Hadoop biedt uitkomst bij Business Intelligence

Data lakes; de oplossing voor Big Data

Jasper Knulst

De opkomst van internetbedrijven, het mobiele internet en social media zorgt voor een ware data-explosie. De term 'Big Data' heeft zijn intrede gedaan om aan te geven dat de omvang van de data zo groot is dat deze niet zonder meer in 'gewone' DWH-omgevingen verwerkt kunnen worden.

Enerzijds omdat de gangbare, overwegend gecentraliseerde DWH-infrastructuur zucht onder dergelijke volumes. Anderzijds zijn 'high end' (MPP) databases die wel acceptabele prestaties leveren onder dergelijke belasting, voor veel organisaties kosten-technisch niet haalbaar. Hiermee dreigt veel informatie onge-bruikt te blijven liggen, wordt off-line geplaatst of zelfs gewist. Big Data, Hadoop en BI lijken elkaar echter gevonden te hebben. Dit levert een nieuwe impuls om het rendement op informatie te vergroten en vormt een basis voor een onderneming om zich te kunnen onderscheiden in de markt.
Met Big Data gaat het niet over duizenden transacties per dag, maar miljoenen. Aantallen die leiden tot Gigabytes, of zelfs Terabytes aan informatie op dagbasis. Te denken valt bijvoor-beeld aan transactielogs, sensor- en scanner reads, machinelogs en niet te vergeten weblogs. Deze data zijn overwegend semi-gestructureerd van aard. In deze bronnen ligt een schat aan informatie verborgen, maar hoe maken we deze gigantische hoeveelheid data toegankelijk voor analyse?

Verdeling werklast

Hadoop, een project van Apache Foundation biedt uitkomst. Hadoop is de open source variant van de technologie waarmee Google groot is geworden. Het is technologie die z'n wortels heeft in de (web)search en indexatie van zeer grote hoeveelhe-den data, te weten het internet. Hadoop is geschreven in Java en opgebouwd rondom twee pijlers; het Hadoop Distributed File System (HDFS) en MapReduce (MR). 'Vele handen maken licht werk' is daarbij het devies. De totale werklast wordt verdeeld over vele gedistribueerde CPU's. Yahoo heeft een Hadoop cluster met meer dan 10.000 cores om het web te indexeren, maar ook meer bescheiden clusters kunnen veel data opslaan en verwer-ken. Gedistribueerde verwerking is zeker niet nieuw. De uitda-ging hierbij is altijd hoe om te gaan met falende deelprocessen. Hoe zorg ik ervoor dat mijn job ondanks alles toch de eindstreep haalt en de juiste resultaten laat zien? Hadoop is ontworpen om 'partial failures' op te vangen en levert, mits goed ingericht, de betrouwbaarheid die men gewend is van een RDBMS.

Hoewel dus niet ontworpen voor BI, kan Hadoop zeer goed (semi)gestructureerde data aan en wordt het in toenemende mate door organisaties ingezet naast traditionele DWH-architectuur. Het HDFS is de ideale oplossing voor de opslag van Big Data vanwege de inherente horizontale schaalbaarheid. Bovendien stelt HDFS geen eisen aan de data die opgeslagen worden en worden verdeeld in logische directory's en files.
Zo ontstaat het concept van een 'data lake', in goed Nederlands een stuwmeer aan data. Een data lake fungeert als opslagplaats voor al die Terabytes aan data. Data die 'on-line' zijn en wachten om te worden gebruikt. In dat geval brengen processen in de stuwdam een (informatie)uitstroom op gang die wordt gereguleerd door een ETL-proces, waarvan MR de motor is. In tegenstelling tot een stuwmeer blijven de reeds gebruikte data in een data lake steeds ter beschikking in het 'lake' voor eventuele toekomstige vragen. De uitstroom vanuit het data lake kan worden geladen in een traditionele DWH-omgeving of direct worden gebruikt.

HDFS

Het proces om de Big Data toegankelijk te maken voor BI-toepassingen begint met het verplaatsen van de brondata naar het data lake, oftewel HDFS. In HDFS worden files ongewijzigd opgeslagen. Er gelden geen beperkingen zoals in een RDBMS. Er zijn geen schema's, tabellen, unique keys en andere (relatio-nele) constraints. Als opslagmedium kan Hadoop dus alle soorten data aan. Onder de motorkap worden de files opgeknipt in blok-ken waarvan standaard drie replica's op verschillende servers ('data nodes') in het cluster worden opgeslagen. Op de centrale

server in het cluster draait de 'name node' die bijhoudt welke blokken behoren tot welke files en waar de kopieën van deze blokken te vinden zijn. Deze opslagmethodiek staat enerzijds borg voor de veiligheid en toegankelijkheid, maar het faciliteert nog iets anders; verwerking van de data door middel van (gedistribueerde) MR.

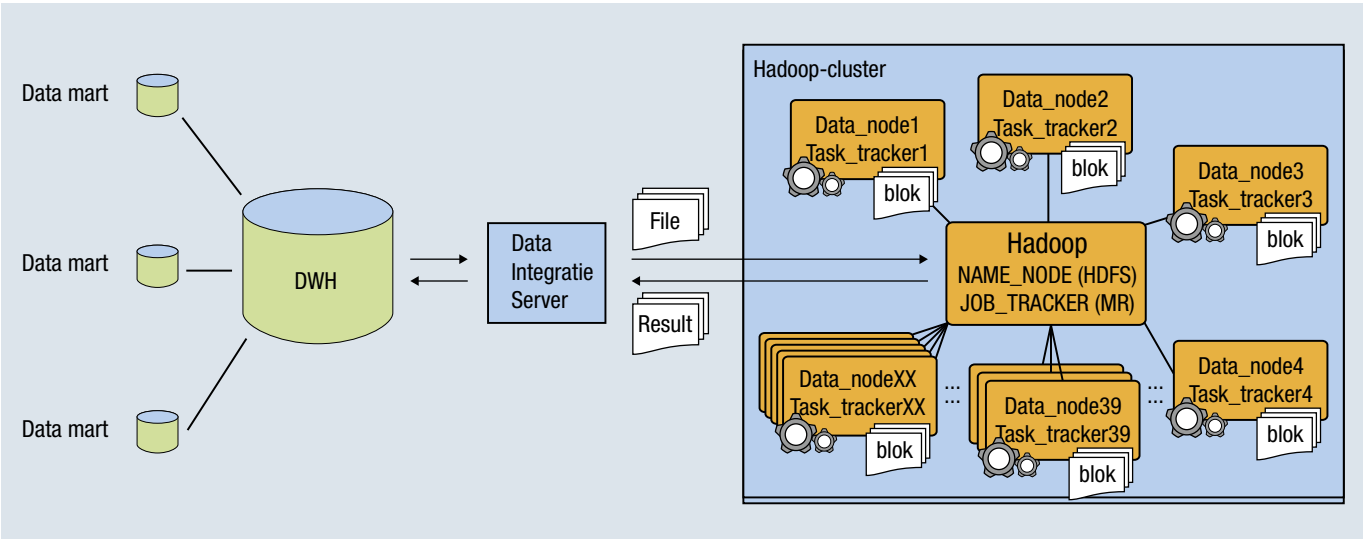
MapReduce

MR is het algoritme waarmee Google eind vorige eeuw de wereld veroverde en is even simpel als geniaal. Er is een 'map' fase waarin alle data in kaart worden gebracht (identificatie, filtering, manipulatie of verrijking) en er *key value pairs* worden gegenereerd. In de 'reduce' fase worden pairs met eenzelfde key bij elkaar gebracht en gereduceerd tot het gewenste resultaat (bijvoorbeeld aggregatie). Het resultaat is een nieuwe file die kan worden geladen in de staging, het EDW en de datamarts. Hier zijn de data tenslotte beschikbaar voor het maken van de gewenste analyses op bestaande BI front-end applicaties.
Bij de uitvoering van een MR job dragen alle CPU cores in het cluster ('task trackers'), onder centrale regie van de 'job tracker', hun steentje bij aan de totale job. Bij de toekenning van CPU's aan blokken data wordt zoveel mogelijk rekening gehouden met 'data-locality', wat wil zeggen dat verwerking plaats vindt daar waar de data zijn. Een radicale breuk met de trend van steeds zwaardere centrale high-end servers die data juist brengt daar waar de CPU's zijn. Hadoop is schaalbaar tot tienduizenden servers en draait op commodity hardware. Met elke extra server neemt zowel de opslag- als de verwerkingscapaciteit toe.
Kleinere clusters vanaf vier servers zijn al in staat om tientallen Terabytes 'on-line' te houden en veel goedkoper dan high-end appliances op basis van een RDBMS.
MR biedt oneindig veel mogelijkheden om data om te zetten naar informatie. Men kan zoeken naar woordvoorkomens in tekst, kolommen in CSV files, tags en values in XML files of zelfs pixels in grafische files. Er gelden twee beperkingen die nieuw zijn voor BI'ers; alle bewerkingen moeten vertaald worden in MR jobs én men moet denken in termen van keys en values.

Een voorbeeld; stel, er is een bronfile met op iedere regel de gegevens van een bepaald feit. De vier attributen van het feit staan achter elkaar in een vaste volgorde, gescheiden door een vast scheidingsteken. Voor MR is de key aan het begin van de map-fase de (byte)offset vanaf het begin van de file naar het begin van een enkele regel. De value is op dat moment de inhoud van de gehele regel. In de map-bewerking moet een ontwikkelaar nu de key en de values kiezen en toekennen die zullen worden doorgeven aan de reducers. Als men geïnteresseerd is in het totaal aantal feiten per attribuut 3 (prod_code), is de uitstromende key de waarde van attribuut 3 en de vaste value de waarde 1. Aangezien alle keys met eenzelfde waarde naar dezelfde reducer worden geleid kan nu een aggregaat per attribuut 3 worden berekend als resultaat. Op deze manier kan iedere denkbare ETL-transformatie worden uitgevoerd, waarbij de gehele Java-gereedschapskist aangesproken kan worden.

Nadelen

Het is ook belangrijk om de juiste verwachtingen te scheppen wanneer Hadoop wordt ingezet voor BI; het is geen vervanging van, maar een aanvulling op traditionele DWH-architectuur. Hadoop is in principe een write-only systeem. Random read-write, updates en inserts zijn en blijven het domein van het RDBMS (alhoewel HBase, de Apache NoSQL variant op basis van Hadoop dat wel kan).
Hadoop is ontworpen om batchgewijs grote hoeveelheden data van begin tot eind te verwerken. Door de grote volumes kan men niet dezelfde responstijden verwachten zoals moderne BI front-end applicaties die bieden. Denk eerder aan minuten dan aan seconden. Hadoop moet worden ingezet waar traditionele oplossingen het vanwege de grote volumes laten afweten of te duur worden. Overigens zal de verwerkingstijd vrijwel rechtevenredig afnemen met de inzet van extra nodes in het cluster.



Afbeelding 1: Hadoop-cluster.

Er zijn ook nog andere nadelen te noemen. Hadoop is geen laagdrempelige techniek voor een breed publiek. De standaard interface is een CLI en dus 'Spartaans' te noemen. Om MR jobs te ontwikkelen is Java-kennis noodzakelijk en moet code worden geschreven. Het vertalen van ETL-transformaties naar MR jobs en key value pairs vergt een geheel andere denkwijze van ontwikkelaars. Door de schemaloze opslag in HDFS moeten bij de verwerking van data vaak extra maatregelen worden genomen om fouten en datakwaliteitsissues te ondervangen. Als het opslagmedium geen integriteit afdwingt dan worden de problemen verlegd naar de verwerkende applicatie. De oplettende lezer zal het niet zijn ontgaan dat voor de headerregel van blok 1 uit het voorbeeld een voorziening zal moeten worden getroffen om deze te filteren. Het debuggen van een falende gedistribueerde applicatie die miljoenen regels verwerkt is als het zoeken naar een speld in een hooiberg; een lastig karwei en tijdrovend.

Hadoop, the easy way

Voor diegenen zonder Java-kennis, maar met SQL skills is er allereerst Hive, een Hadoop subproject dat oorspronkelijk ontwikkeld werd bij Facebook. Hive presenteert soortgelijke HDFS files als logische tabellen en ondersteunt interactie met deze data via een SQL dialect. Hive genereert op basis daarvan automatisch MR jobs. Met Hive kan vrijwel iedereen rechtstreeks ad hoc vragen stellen aan het data lake. Deze interne Hadoop DWH tool is inmiddels zo krachtig en geoptimaliseerd dat ook ontwikkelaars niet meer om Hive heen kunnen om veel sneller MR-functionaliteit (met name joins) te realiseren. Naast Hive biedt Apache ook nog een tool met de naam 'Pig' die soortgelijke voordelen biedt. Door deze applicaties is Hadoop bereikbaar geworden voor een veel breder publiek.
Daarnaast zijn er steeds meer externe pakketten met een Hadoop API die voorzien in een gebruiksvriendelijke Hadoop interface. De meeste leveranciers komen echter niet verder dan de mogelijkheid om files uit te wisselen met HDFS. Gelukkig zijn er ook geavanceerdere producten van de open source BI-leveranciers Pentaho en Talend. Pentaho heeft een integratieproduct voor Hadoop dat het zelf schrijven van MR jobs in Java overbodig maakt. Door middel van visueel programmeren met de data-integratietool Kettle/Spoon wordt code gegenereerd die rechtstreeks uitvoerbaar is als MR job. Hierbij staan vrijwel alle componenten uit de rijk gevulde gereedschapskist van Kettle ter beschikking van de ontwikkelaar. Bovendien kan Hadoop integraal vanuit de Pentaho data integration server ingepland en aangestuurd worden, waarmee Hadoop kan worden geïntegreerd in bestaande DWH architectuur. Op deze manier kan de architectuur zoals in afbeelding 1 wordt getoond daadwerkelijk gerealiseerd worden.

Hadoop in de Cloud

Hadoop kan in de Cloud gedraaid worden, maar dit levert niet de beste performance en de kosten kunnen ongemerkt hoog oplopen. Amazon levert bijvoorbeeld S3 opslag (inzetbaar als HDFS) per GB en MR capaciteit per uur. Uiteraard kan ook een dedicated Hadoop cluster worden gebouwd op basis van meerdere gevirtualiseerde Cloud servers. Aangezien we kenmerkend te maken hebben met grote datavolumes zullen de datatransferkosten van/naar de Cloud relatief hoog zijn. Voor alle Cloud-oplossingen geldt dat men het netwerk met vele andere toepassingen en/of klanten deelt. Hadoop genereert veel intracluster netwerkverkeer en presteert het best als het netwerk niet gedeeld wordt met andere toepassingen. Voor een POC of andere incidentele taken is de Cloud te overwegen, maar wanneer Hadoop een structureel onderdeel wordt van de informatiearchitectuur is het beter om in-house een cluster te exploiteren.

Hadoop en EIM

Hadoop en MR zijn dankzij hun afkomst uit de wereld van de websearch en indexatie nog in staat iets toe te voegen aan traditionele DWH's. MR biedt namelijk ongelimiteerde toegang tot ongestructureerde bronnen als bedrijfsdocumenten, e-mail en social media. Ook wat betreft deze tekstuele data is er sprake van een enorme toename. Veel bedrijven willen een combinatie van gestructureerde en ongestructureerde data gebruiken voor het analyseren van bedrijfssituaties. Hoe waardevol zou het niet zijn om uit alle e-mails aan customer service de meest voorkomende klachten per product te kunnen destilleren en te koppelen aan gestructureerde data (verkoopcijfers)? Ook zou aan analisten tegelijkertijd informatie uit databases en tekstuele bronnen aangeboden kunnen worden wanneer beide daarvoor zijn doorzocht op overeenkomstige waarden door Hadoop MR jobs. We begeven ons nu op het terrein van Enterprise Information Management (EIM). Zover zijn veel organisaties nog niet, maar het is zeker dat Hadoop in dit vakgebied toepasbaar is en waardevol kan zijn.

Conclusie

Door de beschreven technologie heeft Big Data een plaats gekregen binnen het totale informatielandschap van een organisatie. Met Hadoop kunnen we het technisch aan om Big Data ter beschikking te stellen aan een brede groep binnen organisaties. Dankzij aanvullende nieuwe BI-producten is het tevens mogelijk om Hadoop naadloos te integreren met het bestaande DWH.
De organisatie heeft nu een mogelijkheid om nieuwe informatie te genereren waarmee het zich kan onderscheiden in de markt. Het inzetten van deze nieuwe mogelijkheden brengt een organisatie (tijdelijk) op het hoogste maturity-niveau voor die processen die hiervan gebruik maken. Dit niveau wordt aangeduid met 'outsmart' en zorgt voor het onderscheidende vermogen. Met Hadoop lijken zich nog veelbelovender mogelijkheden aan te dienen voor wat betreft de koppeling van gestructureerde en ongestructureerde data, voor de 'outsmart' kandidaten van morgen.

Jasper Knulst is BI-consultant bij VLC.