DBA kan niet meer om XML-DB heen

Performance 'Nerd' Marco Gralike

De DBA krijgt steeds vaker te maken met XML en is daar doorgaans niet erg blij mee. Maar er worden steeds meer XML-documenten gegenereerd. Zo veel zelfs, dat ook de DBA er niet langer omheen kan. De vraag is dan: hoe ga je om met XML in de database. De hiërarchisch opgebouwde taal conflicteert op verschillende fronten met de relationele database. Oracle heeft er een oplossing voor: XML-DB. Dit wordt gratis geleverd bij de database en het kan de – voor een database onhandig grote – bestanden snel verwerken. Als je maar weet hoe. Wij vroegen het aan een deskundige bij uitstek.

Marco Gralike, Principal Database Consultant bij AMIS, is de enige wereldwijd, die gespecialiseerd is op het gebied van Oracle XML-DB. Hij kreeg daarvoor ook een ACE/ACE Director award. Marci is ook lid van het OakTable netwerk, waar onder andere Tom Kyte en Jonathan Lewis deel van uitmaken.
Met zijn kennis van XML-DB hielp hij klanten als UWV, CJIB, ING, Syntel, ProQuest, British Sky Broadcasting (UK) en CCC (Canada). De meest opvallende is wellicht ProQuest: een Amerikaanse uitgeverij met op dit moment de grootste Oracle XML-DB omgeving wereldwijd met ruim 800 Miljoen XML documenten in 1 XMLTYPE kolom. De verwachting is dat dit over de 1 miljard xml documenten zal zijn binnen een 1/2 jaar.

Het begon in 2006 toen bij een grote overheidsinstelling alle historische gegevens moesten worden omgezet naar één grote bak met data. Daar werd besloten om dat met Oracle XML-DB te doen. Dit leek de beste manier om alle hiërarchische, relationele, objectgeoriënteerde en mainframe omgevingen met hun verschillende indelingen te converteren naar XML Schema, te exporteren als XML data en van daaruit een database te creëren, die ook met het landelijke SUWI-net (waar onder andere alle Nederlandse gemeenten op zijn aangesloten) kon communiceren. Alle gegevens van de virtuele Nederlanders van 1977 tot en met 2003 werden zo als XML-DB opgeslagen. Met de functionaliteit van een database, maar in XML-formaat.

De methodiek werd door de architecten van de instelling van het internet gehaald. Toen bleek al snel dat te weinig bekend was van deze technologie, niet alleen bij de betreffende instelling, maar in heel Nederland. Gralike, die door de instelling was ingehuurd om een en ander te begeleiden, is zich daarop op dit onderwerp gaan specialiseren.
"Dat was een hele steile leercurve, want er was inderdaad nog niet veel kennis over XML-DB beschikbaar. Ik heb veel van het Oracle XML-DB forum gehaald en ben ook in contact gekomen met het Oracle developmentteam in Amerika. Het bijzondere van het verhaal is dat in mijn vakgebied – databasebeheer en -performance – wereldwijd eigenlijk weinig navolging is. Het wereldje wordt nog altijd geregeerd door Oracle-developers, die in een SOA-omgeving werken en daarbij relationele data moeten transformeren naar XML of dat ze XML krijgen aangeleverd en dat als relationele data in de database moeten zetten", constateert Marco Gralike.

Pijn aan het hart

"De klassiek opgeleide DBA houdt zich het liefst nog verre van dit onderwerp. De developers pakken het wel op, maar de databasebeheerders kiezen er liever voor om de gegevens als objecten op te slaan, onder het motto 'geen XML in mijn database'. Op zich is dat heel jammer, want er wordt steeds meer XML gegenereerd. Vanwege SOA, webpagina's, blogs, RSS-feeds en noem maar op. Zelfs de Microsoft Office producten van Word, Excel en andere zijn in XML-formaat. Daar kun je heel veel mee doen in de database. De Oracle XML-DB-functie is erin gespecialiseerd om effectiever met die data om te gaan."
De sympathieke Oracle performance 'Nerd', die voorheen naast Oracle databases werkte met middleware, application servers en operating systems, maar nu voornamelijk is gespecialiseerd in XMLDB, implementeerde en configureerde systemen bij bedrijven als ING, D-Reizen, De Zaak, AH, BASF, CJIB, UWV, Stepstone, Sealand/Maersk en verschillende andere.

Marco Gralike: "Met XML heb je veel meer mogelijkheden in de database om metatdata te ontrafelen en efficiënt te behandelen."

En met veel plezier. "Het doet me pijn aan het hart als ik een cursus geef en de cursisten zeggen achteraf dat zij toch liever de bestanden als tekst in de fileserver stoppen. Stel je voor: je hebt XML. Daarmee heb je in de database veel meer mogelijkheden om de daarin opgeslagen metadata te ontrafelen en efficiënt te behandelen. Je kunt de documenten bijvoorbeeld heel goed op inhoud nazoeken. XML heeft al een structuur, dus je kunt daarvan gebruik maken."

Oracle XML-DB heeft een data dictionary, een hoeveelheid tabellen waarin de metadata over de XML zijn opgeslagen. Als gevolg daarvan kun je effectief en veel efficiënter dan met klassieke XML parsers de data ontrafelen of genereren. Oracle heeft daar drie methodieken voor:

- Het oude Character Large Object (CLOB). Gooi alles maar in een grote ton met data.
- Object relationele opslagmethodiek. Die is initieel opgezet met de OO-functionaliteit van de Oracle database.
- Binary XML. Dat is waar de meeste XML database-leveranciers zich mee proberen te profileren als zijnde 'native'.

Je hebt dus in principe drie verschillende methodieken om veel meer use cases met XML-opzet efficiënt te kunnen uitvragen.

SBR-standaard

Eind mei heeft het ministerie van Economische Zaken bekend gemaakt dat in ons land de SBR-standaard (Standard Business Reporting) van kracht wordt. SBR is afgeleid van de officiële taal XBRL: een op XML gebaseerde open standaard voor het samenstellen en elektronisch uitwisselen van business rapportages en gegevens via het internet. Dankzij het gebruik van XBRL zul je met een paar klikken je jaaropgave kunnen genereren. Dat betekent een enorme kostenbesparing voor het bedrijfsleven. De grote softwareleveranciers, zoals SAP en Microsoft, hebben hun software inmiddels XBRL-enabled

gemaakt. Daarnaast bieden verschillende organisaties al specifieke toepassingen aan.
Marco Gralike signaleert dat grote financiële instellingen, uitgeverijen en andere bedrijven met service georiënteerde architecturen heel veel XML genereren en daar ook iets mee willen gaan doen. Het is echter heel moeilijk om op de ouderwetse manier die data in een OLAP-omgeving uit te vragen. Hij heeft inmiddels een aantal implementaties gerealiseerd, waarbij grote volumes aan data kunnen worden uitgevraagd op basis van XML-inhoud of op combinaties van XML en Oracle tekst. De grootste implementatie op dit gebied is bij uitgeverij ProQuest in het Amerikaanse Michigan. Deze bevat tien terabyte aan data, waaronder vijftien miljard XML-documenten. "XML-DB zorgt voor nieuwe interfacing in de database. Je kunt de database gewoon http-enabled maken, dus je kunt met http rechtstreeks naar de database queryen. Dat doe je met Xpath of Xquery in XML-formaat. Je hebt FTP- en WebDav-toegang. Je kunt als het ware de database mounten als schijf. En een van de mooie dingen die je kunt doen is Worddocumenten rechtstreeks via een explorer-window in de database slepen. Omdat het XML is kun je het oppakken, anonimiseren, je eigen logo's erin zetten en daarna terugzetten naar een andere plaats in de database en toegankelijk maken als geanonimiseerde CV. Dat zijn geweldig leuke dingen. Met een beetje fantasie kom je heel ver", stelt Marco met een brede grijns vast.

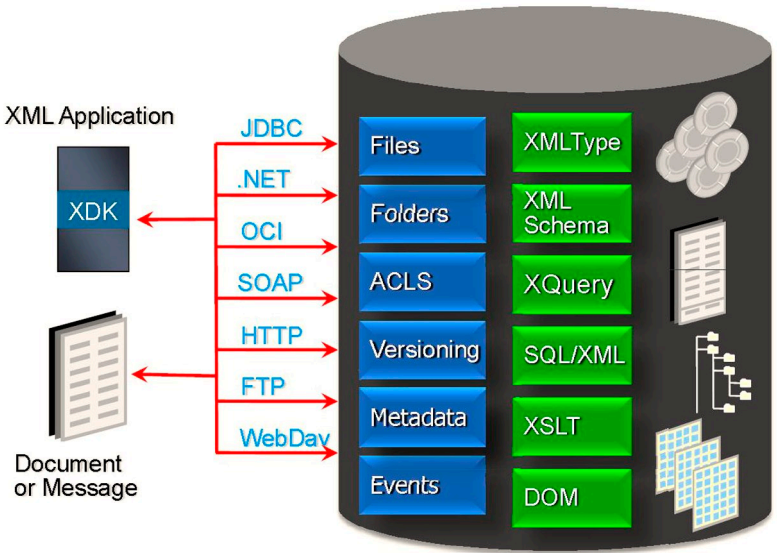
'Mijn grootste passie is om goede, werkende databases neer te zetten.'

"Mijn grote passie is om goede, werkende databases neer te zetten. Er zijn legio voorbeelden te bedenken, waar je XML-DB goed kunt inzetten. Het is helaas nog steeds een redelijk onbekend product. Een van de grote redenen daarvan is dat Oracle XML-DB een no-cost option is. Dat wil zeggen dat het gratis bij de database wordt geleverd. Dat is voor de sales-afdeling van Oracle dus behoorlijk onaantrekkelijk. En ook voor marketing is het moeilijk om hier een voet aan de grond te krijgen. Wat vaak gebeurt is dat een organisatie op een verkeerde manier met XML omgaat, waardoor de database slechter gaat performen. Dan wordt

het probleem bij developers neergelegd, die op een of andere manier bij mij uitkomen en pas dan wordt XML-DB erbij gehaald".

Gratis in huis

"Het leuke is dan dat je de mensen ter plekke kunt bijspijkeren. Dat je hun omgeving honderd keer zo snel kunt maken. Maar duidelijk is dat deze wereld nu nog wordt gedomineerd door developers, die XML-DB oppakken en gebruiken. Voor de DBA is de leercurve heel steil, omdat ze echt de XML-wereld moeten binnentreden. Het Oracle XML-DB-team werkt heel hard aan nieuwe uitbreidingen en verbeteringen van het product en levert regelmatig nieuwe patches. Het nadeel hiervan is dat als je dit niet snel genoeg volgt, je veel performancewinst kunt mislopen. Ik kom ook steeds vaker instellingen tegen met een eigen XML-schema voor bijvoorbeeld XBRL, maar waar nog te weinig met XML-DB wordt gewerkt, gewoon omdat ze niet weten dat ze het al – gratis – in huis hebben. Het is beschikbaar voor alle Oracle databaseproducten, van de standard- tot de enterprise-edition en zelfs in de – ook gratis – Oracle XE-omgeving. Combineer je dat met een andere no-cost option zoals APEX, dan kun je echt geweldige dingen doen".



"Vanaf Oracle 11 is er uit de webdav-hoek een native database webservice waarbij je op basis van PL/SQL in vijf simpele stappen een full-blown webservice uit de grond kunt stampen. Vanaf dat moment is je Oracle-database een SOA-endpoint. In de Oracle SOA-community is de database nog steeds een domme kaartenbak, waarin geen businesslogica wordt opgeslagen. Laat staan dat ze de database gebruiken voor slimmere dingen. Je ziet dat het in die omgeving heel moeilijk is om mensen te overtuigen een deel van de XML-datalast af te dragen aan de klassieke database-kant. Developers zien het als een voordeel, maar de DBA wil er vaak niet aan."

"Als ik een XML-document heb van 100 MB, heb ik een responsetijd van minder dan een seconde. Hoe lang doe je daarover in je middle-tier, vraag ik dan aan de DBA-collega's. Een XML-document is groot als het één MB is. Je ziet wel dat steeds grotere XML-documenten worden gegenereerd. Zelfs voor een Oracle-database is het een hele toer – als je de kneepjes van dit specialisme niet goed kent – om dit goed te laten performen. Maar het is tegenwoordig technisch heel goed mogelijk om het op een efficiënte manier te doen. Je moet een XML-document eigenlijk beschouwen als een database op zichzelf. Je zet een database in een database. En die database is behoorlijk irritant om uit te vragen als je het niet goed doet. Als je het niet goed doet en je verwerkt dat in het geheugen, dan zul je zien dat het geheugen niet voldoet. Die XML-passes in het geheugen trekken met hun 100 MB het geheugen helemaal leeg. Als je dan ook nog carthetische productcombinaties krijgt, blaas je het spul nog verder op en voor je het in de gaten hebt bezwijkt de techniek", aldus Marco Gralike.

Een oplossing zou kunnen liggen in verbeterde hardware. Met meer geheugen en andere configuratie van het systeem kan XML ongetwijfeld sneller worden verwerkt. Marco Gralike zegt daarover: "De Exadata salesafdeling zou mooie resultaten kunnen bereiken, omdat XML-DB zo'n afschuwelijk performancemonster is. Zij kunnen een product neerzetten, dat goed XML kan verwerken. Ik verwacht ook wel dat het die kant op zal gaan. Maar voorlopig kan ik het zonder die bijzondere machines ook nog wel redden."

Tussenoplossingen

"Ik heb altijd naar XML-DB gekeken vanuit mijn traditionele relationele achtergrond en het zijn in principe dezelfde trucjes die je moet toepassen. De XML-wereld, die pas twaalf jaar jong is, is nog steeds zoekende qua design. In de relationele database doe je dingen qua design op een manier, waarbij je dezelfde technieken kunt toepassen. Als je de database best practices volgt zie je dat dezelfde regels ook van toepassing zijn op het XML-schema. Als je die toepast krijg je ook een goede performance. Omdat er vooral developers mee aan de slag gaan wordt te veel een beroep gedaan op het geheugen, waardoor het vaak fout gaat. Dan zegt de klassieke omgeving: zie je wel, het werkt niet."

"Een beetje tegen het gevoel in doe ik het ontrafelen niet in het geheugen, maar als opslagmethodiek. Je zet het in een tabel, die alles automatisch mapt, converteert, in de juiste

proporties en kadertjes plakt, waardoor als je de data later selecteert, je een hele snelle respons krijgt. Het wiel om XML slim in het geheugen te kunnen behandelen is wereldwijd nog niet uitgevonden. Met de opslagtechniek kun je dit omzeilen. Het is het niet ideaal, want de harde schijf is natuurlijk niet zo snel, maar je kunt er toch een gigantische performancewinst mee bereiken."

Het grote voordeel van XML-DB – de metadata-structuren, die de XML netjes catalogiseren en uit elkaar rafelen – is nog niet uitgevonden voor het geheugen. Daar zijn nog geen indexen, mapping-methodieken en datadictionary-oplossingen, waardoor alles als één lange string van soms wel 100 MB wordt verwerkt. Daarna wordt er dan weer XML van gemaakt. Dat is enorm inefficiënt.
Gralike: "Een relationele database van 100 MB is niet veel. Duizend rijtjes in een kolom is niet veel. Waarom werkt het dan niet met duizend XML-rijtjes in een tabel? Veel methodieken die op universitair niveau zijn ontwikkeld zijn gericht op het zo snel en efficiënt mogelijk het XML-gedeelte af te wikkelen. Er is nog geen oplossing gevonden om vooraf in het geheugen te mappen en indexeren. Je kunt niet bijhouden waar het over gaat: een number, een integer, een string? Kan ik daar slimme dingen mee doen? Wat ze wel proberen is om zo snel mogelijk door de XML-boomstructuur heen te gaan, maar er is nog geen methodiek gevonden om te bepalen hoe iets eruit ziet en wat je daar dan wel mee kunt doen."
"Het probleem tussen Java en de relationele wereld doet zich ook voor bij XML. Dit is immers hiërarchisch en relationeel is relationeel. In programmeertechnieken en denkprocessen blijkt het heel moeilijk om dit van de ene naar de andere techniek te zetten. Daar zijn wel tussenoplossingen voor, zoals Java en Hybernate, maar dat kost enorm veel performance.
Er zijn ook generieke oplossingen, die te generiek zijn voor de business. Wat ik meestal adviseer als er geen XML-converteer-oplossing is, laat XML dan gewoon XML en doe het niet relationeel. Je krijgt anders heel veel designproblemen. Omgekeerd is het heel moeilijk om de relationele moeder-kind relaties vast te leggen in een XML-structuur", aldus Marco Gralike.

> *'Ik kan op grote conferenties geen leuke dingen vertellen. Het moet simpel, omdat het publiek het anders niet snapt.'*

Toepassing

Hoe enthousiast hij ook is over de mogelijkheden van Oracle XML-DB, soms kun je het toch beter links laten liggen, vindt Marco Gralike. "XML-DB moet je inzetten voor XML. Is er geen XML in het spel, gebruik het dan niet. Je brengt dan namelijk veel design- en performanceproblemen in die je niet wilt hebben. Goede use cases voor XML-DB zijn die, waar je de relationele omgeving wilt ontsluiten naar de SOA-omgeving. Als je dat gaat doen, doe dat dan tenminste met een functie die je gratis krijgt en geperfectioneerd is qua performance en onderhoud. XBRL is steeds meer in opkomst, gezien het feit dat het door de staat wordt afgedwongen. Een andere goede toepassing van XML-DB is het onderzoeken van de data, daar waar de data al een XML-structuur hebben. En wanneer blogs, rss-feeds of geografische systemen in de database moeten worden verwerkt kun je ook grappige dingen doen met XML-DB."

Vanuit de DBA-vakgroep blijft het een moeilijk, emotioneel onderwerp. Zij hebben zoiets van 'het is niets en het is nooit goed geweest, dus kom niet aan mijn database'. Gralike is in deze wereld een evangelist. Zelfs XML-software concurrenten houden hem in de gaten; kijken wat hij nu weer allemaal blogt. Zijn drijfveer is ook heel begrijpelijk, want het zal in de nabije toekomst onmogelijk blijken om XML uit de database te weren.
"Ik vind het jammer dat ik het op grote conferenties altijd simpel moet houden. Ik kan geen leukere dingen vertellen, omdat het publiek het dan niet snapt. En dan krijg ik een onvoldoende…"
Samen met lotgenoot en APEX-expert Roel Hartman heb ik een gratis APEX versiebeheer applicatie op het internet gezet (XACE) in de hoop dat dit wat enthousiasme gaat kweken. En als lid van de customer advisory board van Oracle voor XML-DB denk ik mee over de ontwikkeling van het product. Het is toch mooi dat je met drie gratis producten van Oracle – XE, APEX en XML-DB – leuke oplossingen kunt neerzetten."

Links

http://blog.gralike.com
http://www.amis.nl
http://www.xbrl-nederland.nl/
http://xace.sourceforge.net
http://www.oracle.com/technetwork/database/features/xmldb/index.html
http://www.oaktable.net/members

Robert de Ruiter is hoofdredacteur van Optimize.