorbeeld een ODBC connectie, of een directe verbinding met Cloudera Impala. De ODBC connectie wordt beschreven in bijlage L *'Technisch ontwerp'* hoofdstuk *'6. Cloudera Impala ODBC*'.

12.1 Functioneel ontwerp

In Het functioneel ontwerp worden de functionele aspecten die nodig zijn om de demo te maken behandeld. Hiervoor zijn allereerst requirements geïnventariseerd.

12.1.1 Requirements

De opdrachtgever had van te voren al een duidelijk beeld welke vraagstukken de demo moest beantwoorden. Omdat het maar om een beperkt aantal vraagstukken ging, was het inventariseren hiervan eenvoudig. Deze vraagstukken zijn vervolgens SMART herschreven. Een voorbeeld van zo'n vraagstuk is: '*Kunnen rapporten eenvoudig worden gevonden, gedeeld?*'. Uit dit vraagstuk zijn de volgende twee requirements gekomen die in tabel [22] staan beschreven.

ID	Beschrijving
UR7	Het systeem moet de mogelijkheid hebben om rapporten te kunnen delen.
UR8	Het systeem moet de mogelijkheid hebben om rapporten op te zoeken.

Tabel [22]: Overzicht requirements bij vraagstuk.

Naast dat de requirements SMART zijn geformuleerd is er ook aangegeven of het om functionele of niet-functionele requirements ging. In totaal waren er dertien functionele requirements en één niet-functionele requirement. De SMART herschreven requirements zijn vervolgens weer teruggekoppeld aan de opdrachtgever. Hierbij werd gevraagd of de opdrachtgever akkoord ging met deze herschreven requirements en of er nog requirements ontbraken. De herschreven requirements waren vervolgens goedgekeurd door de opdrachtgever en waren ook compleet. Voor het volledige overzicht van alle requirements kan er worden gekeken in bijlage K *'Functioneel ontwerp'* hoofdstuk *'2. Requirements'*.

12.1.2 Use cases

Nadat de requirements duidelijk waren is er verder gegaan met het maken van een use case diagram en de uitwerking hiervan in use case beschrijvingen. Als input voor de use cases worden de requirements gebruikt. De reden dat er use cases zijn gemaakt is om een beter inzicht te krijgen welke functionaliteit er tijdens de demo nodig is. Zo is bijvoorbeeld de requirement '*Het systeem moet de mogelijkheid hebben om rapporten op te zoeken*' op de volgende manier verwerkt in een use case beschrijving.

Use case name	Rapport zoeken
Id	UC3
Summary	De Gebruiker zoekt een rapport.
Primary actors	Gebruiker (actor 1)
Secondary actors	-
Preconditions	Het systeem is opgestart.
Main Flow	Actor 1 vult zoekcriteria in. Systeem toont resultaten die overeenkomen met de zoekcriteria [1].
Post conditions	Er is een rapport gezocht.
Alternative Flows	[1] Het systeem toont een melding dat er geen resultaten zijn gevonden.
Requirements	UR8

Tabel [23]: Use case beschrijving.

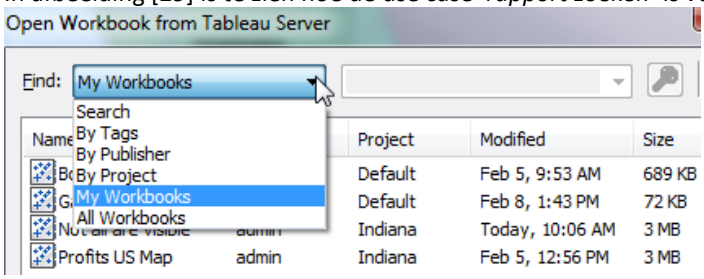
De use case beschrijving in tabel [23] is één van de in totaal acht use case beschrijvingen. De overige use case beschrijvingen, inclusief het use case diagram zijn te vinden in bijlage K *'Functioneel ontwerp'* hoofdstuk *'2. Use case diagram'*. De meeste use case beschrijvingen zijn redelijk basaal, maar zijn alsnog opgenomen omdat deze namelijk wel functionaliteit beschrijven die nodig is voor de demo.

12.1.3 Schermdiagrammen

Nadat de use cases waren beschreven is er verder gegaan met het maken van schermdiagrammen. Omdat er gebruik wordt gemaakt van bestaande visualisatie tools, zijn de schermdiagrammen bijvoorbeeld geen mock-ups geworden, maar screenshots die bepaalde functionaliteit laten zien in de visualisatie tools. De schermdiagrammen zijn opgenomen in het functioneel ontwerp, zodat er documentatie beschikbaar is waarin wordt beschreven hoe Tableau en Microsoft Excel Powerpivot de verschillende vraagstukken en requirements beantwoorden. Verder kan

er dankzij de schermdiagrammen worden aangetoond dat het doel van de demo is bereikt: als eindgebruiker op een gebruiksvriendelijke manier met de data te interacteren die is opgeslagen in HDFS.

In afbeelding [29] is te zien hoe de use case '*rapport zoeken'* is verwerkt in een schermdiagram in Tableau.



Afbeelding [29]: Schermdiagram rapport zoeken.

In bijlage K *'Functioneel ontwerp*' hoofdstuk *'5. Schermdiagrammen'* zijn alle schermdiagrammen te vinden.

Naast de schermdiagrammen die opgenomen zijn in het Functioneel ontwerp is er ook nog een live demo gegeven aan de opdrachtgever. Hierbij zijn de functionaliteiten van Tableau en Microsoft Excel Powerpivot behandeld.

12.1.4 Conclusie

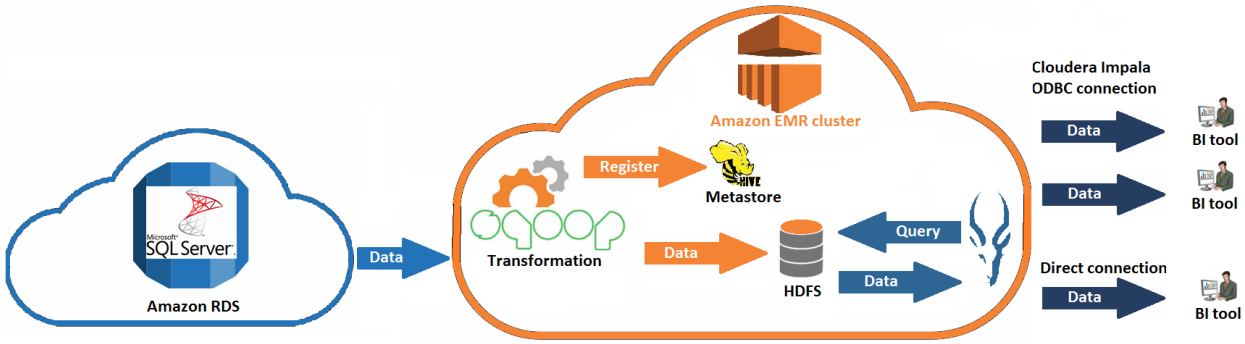
Tot slot is er in het Functioneel Ontwerp een conclusie geschreven. In deze conclusie wordt beschreven hoe de schermdiagrammen de verschillende requirements en vraagstukken beantwoorden. Daarna wordt beschreven dat het doel van de demo, om via Cloudera Impala de data op een gebruiksvriendelijke manier te tonen aan de eindgebruiker, behaald is. De complete conclusie kan worden gevonden in bijlage K *'Functioneel ontwerp*' hoofdstuk *'6. Conclusie'*

12.2 Technisch ontwerp

In een technisch ontwerp worden bijvoorbeeld klassendiagrammen of sequencediagrammen behandeld. Omdat er met bestaande software werd gewerkt waren deze stappen niet nodig. Daarom is ervoor gekozen om in het technisch ontwerp alle technische keuzes te beschrijven die zijn genomen tijdens de demo. Een aantal van deze keuzes zijn al eerder in dit verslag beschreven. Een voorbeeld hiervan is hoe de gegevensconversie is uitgevoerd. Het technisch ontwerp functioneert meer als een handleiding hoe een Amazon EMR cluster opgezet kan worden, of welke configuratie nodig is om de data in Tableau of Microsoft Excel Powerpivot te laden. Het document is te vinden in bijlage L '*Technisch ontwerp*'.

12.2.1 Architectuur

Een belangrijk onderdeel van het Technisch Ontwerp is de architectuur van de demo. Hierbij wordt getoond dat in principe elke Business Intelligence visualisatie tool gebruikt kan worden, zolang er een ondersteuning is voor een ODBC connectie, of een directe connectie met Cloudera Impala. Cloudera Impala is vervolgens verantwoordelijk voor het ophalen van de data uit HDFS via een query. De data wordt vervolgens weer teruggestuurd naar de desbetreffende visualisatie tool. In afbeelding [30] is deze architectuur geïllustreerd.



Afbeelding [30]: Architectuur demo.

13. Project afsluiting

Nadat de demo was afgerond en goedgekeurd is het project vervolgens afgesloten. In PRINCE2 is hier een aparte fase voor, namelijk 'Closing a project'. Hierbij worden de documenten die tijdens het project zijn gemaakt overgedragen aan de opdrachtgever en wordt het project definitief beëindigd. In dit hoofdstuk wordt het adviesrapport beschreven dat is gemaakt voor Info Support. Het complete adviesrapport is te vinden in bijlage D *'Adviesrapport*'.

13.1 Advies

Op basis van de resultaten kan er een advies worden gegeven. Hierbij wordt gekeken naar de hoofdvraag die tijdens het onderzoek is opgesteld.

Welke van de bestaande 'SQL-on-Hadoop' pakketten maakt Hadoop het meest geschikt om ad-hoc queries op uit te voeren?

Het antwoord op de hoofdvraag luidt als volgt: Cloudera Impala is het meest geschikte pakket om ad-hoc queries op Hadoop uit te voeren.

Dit is vanwege de volgende punten:

1. De resultaten uit de verschillende onderdelen van het experiment tonen aan dat Cloudera Impala overal het beste op gescoord heeft vergeleken met de andere pakketten.
2. De SQL beperkingen van de oude versie van Cloudera Impala zijn in nieuwere versies opgelost.
3. De 'out-of-the-box' configuratie op Amazon EMR was eenvoudig te gebruiken.
4. Cloudera Impala is ook op Microsoft Azure Marketplace 'out-of-the-box' beschikbaar.
5. Cloudera Impala is open-source. Dit is voor de opdrachtgever een groot pluspunt.
6. Cloudera is een van de marktleiders samen met Hortonworks en MapR op het gebied van Hadoop distributies. Cloudera was het eerste bedrijf van deze drie dat zich bezig hield met Apache Hadoop en heeft ook het grootste aantal gebruikers [59].

Wel zijn er een aantal punten waar rekening mee moet worden gehouden wanneer Cloudera Impala wordt gebruikt. Zo is Cloudera Impala open-source, maar op het moment dat er ondersteuning nodig is vanuit Cloudera bij bijvoorbeeld de installatie van Impala of een ander product van Cloudera, moet er worden betaald voor de support. De prijzen voor deze ondersteuning zijn niet duidelijk omschreven. Ook moet het echt noodzakelijk om over te stappen naar een 'SQL-on-Hadoop' oplossing. Wanneer bij bijvoorbeeld een klant de huidige technologie niet meer voldoende is om de grote hoeveelheid data aan te kunnen, zou een overstap naar een 'SQL-on-Hadoop' oplossing mogelijk zijn. Wanneer de overstap wordt gedaan zonder duidelijke reden is het vaak niet noodzakelijk.

Tot slot is het ook belangrijk om rekening te houden dat de 'SQL-on-Hadoop' markt snel kan veranderen. In oktober 2010 was de eerste versie van Apache Hive uitgebracht. Dit was de eerste 'SQL-on-Hadoop' oplossing. Ruim vier jaar later, zijn er negentien 'SQL-on-Hadoop' pakketten beschikbaar en dit aantal zal alleen nog maar toenemen. Cloudera zal proberen haar sterke positie in deze snel veranderende markt te behouden. Daarom ben ik van mening dat Impala voorlopig zal worden doorontwikkeld. Toch is het belangrijk om de ontwikkelingen binnen 'SQL-on-Hadoop' oplossingen in de gaten te houden. Andere, of toekomstige SQL-on-Hadoop oplossingen zouden wellicht over een aantal jaar sneller kunnen zijn.

14. Evaluatie

In dit hoofdstuk wordt de evaluatie over dit project behandeld. Hierbij wordt ingegaan op de procesevaluatie, de productevaluatie en de behaalde beroepstaken. Verder wordt ook beschreven hoe ik het als afstudeerder binnen Info Support heb ervaren.

14.1 Procesevaluatie

14.1.1 Projectmanagement: PRINCE2

PRINCE2 is als projectmanagement methodiek gekozen voor dit project en is voornamelijk gebruikt in het begin van het project en op het einde. Helaas zijn niet alle onderdelen van de methode gebruikt. Zo kon de sturing tijdens het project via PRINCE2 beter. Een voorbeeld hiervan was dat er sneller moest worden aangekaart dat er uitloop was, waardoor er misschien minder vertraging was opgelopen. Ook is er niet formeel aan het einde van elke fase een ´fase-einde rapport´ opgesteld en is dit niet altijd bij de opdrachtgever aangekaart. Voor een volgende keer zou dit zeker beter kunnen, door duidelijk de fasen af te sluiten. Daarom is er eerder sprake van Prince In Name Only, dan van een volwaardige implementatie van de methode. Ondanks de niet volledige implementatie van PRINCE2 heeft het er uiteindelijk wel voor gezorgd dat alle betrokkenen tijdens dit project wisten waar ze aan toe waren en wat er werd uitgevoerd. Daarnaast heeft het mij meer inzicht gegeven hoe PRINCE2 een volgende keer beter gebruikt zou kunnen worden. Daarom zou ik het een volgende keer weer gebruiken.

14.1.2 Pakketselectie: KPMG

KPMG is tijdens de pakketselectie gebruikt als methode, samen met de selectiecriteria lijst van Indora. Dankzij deze combinatie was er al van te voren een grote hoeveelheid aan voorbeeld selectiecriteria beschikbaar en heeft ervoor gezorgd dat de pakketselectie op een gestructureerde manier is uitgevoerd. Tijdens het inventariseren van de requirements voor de longlist had ik moeten vragen of er nog meer stakeholders waren. Ik had namelijk aangenomen dat de opdrachtgever de enige stakeholder was. Dit had ik later nog gevraagd en dit bleek zo te zijn. Maar voor een volgende keer is dit beter om van tevoren te vragen. Wanneer er dan nog andere stakeholders zijn kan hier rekening mee worden gehouden. Verder hadden er duidelijkere grenzen moeten worden gesteld aan het aantal ´KeyCriteria´ dat voortkwam uit de requirements. Het aantal KeyCriteria was namelijk behoorlijk groot. Als van tevoren duidelijk was gemaakt richting de opdrachtgever dat er bijvoorbeeld maximaal vijf KeyCriteria gebruikt mochten worden zou de lijst van KeyCriteria er anders uit hebben gezien.

Uiteindelijk is niet alles uit de methode toegepast. Een voorbeeld hiervan is de laatste stap, het contracteren. Dit kwam omdat het uiteindelijke pakket namelijk open-source was. Desondanks zou ik voor een volgende keer wel weer de KPMG pakketselectie methode in combinatie met de selectiecriteria van Indora gebruiken. De voornaamste reden hiervoor is dat er namelijk een weloverwogen keuze kan worden gemaakt met behulp van een pakketselectie methode.

14.1.3 Onderzoek

Het onderzoek is het onderdeel dat het meeste tijd heeft gekost in dit project. Over het proces zelf ben ik behoorlijk tevreden hoe dit is aangepakt, voornamelijk hoe het onderzoek zelf is opgezet. Door eerst een literatuuronderzoek uit te voeren en daarna pas te beginnen aan het experiment was er een goede technische basis gelegd die van pas kwam tijdens het experiment. Het onderzoek zou ik voor een volgende keer op dezelfde manier aanpakken. Wel is er vertraging opgelopen tijdens de onderzoeksfase.

De eerste vertraging ontstond bij het regelen van hardware. Om ervaring met de pakketten op te doen, voor het experiment, was er een krachtigere computer nodig. Deze had ik pas aangevraagd toen ik de computer nodig had. Aangezien dit via verschillende personen moet worden geregeld neemt dit toch al gauw een aantal dagen in beslag. Voor een volgende keer zou er vooraf beter moeten worden bekeken wat er precies nodig is.

De tweede vertraging ontstond bij het installeren van de pakketten op een lokale machine. Omdat ik met Hadoop en Linux aan de slag moest, waar ik geen ervaring mee had, heb ik meerdere keren vast gezeten op foutmeldingen of problemen waarvan ik geen idee had hoe ze opgelost moesten worden. Hierbij had ik actiever hulp moeten zoeken, in plaats van te lang proberen het zelf op te lossen. Daarnaast had ik vooraf beter moeten bekijken wat er allemaal nodig is. De pakketten moesten namelijk uiteindelijk op een cloud omgeving worden geïnstalleerd. Hier bleek al dat twee pakketten ‘out-of-the-box’ beschikbaar waren, waarbij de installatie zeer eenvoudig is.

14.1.4 Testen: TMAP Next

Met behulp van TMAP Next is de gegevensconversie getest. In het document dat hiervoor beschikbaar was werd duidelijk beschreven welke kwaliteitsattributen moesten worden getest, namelijk ‘juistheid’ en ‘volledigheid’. Het nadeel was wel dat hier niet duidelijk in werd verteld hoe dit getest moest worden. Dit heb ik uiteindelijk opgelost door aan de opdrachtgever en technisch begeleider te vragen welke testen hiervoor nodig zijn. Het uitvoeren van de testen zelf was eenvoudig en heeft ook niet veel tijd in beslag genomen. Wel heeft het ervoor gezorgd dat er een fout in de gegevens conversie gevonden is. Voor een volgende keer zou ik de testen op dezelfde manier uitvoeren. Voornaamste punt is hierbij dat TMAP Next de enige methode is die beschrijft hoe een gegevensconversie getest moet worden.

14.1.5 Info Support

Binnen Info Support is er een goede begeleiding. Zo is er een proces begeleider, die het proces in de gaten houdt. Een technisch begeleider, waar alle technische vragen aan kan stellen en een opdrachtgever. De communicatie met beide begeleiders en de opdrachtgever heb ik als prettig ervaren. Er werd snel geantwoord op vragen via de email en de feedback op documenten was bruikbaar. Zelf heb ik vier dagen in de week in Zoetermeer gezeten en één dag in Veenendaal. Ondanks dat je de begeleiders één keer in de twee weken ziet en de opdrachtgever één keer in de week heb ik dit niet als vervelend beschouwd. Hierdoor word je juist uitgedaagd om niet voor elk klein probleem direct een vraag te stellen, maar eerst het proberen zelf op te lossen. Daarnaast heerst er een professionele sfeer en zijn mensen altijd wel bereid om je te helpen en wordt er veel gedaan aan kennisdeling. Zo waren er verschillende presentaties gegeven tijdens de afstudeerperiode over bijvoorbeeld het opstellen van een Plan van Aanpak, of het schrijven van een afstudeerverslag.

14.2 Productevaluatie

14.2.1 PID

Over het PID ben ik tevreden hoe dit tot stand gekomen is. Voordat ik begon met afstuderen heb ik de afstudeeropdracht meerdere keren moeten herschrijven. Daardoor had ik een duidelijk beeld wat er tijdens het afstuderen gedaan moest worden. Dit heb ik mee kunnen nemen tijdens het schrijven van het PID. Wel was er een miscommunicatie geweest tussen mij en de opdrachtgever over de demo tijdens het schrijven van het afstudeerplan en PID. De opdrachtgever wilde namelijk een Proof of Concept in plaats van een demo. Uiteindelijk is dit opgelost door het wel bij een demo te houden, die verschillende vraagstukken moet beantwoorden. Dit had voorkomen kunnen worden door de laatste versie van het afstudeerplan nog terug te koppelen naar de opdrachtgever.

In het PID wordt de projectaanpak beschreven, de planning en fasering, de scope en de op te leveren producten beschreven. Dankzij het PID was het voor de opdrachtgever duidelijk wat er tijdens deze afstudeerperiode uitgevoerd zou worden. Het opstellen van de risico's had ik voornamelijk gekeken naar risico's waar ik zelf centraal sta. Maar bijvoorbeeld organisatorische risico's was ik vergeten hierin op te nemen. Dit zou ik voor een volgende keer anders doen. Verder zou ik het opstellen van een PID op dezelfde manier doen.

14.2.2 Requirements rapport

Het requirements rapport ben ik tevreden over. Hier staan alle requirements in, met een omschrijving en de prioritering. Ook wordt hierin beschreven hoe de requirements tot stand zijn gekomen. Het opstellen van dit rapport zou ik voor een volgende keer op dezelfde manier doen.

14.2.3 Pakketselectie document

Dit document beschrijft hoe de longlist is aangepakt. Hierbij wordt ingegaan op de verschillende selectiecriteria. Het definiëren van de belangrijkste selectiecriteria, de 'KeyCriteria' is niet duidelijk beschreven in dit document. Voor een volgende keer zou ik duidelijker beschrijven hoe deze KeyCriteria zijn gedefinieerd. Verder ben ik wel tevreden over de beschrijving van de tekortkomingen per pakket en hoe de selectiematrix is opgesteld. Voor een volgende keer zou ik deze selectiematrix zeker weer gebruiken, omdat dit namelijk de opdrachtgever direct een overzicht geeft hoe pakketten ten opzichte van elkaar scoren.

14.2.4 Onderzoeksdocument

Het onderzoeksdocument is het grootste document dat is opgeleverd. Over het document zelf ben ik zeer tevreden. Uiteraard zijn er wel een paar onderdelen die beter hadden gekund. Zo was er geen duidelijke criteria waar de deelvragen aan moesten voldoen. Voor een volgende keer zou bijvoorbeeld de checklist in het boek '*Wat is onderzoek*' [58] gebruikt kunnen worden voor het definiëren van de deelvragen. Deze checklist had ik namelijk niet gebruikt voor het opstellen van de deelvragen. Verder heb ik de deelvraag van de technische werking van de pakketten niet helemaal goed kunnen uitwerken. Waarom een pakket sneller is vergeleken met een ander pakket is hierdoor maar gedeeltelijk beantwoord. Dit kwam doordat er niet veel documentatie beschikbaar was over de technische werking van de pakketten. Wanneer ik vooraf had bekeken welke deelvragen er tijdens het experiment zouden worden behandeld, had ik rekening kunnen houden hoe de pakketten de queries uitvoeren. Dit zou

bijvoorbeeld via de query planner of optimizer kunnen worden bekeken. Dit zou ik voor een volgende keer zeker doen.

14.2.5 Adviesrapport

Het adviesrapport is een beknopte weergave van de resultaten die voortkomen uit het onderzoeksdocument. Over het adviesrapport zelf ben ik tevreden, omdat er namelijk een duidelijk advies in staat beschreven welk SQL-on-Hadoop pakket het meest geschikt is voor Info Support en waar nog meer rekening mee moet worden gehouden. Voor een volgende keer zou ik dit weer op dezelfde manier doen.

14.2.6 Demo

Voor de demo is er een functioneel en technisch ontwerp gemaakt. Over het resultaat van de demo, inclusief de documenten ben ik tevreden. Het doel van de demo was om aan te kunnen tonen dat de eindgebruiker op een gebruiksvriendelijke manier met de data kan interacteren. Het behaalde doel wordt voornamelijk beschreven in het functioneel ontwerp. Hier zijn requirements, use case beschrijvingen en schermdiagrammen voor gemaakt. Voor een volgende keer zou ik dit weer doen, wanneer er een demo met bestaande software moet worden gemaakt. Wel zijn de use case beschrijvingen redelijk basaal.

Naast het functioneel ontwerp is er nog het technisch ontwerp. Omdat er een mogelijkheid is dat Cloudera Impala gebruikt wordt binnen Info Support, is er nu al gedocumenteerd hoe dit op Amazon EMR gebruikt kan worden. Ook wordt hierin beschreven welke technische stappen er nodig zijn om de rapportages uit te kunnen voeren.

14.2.7 Testrapport

Over het testrapport ben ik tevreden hoe dit is opgesteld. Er wordt namelijk behandeld welke kwaliteitsattributen worden getest en hoe deze worden getest. Vervolgens werd per test beschreven hoe dit is aangepakt. Tot slot zijn alle resultaten van de testen zichtbaar en wordt er een conclusie getrokken aan de hand van de resultaten. Voor een volgende keer zou ik dit zeker op dezelfde manier doen.

14.2.8 Doel afstuderen

Het doel van het afstuderen was om een onderzoek voor Info Support uit te voeren naar het pakket dat het meest geschikt is om ad-hoc queries uit te voeren binnen Hadoop. Ik ben van mening dat dit doel is behaald. Door eerst requirements op te stellen, die als input dienen voor de selectiecriteria voor de longlist, is het aanbod SQL-on-Hadoop pakketten teruggebracht naar drie pakketten. Deze drie pakketten zijn op de shortlist terecht gekomen. Vervolgens is het experiment uitgevoerd voor de pakketten op de shortlist. Het voorbereidende werk voorafgaand aan het experiment, zoals het definiëren van een ad-hoc query set, of de dataset, hebben ertoe geleid dat het experiment op een goede manier kon worden uitgevoerd. Al deze stappen zijn beschreven in verschillende documenten. Uiteindelijk is Cloudera Impala aan de hand van de resultaten en in overleg met de opdrachtgever als beste pakket gekozen om ad-hoc queries te versnellen op Hadoop. Ondanks de verbeteringen die in de evaluatie zijn beschreven vind ik dat het uiteindelijke resultaat zeer bruikbaar is voor Info Support.

14.3 Beroepstaken

In deze paragraaf worden de behaalde beroepstaken behandeld. Hierbij wordt ingegaan op de volgenden punten:

1. Wat is er uitgevoerd om de beroepstaak te behalen?
2. Wat was het resultaat?
3. Is de beroepstaak behaald?

14.3.1 Selecteren methoden, technieken en tools

Om deze beroepstaak te behalen zijn de volgende methodieken geselecteerd tijdens het afstuderen:

1. PRINCE2
2. KPMG
3. TMAP Next

Daarnaast is er gebruik gemaakt van de volgende tools:

1. SQL Data Generator
2. SQOOP
3. Tableau
4. Microsoft Excel Powerpivot

PRINCE2 is als projectmanagement methodiek gebruikt. Hierdoor is het project op een gestructureerde manier aangepakt. Zo is er conform PRINCE2 een PID opgesteld en zijn er meerdere fasen doorlopen van PRINCE2 (hoofdstuk *'5.3 Fasering*'). Met behulp van KPMG is er een pakketselectie uitgevoerd. Hierbij zijn er eerst requirements en selectiecriteria opgesteld (hoofdstuk *'7.2 Aanpak*'). Vervolgens zijn deze selectiecriteria voor de longlist gebruikt (hoofdstuk *'8. Longlist*'). De beste pakketten op de longlist zijn vervolgens op de shortlist terechtgekomen. Aan de hand van een experiment voor de pakketten op de shortlist is het beste pakket geselecteerd (hoofdstuk *'11.7 Definitieve keuze*'). Hier is een adviesrapport voor geschreven. Het testen van de gegevensconversie is gedaan volgens TMAP Next (hoofdstuk *'10. Testen*').

Naast de methodieken zijn er ook tools gebruikt. Zo is de dataset uitgebreid met behulp van SQL Data Generator (hoofdstuk *'9.1 Dataset*'). De gegevensconversie is uitgevoerd met behulp van SQOOP (hoofdstuk *'9.2 SQOOP*'). Voor de demo is er gebruik gemaakt van Tableau en Microsoft Excel Powerpivot (hoofdstuk *'12. Demo'*). De keuzes voor deze tools worden onderbouwd in de hoofdstukken waarin de tools worden behandeld.

Met de bovenstaande resultaten en de onderbouwing die is beschreven in de verschillende hoofdstukken waarom er gebruik wordt gemaakt van de methoden en tools ben ik van mening dat deze beroepstaak behaald is.

14.3.2 Selecteren van standaardsoftware

Om deze beroepstaak te behalen is er een pakketselectie uitgevoerd volgens de KPMG pakketselectie methodiek. Hier is allereerst een set van requirements opgesteld. Deze requirements zijn vervolgens gebruikt voor de selectiecriteria. Tijdens de longlist worden de pakketten beoordeeld aan de hand van de selectiecriteria (hoofdstuk *8.2 Selectiecriteria*'). De beste drie pakketten zijn op de shortlist terechtgekomen. Voor de pakketten op de shortlist is er een experiment uitgevoerd (hoofdstuk *'11. Experiment & shortlist'*). Aan de hand van de resultaten van het experiment is het beste pakket geselecteerd, namelijk Cloudera Impala. De keuze voor het beste pakket is ook gedocumenteerd in een adviesrapport. Daarnaast zijn alle stappen gedocumenteerd. Ondanks dat de hoeveelheid KeyCriteria wellicht te groot was voor de longlist ben ik wel van mening dat ik deze beroepstaak behaald heb.

14.3.3 Uitvoeren analyse door definitie van requirements

Om deze beroepstaak te behalen zijn er meerdere sessies geweest om de requirements voor de longlist te inventariseren. Hierbij is tijdens de eerste sessie een groot gedeelte van de requirements geïnventariseerd (hoofdstuk *'7.2.1 Eerste requirements inventarisatie sessie'*). In de tweede sessie zijn er nog een aantal requirements naar boven gehaald en is er een prioritering aan de requirements gegeven (hoofdstuk *'7.2.2 Tweede requirements inventarisatie sessie').*

Het resultaat was een lijst van SMART geformuleerde en geprioriteerde requirements waar de opdrachtgever akkoord mee was. Deze lijst heeft vervolgens als input gediend voor de selectiecriteria voor de longlist.

Ondanks dat er maar één stakeholder was ben ik wel van mening dat ik met de bovenstaande beschreven stappen deze beroepstaak heb behaald.

14.3.4 Uitvoeren gegevensconversie

Voor het behalen van deze beroepstaak is er een gegevensconversie uitgevoerd. Hiervoor is eerst de structuur van de dataset in kaart gebracht door middel van een datadictionary en een databasediagram. Daarna is er een mapping gemaakt tussen de bron data en de target data. Vervolgens is deze dataset met behulp van SQL Data Generator uitgebreid (hoofdstuk *'9.1.1 Aanpak'*). De conversie zelf is uitgevoerd met behulp van SQOOP (hoofdstuk *'9.2.3 Uitvoer conversie'*). Tijdens de conversie is er een transformatie uitgevoerd op attributen die niet in de target data worden ondersteund. Wanneer de conversie was uitgevoerd is de data uit SQL beschikbaar in HDFS.

Ondanks dat de gegevensconversie qua complexiteit niet heel hoog was, ben ik wel van mening dat ik deze beroepstaak heb behaald. Dit komt doordat er uitgebreid is gedocumenteerd hoe de gegevensconversie is uitgevoerd en ik hierbij alle stappen heb gevolgd voor een gegevensconversie.

14.3.5 Uitvoeren van en rapporteren over het testproces

Het behalen van deze beroepstaak is gedaan door middel van de volgende twee onderdelen. Als eerste is er in het onderzoeksdocument uitgebreid beschreven welke stappen er zijn genomen voor het experiment. Hierbij wordt beschreven hoe de ad-hoc query set tot stand is gekomen, welke dataset er wordt gebruikt inclusief de verantwoording en hoe het experiment is uitgevoerd (hoofdstuk *'11.3 Uitvoer'*). Daarnaast is de 'rauwe' data beschikbaar van de resultaten. Bij de resultaten kon nog wel uitgebreider worden ingegaan op de technische achtergrond waarom bepaalde pakketten nou sneller zijn. Dit is gedeeltelijk gelukt.

Verder is er een testrapport opgesteld voor de gegevensconversie test. Hierin is beschreven welke kwaliteitsattributen zijn getest en hoe deze tests zijn uitgevoerd. Tijdens het uitvoeren van de tests is er ook nog een fout gevonden in de conversie. Het nut van deze tests was dus ook aangetoond. Vervolgens zijn de resultaten van de tests opgenomen en is er een conclusie geschreven dat de gegevensconversie goed is uitgevoerd.

Ik ben van mening dat ik deze beroepstaak heb behaald. Dit komt mede doordat het experiment gedegen is beschreven en hoe de resultaten tot stand zijn gekomen. Daarnaast is er een testrapport beschikbaar voor de gegevensconversie test.

Literatuurlijst

[1]: https://www.infosupport.com/missie-en-kernwaarden/
[2]: https://www.infosupport.com/klantreferenties/
[3]: http://inteledyne.com/wp-content/uploads/2013/01/werehousing.jpg
[4]: http://www.twynstraguddekennisbank.nl/projectmanagement/methodes-vergeleken
[5]: http://www.viergever.info/nl/pmbokp2.aspx
[6]: http://www.apollo-training.com/index.php/pmbok-vs-prince2.html
[7]: http://www.compact.nl/artikelen/C-2009-1-Hofland.htm
[8]: http://www.ictaccountancy.nl/downloads/INDORA_Software_en_leverancierselectie.PDF
[9]: http://www.tmap.net/tmap-next
[10]: https://blog.udemy.com/sql-queries/
[11]: http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&appname=SWGE_IM_EZ_USEN&htmlfid=IMW14800USEN&attachment=IMW14800USEN.PDF#loaded
[12]: http://researcher.ibm.com/researcher/files/us-aflorat/BenchmarkingSQL-on-Hadoop.pdf
[13]: http://sonra.io/what-makes-mapr-superior-to-other-hadoop-distributions/
[14]: http://www.datadoghq.com/wp-content/uploads/2013/07/top_5_aws_ec2_performance_problems_ebook.pdf
[15]: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/dedicated-instance.html
[16]: http://searchbusinessanalytics.techtarget.com/feature/Selecting-the-right-SQL-on-Hadoop-engine-to-access-big-data
[17]: http://www.ictaccountancy.nl/downloads/INDORA_Software_en_leverancierselectie.PDF
[18]: https://hadoopecosystemtable.github.io/
[19]: http://www.vertica.com/tag/sql-on-hadoop/
[20]: http://svn.apache.org/viewvc/hive/branches/?sortby=date&sortdir=down#dirlist
[21]:http://www.slideshare.net/BobSloot/pakketselectie-de-juiste-keuze-tot-succes-bob-sloot-juni-2012-13338259
[22]: http://www.crmsystemen.nl/crm-selectie/pakketten-vergelijken
[23]: http://www.compact.nl/artikelen/C-2002-2-Mancham.htm
[24]: https://cwiki.apache.org/confluence/display/Hive/ViewDev
[25]: http://sqoop.apache.org/
[26]: http://flume.apache.org/
[27]: https://cwiki.apache.org/confluence/display/Hive/LanguageManual+Types
[28]: https://dev.mysql.com/doc/refman/5.0/en/numeric-type-overview.html
[29]: https://msdn.microsoft.com/en-us/library/ms177603.aspx
[30]: http://sqlblog.com/blogs/aaron_bertrand/archive/2008/04/27/performance-storage-comparisons-money-vs-decimal.aspx
[31]: http://rusanu.com/2010/03/22/performance-comparison-of-varcharmax-vs-varcharn/
[32]: http://mail-archives.apache.org/mod_mbox/sqoop-ser/201305.mbox/%3C20130510032057.GC20802@Odie%3E
[33]: http://www.tmap.net/sites/default/files/Overzicht_Toegepaste_testvormen.doc
[34]: http://researcher.ibm.com/researcher/files/us-aflorat/BenchmarkingSQL-on-Hadoop.pdf
[35]: https://developer.ibm.com/hadoop/blog/2014/12/02/big-sql-3-0-hadoop-ds-benchmark-performance-isnt-everything/
[36]: https://www.spec.org/benchmarks.html
[37]: https://stacresearch.com/who-we-are

[38]: http://www.tpc.org/tpcx-hs/results/tpcxhs_perf_results.asp
[39]: http://www.tpc.org/results/fdr/tpcxhs/cisco~tpcxhs~cisco_ucs_integrated_infrastructure_for_big_data~fdr~2015-01-08~v05.pdf
[40]: http://www.cloudera.com/content/cloudera/en/documentation/cloudera-impala/latest/topics/rg_impala_vd.html
[41]: http://drill.apache.org/docs/release-notes/
[42]: https://hive.apache.org/downloads.html
[43]: http://tez.apache.org/releases/index.html
[44]: http://www.cloudera.com/content/cloudera/en/documentation/cloudera-impala/v2-0-x/topics/impala_subqueries.html
[45]: http://www.cloudera.com/content/cloudera/en/documentation/cloudera-impala/v1/latest/Installing-and-Using-Impala/ciiu_order_by.html
[46]: https://cwiki.apache.org/confluence/display/DRILL/Data+Types
[47]: https://issues.apache.org/jira/browse/DRILL-1959
[48]: https://cwiki.apache.org/confluence/display/DRILL/SQL+Functions
[49]: https://cwiki.apache.org/confluence/display/DRILL/SELECT+Statements
[50]: http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.1.5/bk_dataintegration/content/ch_using-hive-using-subqueries.html
[51]: https://cwiki.apache.org/confluence/display/Hive/LanguageManual+SubQueries
[52]: http://blog.cloudera.com/blog/2012/10/cloudera-impala-real-time-queries-in-apache-hadoop-for-real/
[53]: https://www.mapr.com/sites/default/files/apache_drill_interactive_ad-hoc_query_at_scale-hausenblas_nadeau1.pdf
[54]: https://cloud.google.com/files/BigQueryTechnicalWP.pdf
[55]: http://static.googleusercontent.com/media/research.google.com/nl//pubs/archive/36632.pdf
[56]: http://drill.apache.org/architecture/
[57]: http://www.slideshare.net/Hadoop_Summit/w-235phall1pandey
[58]: Verhoeven, N. (2011) *Wat is onderzoek?*, Boom Lemma Den Haag.
[59]: http://www.experfy.com/blog/cloudera-vs-hortonworks-comparing-hadoop-distributions/