Rapport van Bloor Research belangwekkend

Structureel andere oplossingen voor bouw datawarehouse

Paul van der Linden

Apama, Aleri, Alterian, Aruna. Het klinkt als een bezweringsformule tegen zweetvoeten die zo uit een willekeurige Harry Potter-film zou kunnen komen. Niets is echter minder waar. Samen met Sand en Sybase zijn dit de zes producten die door Bloor Research onder de loep zijn genomen en de inhoud vormen van het rapport 'Innovations in Datawarehousing. Answering complex, real-time and dynamic queries'.

Om maar gelijk met de deur in huis te vallen: voor alle geïnteresseerden in de ontwikkeling van de datawarehousing-markt in de komende vijf jaar is dit een belangwekkend rapport. Misschien niet de besproken leveranciers en producten, maar in ieder geval wel de besproken technieken zullen hun plaats verwerven in de datawarehouse producten van de nabije toekomst. De zes in dit rapport besproken producten vormen de voorhoede van deze nieuwe generatie.

Treurige geschiedenis

De geschiedenis van datawarehousing is goed beschouwd een treurige. Het is de verdienste van Bill Inmon geweest om duidelijk uit te leggen waarin datawarehousing nou verschilt van on-line transaction processing (OLTP). Menigeen kan de verschillen nu nog uit zijn hoofd opdreunen. Maar vervolgens zijn datawarehouses toch weer geïmplementeerd met dezelfde relationele databases die ontworpen waren voor OLTP-doeleinden. De record-georiënteerde database (raceauto) wordt ingezet om enorme hoeveelheden data te verstouwen. Een taak waar we liever een tractor voor hadden. Om dat alles enigszins mogelijk te maken werd aan die raceauto allerlei extra functionaliteit geplakt, maar het bleef in essentie een racewagen. Te veel functionaliteit waar je niets aan hebt (bijvoorbeeld locking) en te weinig waar je wél iets aan hebt (bijvoorbeeld delta-processing). En of het door Kimball gepopulariseerde stermodelleren een zegen of een vloek is, blijft discutabel. Zeker, het levert een bijdrage aan een betere performance. Maar daarmee draagt het ook bij tot het in stand houden van onze uitgebouwde raceauto. U raadt al wie dit absoluut niet erg vinden: de databaseleveranciers.

> Het bestaande datawarehouse is gebouwd met tweedehands stenen

Stel dat je software moest ontwikkelen om datawarehousing te ondersteunen en er was nog niet zoiets voorhanden als een relationele database. Wat is dan de kans dat je daarop uit zou komen? Behoudens een hersenafwijking is die kans niet zo groot. Immers, waar OLTP betekent dat je hele records invoert en bewerkt, betekent datawarehousing dat je geïnteresseerd bent in specifieke kolommen. Ga maar na: je wilt bijvoorbeeld weten wat de kenmerken (leeftijd, inkomen, postcodegebied) zijn van klanten die een bepaald product hebben gekocht. De selectie is kolomgericht. Zo ook de resultaten. Wat dus niet gewenst is, is dat hele records worden opgehaald uit de database. Hetgeen precies is wat er gebeurt. Dat dat niets doet voor de performance zal duidelijk zijn. Overigens geldt dit niet in alle gevallen: naarmate meer kolommen van een record geselecteerd worden zal de recordgeoriënteerde aanpak beter presteren dan de kolomgeoriënteerde aanpak. Reden waarom de leveranciers van de CBRD's (colum based relational databases) hun producten als complementair positioneren en niet als vervanger voor de reguliere RDBMS-en (met Sybase als uitzondering).

Onbeantwoorde vragen

Reguliere datawarehouses hebben moeite met onvoorspelbare queries. Kern van het probleem is, dat de vraag van te voren onbekend is. Hiermee is onduidelijk of gebruik moet worden gemaakt van geaggregeerde data dan wel van transactionele data. In het geval van geaggregeerde data kan gegrepen worden naar een OLAP-oplossing. Moet terug worden gegrepen op transactiedata, dan zal het beantwoorden van de vraag relatief veel tijd vragen. Het probleem bij OLAP is weer dat het gaat om voorgedefinieerde dimensies en hiërarchieën. Zit hier niet in wat

de gebruiker wil, dan is het weer terug naar af en moet een nieuwe kubus worden gegenereerd.
Een tweede categorie moeilijke vragen bestaat uit de zogenaamde complexe queries. Complexe queries hebben betrekking op gegevens op transactieniveau, hebben te maken met verschillende business rules, verschillende joins en vereisen in de uitvoering vaak full table scans. Het gaat dus om multiple of recursieve set operaties. Hiermee wordt bedoeld dat op basis van het resultaat van een selectie een verdere selectie wordt gemaakt. Denk hierbij aan vragen als: 'In welke mate heeft de introductie van een nieuw product de verkoop van bestaande producten gekannibaliseerd?' of 'Welke verkoopacties verkorten de salescyclus het meest?'.

Real time queries en datawarehouses

Dan de large table scans: indien er geen index aanwezig is zal een query leiden tot het geheel doorlopen van het betreffende bestand. Maar ook in het geval dat er wel sprake is van een index zal de database-optimizer soms besluiten dat het geen zin heeft om de aanwezige index te gebruiken. Een voorbeeld: 'Maak een overzicht van naam en emailadres van alle klanten die in juli zijn geboren'. Een hitrate van 1 op 12 zal de optimizer doen besluiten tot een full table scan. In een grote (klanten-)tabel leidt dit tot een aanzienlijke doorlooptijd.
Real time queries en datawarehouses zijn niet voor elkaar in de wieg gelegd. Immers, datawarehouses bevatten consistente en historische gegevens die niet gericht zijn op het beantwoorden van real time queries, maar bedoeld zijn voor tactische en strategische analyses. Dit betekent dat ook real time queries een uitdaging zijn voor de hedendaagse datawarehouses. Theoretisch zou hier een slag gemaakt kunnen worden als bij het beantwoorden van deze queries gebruik gemaakt kan worden van delta's (wijzigingen tussen de huidige en de vorige toestand). Je kan dan immers uitgaan van de vorige situatie en hebt alleen te maken met de wijziging (delta). Betekent minder in te lezen en te verwerken en derhalve een aanzienlijke tijdwinst. De meeste datawarehouses werken echter niet op deze manier en ook SQL is hier niet voor gemaakt. Aleri Modeler ondersteunt echter wel delta processing. Sand Technology en Sybase ondersteunen het concept van versies van data (versioned data).
Een dynamische omgeving stelt eveneens complexe eisen aan een datawarehouse. Hierbij kan het gaan om een dynamische organisatie (bijvoorbeeld als gevolg van een fusie of overname) waardoor de onderliggende databasestructuur regelmatig veranderd. Het kan ook gaan om het dynamisch veranderen van business rules en/of de data waarop het betrekking heeft. In een dynamische omgeving moeten dit soort wijzigingen ook snel worden opgepakt en doorgevoerd en bestaat er geen tijd om het probleem bij de IT-afdeling neer te leggen.
Ook time-based queries kunnen een datawarehouse hoofdpijn bezorgen. Een vraag als: 'Welke klanten kochten een barbecue een week nadat ze tuinmeubelen kochten?' is met OLAP niet te beantwoorden. Enerzijds ligt dat aan de regide structuur van de kubus, anderzijds aan de behoefte aan data op transactieniveau. Teruggaan naar de database is een optie (gesteld dat de data aanwezig is), maar vervolgens blijkt dat de SQL niet over de syntax beschikt om deze vraag te formuleren. In laatste instantie kan wellicht een gespecialiseerde data extentie worden ingezet, maar qua performance zal het nooit een prijs winnen.

Technologies to the rescue

De in het rapport besproken producten zijn kolom-georiënteerde relationele databases en vector-databases, of maken gebruik van tokenisation of vector- processing om de genoemde problemen die reguliere databases ondervinden op te lossen.
Over CBRD's hebben we het eerder gehad. Tokenisation is gebaseerd op het onderscheid tussen de waarde van de data en het gebruik ervan. Vector-databases worden met name toegepast bij document retrieval, maar zijn geschikt voor elke situatie waarin patronen vergeleken moeten worden. Toepassingsgebied is met name wetenschap en research, en minder de commerciële markt. Dit heeft ook te maken met de benodigde verwerkingscapaciteit. In specifieke toepassingen zoals Internet search engines wordt wel al gewerkt met een vectorgebaseerde aanpak.
Voorbeelden hiervan zijn Google en AltaVista. Aleri, Sand en Aruna maken allemaal gebruik van vector-technologie. Al deze technologieën worden in het rapport uitgebreid behandeld.

Conclusie

Het bestaande datawarehouse is gebouwd met tweedehands stenen. Bloors rapport laat zien dat er partijen zijn die structureel andere oplossingen bieden om een datawarehouse mee te bouwen. Waarmee een aantal van de bestaande problemen zoals complexe queries en time-series analyses eleganter en sneller kunnen worden opgepakt. Of het A-team (Apama, Aleri, Alterian en Aruna) dan wel Sand of Sybase krachtig genoeg zijn om overeind te blijven moet nog blijken. De technologie die onder de motorkap van hun producten zit is dat in ieder geval wel. En dat is een goede boodschap voor datawarehousing.

Paul van der Linden (paul.pfh.vanderlinden@atosorigin.com) is senior consultant Datawarehousing/BI bij AtosOrigin.