ABIS NIEUWSBRIEF VOOR DB2 PROFESSIONALS

JAARGANG 1 - NUMMER 4, DECEMBER 2002

OPEN CURSOR

Dit is het vierde, en tevens laatste nummer van 'Exploring DB2' van 2002.
De positieve respons op deze publicatie was groot: zowel inhoud als kwaliteit werden door de meeste lezers zeer gewaardeerd. Dit sterkt onze motivatie om ook volgend jaar deze publicatie verder te zetten. Met onderwerpen die ons allen interesseren: applicatie- en systeemperformance, nieuwe tendensen in applicatieontwikkeling, de impact van nieuwe releases, en uiteraard de integratie van DB2 in haar omgeving.
Tijdens de komende wintermaanden beraden we ons over de inhoud van de volgende nummers - uw suggesties en vragen zijn welkom. Rond 15 februari mag u het volgende nummer verwachten.
Het ABIS team wenst u prettige eindejaarsfeesten, en een uitdagend 2003!

IN DIT NUMMER:

- XML en DB2 - een overzicht van de *DB2 UDB XML Extender*. In dit eerste deel wordt de kolommethode besproken.
- We doen steeds meer online - *Dossier 7: de online load.*
- Ideaal voor gedistribueerde operaties in een heterogene database omgeving, of specifiek voor gedistribueerde datawarehouses: de *DB2 UDB Federated Database*. We lichten een tipje van de sluier ...
- *Cursusplanning jan 2003 - mrt 2003.*

CLOSE CURSOR

Het volgende nummer staat weer voor een groot deel in het teken van applicatieperformance. Ook beveiliging en performance in een 'Federated Database' zijn er aan de orde.

Tot dan!

De DB2 XML Extender - 1

Tom Avermaete (ABIS)

Of hoe DB2 omgaat met XML documenten ...

Bewaar- en opslagmethodes

DB2 versie 7 beschikt over twee methodes om XML documenten (in de meest uitgebreide zin van het woord) te bewaren.

De eerste manier, de zogenaamde Columns methode, bewaart het volledige XML document - inclusief tags - in een speciale XML kolom. Het XML document dat zich in de kolom bevindt, kan doorzocht worden door gebruik te maken van Xpath queries. Bovendien kunnen gegevens ook gemapped worden naar zogenaamde side-tables, die dan de functie van een index krijgen. Er bestaan in de XML Extender verschillende functies voor XML kolommen. Met behulp van deze functies kan het opgeslagen XML document doorzocht en geüpdated worden. De Columns methode is vooral voordelig indien de XML documenten al bestaan, ze zelden geüpdated, maar veelvuldig gelezen worden.

Een tweede methode is de Collections methode. Bij deze methode wordt het XML document volledig gemapped naar DB2 tabellen. Het binnenkomende XML document wordt dus volledig geanalyseerd en de opgenomen gegevens worden in DB2 tabellen bewaard - de tags en de XML markup (lees: het document) dus niet. De mapping tussen het XML document en de DB2 omgeving wordt beschreven in een zogenaamd DAD (Document Access Definition) bestand. Het document dat in DB2 wordt opgeslagen duidt men aan als shredded of decomposed. Indien men het XML document terug wil opvragen wordt het op basis van de gegevens in de tabellen en de bijbehorende DAD terug van nul opgebouwd. Men spreekt van een composed XML document. Merk op dat elke referentie naar het originele document ontbreekt: strikt genomen hoeft het gegenereerde document dus niet identiek te zijn aan het originele document - zelfs niet inhoudelijk.

DTD: Document Type Repository

Een DTD (Document Type Definition) bevat informatie over de structuur van een XML document. Via deze DTD kan men nakijken of een XML document wel juist gestructuureerd is, of het met andere woorden wel "valid" is.

De DB2 XML Extender beschikt over een DTD repository, de zogenaamde DTD_REF tabel. Deze kan worden gebruikt:

- louter als een DTD repository voor de XML documenten door DB2 in behandeling;
- als een bron voor XML documentvalidatie door DB2 zelf uitgevoerd tijdens het behandelen van XML documenten.

DAD: Document Access File

Een DAD (Document Access Definition) bestand is een geformatteerd XML bestand dat instructies bevat over hoe het XML document gemapped moet worden met de gegevens in DB2. Indien de Columns methode wordt gebruikt, worden aan de hand van de DAD de sidetables opgebouwd. Bij de Collections method staat de mapping voor alle gegevens beschreven. Net zoals de DTD's worden de referenties naar de bijhorende DAD bestanden bewaard in een aparte tabel met de naam XML_USAGE.

XPATH

DB2 gegevens worden opgevraagd met SQL; XML data worden opgevraagd met behulp van de Xpath query taal.

XPath is de query taal die door DB2 werd geïmplementeerd om elementen of attributen binnen een XML document op te vragen. Zoals elke query taal, bestaat XPath uit een aantal expressies. Een XPath expressie bestaat uit steps, gescheiden door een voorwaartse slash. Een step bestaat typisch uit drie delen:

- een 'axis' die aanduidt of er naar een element (standaard) of naar een attribuut (@) wordt gezocht;
- een 'nodetest', die meestal de naam van het element dat men zoekt, aanduidt; met een * kan men alle elementen binnen een niveau specificeren;
- een predicate, met name, een optie waarmee men voorwaarden kan leggen op de nodes die men wil selecteren.

Standaard beginnen queries met het root of eerste element en zakken ze zo de hiërarchie af. Indien men // specificeert, zoekt men alle elementen met een specifieke naam, waar ook in de hiërarchie.

Voorbeeld 1

```
<? xml version="1.0" ?>
<PersonList>
   <Person id="1">
      <FirstName>Mieke</FirstName>
      <LastName>Moens</Lastname>
   <Person id="2">
      <FirstName>Jos</FirstName>
      <LastName>Bosmans</Lastname>
   </Person>
</PersonList>

/PersonList/Person: selecteert de elementen Person binnen de root
PersonList.
//LastName: selecteert alle elementen met de naam LastName.
/PersonList/Person[@id="2"]: selecteert de persoon die een attribuut id
heeft met als waarde 2.
```

Voorbeeld: zie *XML Column Methode* op p. 9.

Een XML document toevoegen: de Column methode

Eerst moet de database XML-enabled worden. Dit creëert het DB2 schema met de naam DB2XML en stelt enkele user defined types en functies ter beschikking. Ook de tabellen DTD_REF en XML_USAGE worden aangemaakt.

Vervolgens maken we de kolom aan waarin we het XML document willen bewaren. Deze kolom moet van het type DB2XML.XMLVarchar (beperkte omvang), DB2XML.XMLCLOB (max. 2 GB) of DB2XML.XMLFILE (document wordt extern bewaard) zijn.

De DAD kan nu worden aangemaakt. Deze DAD volgt een bepaalde opgelegde structuur en wordt op zijn beurt gevalideerd door het bestand DAD.DTD. In de DAD wordt bepaald welke methode (Column of de Collection) we willen gebruiken en of we bij het inserten willen valideren of niet. De mappings tussen het XML document en de DB2 tabellen alsook de datatype mapping tussen DB2 en XML worden in deze file eveneens meegegeven. Indien sommige gegevens in het XML document meermaals kunnen voorkomen, spreken we van "Multiple Occurrences" (In dit voorbeeld Person). Elementen (voorbeeld Firstname en LastName) die op hetzelfde niveau maar 1 maal voorkomen, mogen in dezelfde side-table weggeschreven worden.

```
<?xml version="1.0"?>
<!DOCTYPE DAD SYSTEM
"C:\dxx\dtd\dad.dtd">
<DAD>
<dtdid>c:\dtd\person.dtd<
/dtdid>
<validation>NO</
validation>
<Xcolumn>
<table
name="person_names">
  <column name="plname"
   type = varchar(50)"
   path="/PersonList/
Person/LastName"
   multi_occurence="YES"/>
</table>
</Xcolumn>
</DAD>
```

Eenmaal de DAD aangemaakt kan de XML column geënabled worden. Dit is niet verplicht maar wel noodzakelijk indien men side-tables wil aanmaken (en vullen). Vooreerst wordt het DAD bestand ingelezen (of geparsed). De side-tables worden aangemaakt (in dit geval person_names) en een aantal triggers worden eveneens gedefinieerd. Deze triggers zorgen ervoor dat de side-tables gesynchroniseerd blijven wanneer we die XML kolom manipuleren. In de XML_Usage tabel wordt een rij toegevoegd die bijhoudt welke DAD moet gebruikt worden voor welke kolom. Een kolom die met succes enabled is kan geïnserted worden met behulp van een aantal functies. Op deze manier wordt een XML document geparsed en in een DB2 kolom bewaard.

DOSSIER 7

Na de online reorg, nu ook de online load

Het load utility wordt gebruikt om bulk data te inserten in DB2 tabellen. Tot voor versie 7 van DB2 for OS/390 had geen enkele andere transactie of applicatie toegang tot data in de tablespace tijdens het loaden. Er waren weinig andere mogelijkheden om dit te omzeilen, tenzij insert applicaties schrijven. Ondanks de nadelen die deze insert applicaties hebben, waren ze soms toch de enige optie, bijvoorbeeld wanneer men ook wenst dat triggers uitgevoerd worden. Vanaf versie 7 kan men een online load resume uitvoeren. Qua syntax is het alsof je het load utility uitvoert, maar intern speelt er zich iets heel anders af. Het intern proces gelijkt veel meer op hoe DB2 inserts uitvoert. Hierdoor heeft men wel het nadeel dat de performance iets minder goed is dan een gewone load maar anderzijds zijn er een aantal voordelen. Wil je weten hoe het nu juist werkt? Wel, kijk naar hoe DB2 een insert uitvoert: we maken de oefening voor logging, free space, referentiële integriteit, triggers. Aan u om ze over te doen voor het maken van de indexes en de duplicate keys.

Free space: Er wordt geen rekening gehouden met de instellingen PCTFREE en FREEPAGE, integendeel de vrije ruimte wordt opgeconsumeerd. Net zoals bij een insert wordt er daarentegen wel rekening gehouden met de clustering om de plaats van het nieuwe record te bepalen. De vraag is natuurlijk in welke mate dit goed kan gebeuren bij het loaden van veel records. Een reorg dringt zich op.

Referentiële integriteit: Per record dat geladen wordt, wordt de referentiële integriteit meteen nagekeken. Dit heeft gevolgen voor tabellen met self-referencing foreign keys. De records moeten voor het laden in de juiste volgorde gesorteerd worden. De optie ENFORCE NO kan niet gebruikt worden.

Triggers: Bij een gewone load worden triggers niet uitgevoerd, bij een online load resume wel.

Logging: Alle acties moeten gelogd worden, dus de optie LOG YES is verplicht

Wat de verschillende uitvoeringsfases van het load utility betreft, zijn er ook een aantal verschillen. De insert acties worden in de eerste fase na de UTILINIT fase uitgevoerd: de RELOAD fase. Aangezien er nu per insert actie rekening gehouden wordt met de databaseomgeving (vb. de eerder behandelde referentiële integriteit) zijn de daarop volgende fases niet meer relevant. De SORT, BUILD, SORTBLD, INDEXVAL, ENFORCE en REPORT fases worden niet uitgevoerd.

Hoe voer je nu een online load uit? Wel met LOAD RESUME YES SHRLEVEL CHANGE. Denk eraan dat een aantal andere load opties zoals REPLACE niet meer mogelijk zijn.

En wat zou er nu gebeuren met indexes en duplicate keys?

Wenst u meer informatie over de performance van de online load? Dan moet u hier zijn: *http://www7b.software.ibm.com/dmdd/library/techarticle/0203shibamiya/0203shibamiya2.html*.

Katrien Platteborze (ABIS)

Federated Databases - 1

Kris Van Thillo (ABIS)

Het is voor de meeste bedrijven een realiteit: data zijn opgeslagen in een veelheid van database systemen - systemen met uiteenlopende karakteristieken, gebaseerd op zeer specifieke zoektechnieken. Hoe slaagt men er dan in de gegevens uit deze verschillende bronnen te integreren, om relevante bedrijfsinformatie te bekomen? Aan de hand van IBM 'Federated Database' technologie. Een aantal basis-componenten van deze technologie, geïntegreerd in DB2 UDB, worden in wat volgt uiteengezet.

Karakteristieken

Een DB2 UDB 'Federated Database' kan het best worden omschreven als een 'virtuele database' - eigenlijk een aaneenschakeling (i.e. federatie) van onafhankelijke, externe, al dan niet relationele databases. Merk op dat afhankelijk van de aard van de bron waarmee men wil integreren, bijkomende software vereist kan zijn: IBM DataJoiner of IBM (non-)Relational Connect. Eén specifieke DB2 UDB database in de opzet krijgt echter wel de rol van 'Federated Server'.

Een DB2 UDB 'Federated Database' heeft volgende specifieke karakteristieken:

- transparantie voor de gebruiker: lokatie, netwerk, zoekmethode (e.g. SQL), SQL dialecten etc. worden door de 'Federated Server' opgevangen;
- heterogeniteit en onafhankelijkheid: de virtuele database is een federatie van op zichzelf staande databases; elke database blijft een afzonderlijke entiteit, en moet als dusdanig worden beheerd. De lokale applicaties ondervinden van de federatie geen hinder;
- functiecompensatie en -specialisatie: de UDB optimizer gaat na of het zinvol is bepaalde functies op de externe data source, dan wel lokaal in de federated server, uit te voeren.
- optimalisatie: de DB2 UDB optimizer houdt rekening met de karakteristieken van de externe data source. Op basis van de query karakteristieken zal worden nagegaan in hoeverre een query locaal of remote moet worden uitgevoerd, waar de bulk van de data zal worden gesorteerd, etc.

Componenten

Voorbeeld: zie *Federated Database* op p. 10.

De volgende componenten maken deel uit van een 'Federated Database'.

- Een 'Federated Database server', i.e. een gewone DB2 UDB instance, waarbij de 'Federated' module werd geactiveerd. Deze module maakt standaard deel uit van de DB2 UDB Enterprise Edition database oplossing. Eens geactiveerd, kunnen 'wrappers', 'servers' en 'nicknames' worden gebruikt.

- Wrappers, het best te definiëren als een logische naam die gegeven wordt aan een specifiek type externe data source - i.e. een specifiek client dll nodig om de externe data source aan de hand van een C/S architectuur te benaderen. Eén 'wrapper' per data source is vereist. Soms kan het nodig zijn een 'wrapper' per data source versie te definiëren. Bijvoorbeeld - drie Oracle7 data sources gebruiken dezelfde 'wrapper'; één Oracle 8i data source vereist een specifieke Oracle8i 'wrapper'.

- Een 'Server' is een concrete implementatie van een 'wrapper'. Of om het bovenstaande voorbeeld te vervolgen: drie Oracle7 servers zullen moeten worden gedefinieerd, en één Oracle 8i server. Een 'server' bevat alle gegevens die de specifieke client voor die externe data source nodig heeft om de externe data source te kunnen identificeren en benaderen. Of om het Oracle voorbeeld verder te zetten: u specificeert de naam van de te benaderen Oracle instance.

- Een 'Nickname' heeft veel weg van een view en/of alias. Een 'nickname' is een lokale, logische naam opgeslagen in de cataloog van de 'federated database server' die verwijst naar een specifiek benoemd object, aangemaakt in de externe data source, en aan de hand van de server/wrapper combinatie geïdentificeerd.

Queries

Een federated database wordt typisch benaderd aan de hand van SQL. Eens de query is aangeboden, wordt deze in query fragmenten ontleed. Deze fragmenten bevatten een optimaal plan per impliciet aangeroepen externe source. Deze fragmenten zijn samengesteld door DB2 aan de hand van de informatie door de wrappers en servers doorgespeeld, waardoor voor de totale query een kostindicatie voor verschillende planalternatieven kan worden bepaald.

DB2 UDB versie 7 ondersteunt enkel read-only toegang tot externe databronnen.

Besluit

Een federated database bevat een hele reeks componenten die het gebruikers en administrators moeten mogelijk maken op een transparante wijze disparate databronnen te integreren. Alhoewel opzet en configuratie niet echt complex zijn, mag de beheerder een aantal cruciale elementen van performance en beveiliging niet uit de weg gaan. In een volgend nummer wordt hier dieper op ingegaan.

CURSUSPLANNING JAN-MRT 2003

DB2 for OS/390, een totaaloverzicht	1625 EUR	13-17/01 (W), 03-07/02 (L), 17-21/02 (W), 24-28/03 (L)
DB2 UDB, een totaaloverzicht	1625 EUR	17-21/02 (W), 24-28/03 (L)
RDBMS concepten	325 EUR	13/01 (W), 03/02 (L), 17/02 (W), 24/03 (L)
Basiskennis SQL	325 EUR	14/01 (W), 04/02 (L), 18/02 (W), 25/03 (L)
DB2 for OS/390 basiscursus	975 EUR	15-17/01 (W), 05-07/02 (L), 19-21/02 (W), 26-28/03 (L)
DB2 UDB basiscursus	975 EUR	19-21/02 (W), 26-28/03 (L)
DB2 concepten	375 EUR	07/03 (W)
SQL workshop	700 EUR	27-28/01 (W), 24-25/02 (L)
DB2 for OS/390 programmering voor gevorderden	1050 EUR	02-04/04 (W), 19-21/05 (L)
DB2 for OS/390: SQL performance	1200 EUR	03-05/03 (W), 11-13/06
Fysiek ontwerp van relationele databases	700 EUR	11-12/02 (W), 28-29/04 (L)
DB2 for OS/390 database administratie	1600 EUR	18-21/03 (W), 12-15/05 (L)
DB2 UDB database administratie	1600 EUR	17-20/03 (W), 05-08/05 (L)
DB2 UDB systeembeheer en performance	400 EUR	21/03 (W), 09/05 (L)
DB2 UDB for OS/390 V7 upgrade voor ontwikkelaars	375 EUR	28/02 (L), 06/03 (W)
DB2 UDB en zijn extenders: XML en text search	200 EUR	28/03 (W), 16/05 (L)
DB2 UDB integratie met MQSeries	200 EUR	28/03 (W), 16/05 (L)

Plaats: L = Leuven; W = Woerden
Details, andere data en bijkomende cursussen: www.abis.be

Postbus 220
Diestsevest 32
BE-3000 Leuven
Tel. 016/245610
Fax 016/245691
training@abis.be

abis
TRAINING & CONSULTING

Postbus 122
Pelmolenlaan 1-K
NL-3440 AC Woerden
Tel. 0348-435570
Fax 0348-432493
training@abis.be

Bijlage

XML Column Methode

```
db2 connect to dbeb46
dxxadm enable_db dbeb46
db2 bind "@dxxbind.lst"

db2 create table xmlVisa (id smallint, XMLvisakaart db2xml.xmlvarchar)

db2 insert into db2xml.dtd_ref
values('f:\library\udb\XmlExtender\xmlColumn\visa.dtd',
db2xml.XMLClobFromFile('f:\library\udb\XmlExtender\xmlColumn\visa.dtd
'), 0, 'db2', 'db2', 'db2')

dxxadm enable_column dbeb46 xmlVisa XMLvisakaart
f:\library\udb\XmlExtender\xmlColumn\visa.dad

db2 select * from xmlvisa
db2 select * from visa_uittreksel
db2 select * from visa_lijn_product
db2 select * from visa_lijn_price

db2 insert into xmlvisa (id, XMLvisakaart) values(1 ,
db2xml.XMLVarcharFromFile('f:\library\udb\XmlExtender\xmlColumn\visa1
.xml') )

db2 insert into xmlvisa (id, XMLvisakaart) values(2 ,
db2xml.XMLVarcharFromFile('f:\library\udb\XmlExtender\xmlColumn\visa2
.xml') )
```

De visa DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT visakaart (valid,entry+)>
<!ATTLIST visakaart nr CDATA #REQUIRED>
<!ELEMENT valid (startdate,enddate)>
<!ELEMENT startdate (#PCDATA)>
<!ELEMENT enddate (#PCDATA)>
<!ELEMENT entry (product,price)>
<!ELEMENT product (#PCDATA)>
<!ELEMENT price (#PCDATA)>
```

De visa DAD

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE DAD SYSTEM "E:\dxx\dtd\dad.dtd">
<DAD>
<validation>NO</validation>
<Xcolumn>
  <table name="visa_uittreksel">
    <column name="kaartnr" type="character(10)"
            path="/visakaart/@nr" multi_occurrence="NO"/>
    <column name="startdate" type="character(10)"
            path="/visakaart/valid/startdate" multi_occurrence="NO"/>
    <column name="enddate" type="character(10)"
            path="/visakaart/valid/enddate" multi_occurrence="NO"/>
  </table>
  <table name="visa_lijn_product">
    <column name="product" type="character(30)"
            path="/visakaart/entry/product" multi_occurrence="YES"/>
  </table>
  <table name="visa_lijn_price">
    <column name="price" type="integer"
            path="/visakaart/entry/price" multi_occurrence="YES"/>
  </table>
</Xcolumn>
</DAD>
```

Federated Database

Voorbeeld: DB2 OS/390

```
connect to test;                      <de federated db server>
create wrapper drda;                  <de wrapper>
create server tbd2 type DB2/MVS version '6.1'
      wrapper drda                        <een server wordt aangemaakt
      authorization <user>                  voor een bepaalde wrapper>
        password <paswoord>
        options (node 'node', dbname 'LOCATION_NAME');
create nickname rtutcourses for tbd2.tbaccad.tutcourses;
create nickname rtutsessions for tbd2.tbaccad.tutsessions;

<nickname geeft weer:
   de server naam, TBD2
   de schema naam, tbaccad
   de tabel naam, tutsessions>
```

Voorbeeld: Oracle

```
connect to test;
create wrapper net8;
create server tsta type ORACLE version '8.1'
       wrapper net8
       authorization system
       password manager
       options (node 'TSTA');
create nickname otutcourses for tsta.tbaccad.tutcourses;
create nickname otutsessions for tsta.tbaccad.tutsessions;
```