Beknopt overzicht van enkele nieuwe functies

DB2 for z/OS Version 8

Klaas Brant

De nieuwe release DB2 for z/OS Version 8 is de grootste die het IBM Silicon Valley lab ooit gemaakt heeft; niet alleen wat betreft het aantal regels coding maar ook gemeten in man-uren en nieuwe features. Op dit moment is het presentatiemateriaal voor trainingen af en het vergt minstens twee dagen om alles in redelijke details uit te leggen. Het is dus onmogelijk om in een paar pagina's alle nieuwe faciliteiten uit te leggen en dit artikel beperkt zich dan ook tot de grootste veranderingen of die zaken die enigszins een voorbereiding vereisen.

Zoals gebruikelijk bestaan er omtrent dit soort nieuwe releases ook veel misverstanden, omdat binnen het geruchtencircuit dingen vaak een eigen leven gaan leiden en mensen soms een compleet verkeerde indruk krijgen.

Kernthema's

Re-engineering en Renaissance zijn de kernthema's van versie 8. Zoals men reeds aan de officiële titel van de release kan zien is deze release alleen geschikt voor het 64-bit operating systeem z/OS op het mainframe en niet voor het oudere OS/390. z/OS draait alleen op de nieuwere IBM z/Serie mainframes en aangezien niemand een kloon van deze hardware maakt, heeft IBM de touwtjes weer stevig in handen. Het her-programmeren van DB2 voor 64-bit was niet gemakkelijk en deze release bevat dan ook meer nieuwe en gewijzigde regels coding dan de initiële release van DB2 zo'n 20 jaar geleden.
Natuurlijk had men de coding zo kunnen laten zoals die onder OS/390 (31-bits) was, maar dan zouden we geen gebruik kunnen maken van storage boven de 2 GB lijn. Veel mensen komen nu reeds in de problemen omdat ze meer dan 2 GB nodig hebben. Dus moesten alle algoritmes veranderd worden en was het tijd voor een re-design van de architectuur. Sommige oude zaken, die eigenlijk kunstgrepen waren om met 31-bits toch uit de voeten te kunnen, zijn dan ook verdwenen. Het migratiepad is dan ook van DB2 versie 7 eerst naar een z/Serie machine met z/OS, en daarna pas upgraden naar versie 8. De software upgrade naar versie 8 gaat overigens anders dan bij vorige release-overgangen. De release-overgang doorloopt een aantal stappen die zorgvuldig gepland en bekeken moeten worden.

Maar memory limits zijn niet de enige grenzen die doorbroken zijn in deze release. Op bijna alle plaatsen waar DB2 een fysieke grens had is deze doorbroken. Zie afbeelding 1 voor een complete lijst van verlegde grenzen. Door deze veranderingen is deze release niet alleen zeer ingrijpend voor IBM, maar ook voor de zogenaamde third party vendors die software tools voor DB2 maken. Zelfs een eenvoudig Windows-pakket kan nu velden terugkrijgen van 128 posities in plaats van 18 en kan daardoor in de fout gaan of hopeloos onbruikbaar worden. Dit betekent dat men een goede inventarisatie moet maken van de gebruikte software en goed zal moeten bekijken of men deze aanpassing nodig heeft.

Een andere zeer ingrijpende verandering is het gebruik van Unicode in DB2. De dagen van de EBCDIC character set zijn geteld en intern gebruikt DB2 nu Unicode. Hierdoor zijn wel veel misverstanden ter wereld gekomen. Veel mensen denken dat het nu ook verplicht wordt om in user-applicaties over te gaan naar Unicode. Niets is minder waar. Hoewel een groot deel van de catalog en de interne structuur van DB2 Unicode gebruikt, zal DB2 met de buitenwereld communiceren met de gekozen character set van de interface. Dus een catalog query via een van de standaard interfaces geeft nog steeds een EBCDIC-resultaat. Wel is het uitkijken geblazen om ervoor te zorgen dat DB2 niet constant character sets aan het vertalen is.

SQL Performance-veranderingen

Soortgelijk als DB2 op de kleinere platformen heeft deze versie nu ook de beschikking over Materialized Query Tables (MQT). Deze

Virtual Storage	2^{31}	2^{64}
Table, View, and Alias Name Size	18	128
Column Name Size	18	30
Number of Partitions	254	4096
SQL Statement Length	32K	2MB
Index Key Size	255	2000 (part: 255)
Character Literals	255	32704
Hex Literal Digits	255	32704
Predicates	255	32704
Number of Active Logs	31	93
Number of Archive Logs	1000	10000
Current Optimization Hint	8	128
Character Literals	255	32704
Number of Tables in Join	•	225
Max Sort Key Length	255	16000

Afbeelding 1: De verlegde grenzen van DB2 versie 8.

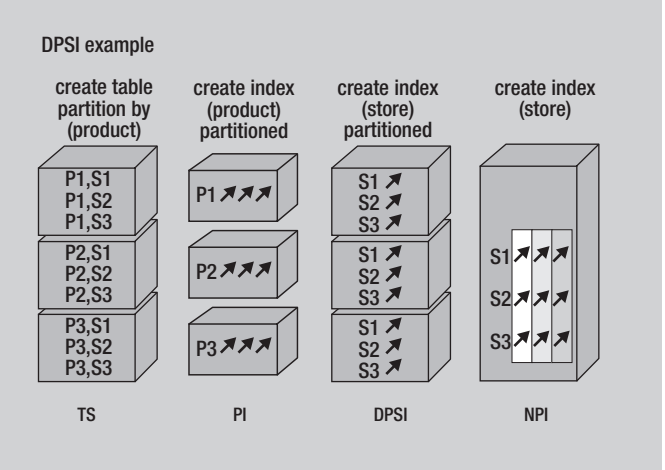
MQT's slaan het resultaat van een (ingewikkelde) query op. De optimizer zal de MQT gebruiken als dezelfde query weer voorbij komt. Het resultaat zal natuurlijk snel onbruikbaar worden als de brongegevens gemuteerd worden, maar in een datawarehouse of operational warehouse kan een dergelijk feature een grote besparing opleveren, zeker als de MQT query dezelfde is als die in een view-definitie.
Een mooie feature is de zogenaamde wide-cursor. Bij een normale cursor komen de rijen een voor een de applicatie binnen (fetch), maar bij een wide-cursor komt bij een fetch een aantal rijen tegelijk binnen. Het resultaat is dan ook een array van rijen. Via een slim mechanisme is het nog steeds mogelijk om de individuele rijen in het array te wijzigen en terug te schrijven in de database. Dit alles natuurlijk ook weer in één SQL call. Ook de INSERT kent nu een dergelijk array insert om in één SQL call meerdere rijen tegelijk te inserten. Door minder met de database te communiceren kan men een hogere snelheid krijgen. Hiermee ligt natuurlijk de weg open om de laatste bottleneck van parallellisme, de single row fetch, ook weg te werken (nog niet in deze release echter).

> Men kan nu detecteren dat door middel van een insert trigger data veranderen

Een ander probleem, dat inmiddels ook door het SQL standaardisatie-comité is erkend, is dat een applicatie die in een database een insert doet, zelf niet weet wat er ingevoegd wordt (denk aan CURRENT TIMESTAMP). Ook is het mogelijk dat een insert trigger nog data van de applicatie verandert. De enige manier om te ontdekken in een applicatie wat men nu werkelijk ingevoegd heeft is de rij terug lezen. Versie 8 neemt een voorsprong op de standaardisatie, door nu reeds een voorstel te implementeren in de vorm van een INSERT INTO. Dit betekent dat men in de applicatie nieuwe velden aanwijst waarin DB2 terug meldt wat er fysiek de database ingegaan is. Het teruglezen kan overgeslagen worden waardoor de performance toeneemt, maar het feature kan ook gebruikt worden om secure applicaties te maken. Immers, men kan nu in de applicatie detecteren dat iemand door middel van een insert trigger probeert de data te veranderen.

Scrollable Cursor

Een nieuw feature van DB2 versie 7 was de Scrollable Cursor. Een mooi feature maar minder bruikbaar, omdat het gebruik maakte van temporary tables om de scrolling te implementeren. Versie 8 maakt het mogelijk om op life data een scrollable cursor te openen. Dit dus zonder een temporary table. Hoe waardevol een dergelijk feature is moet nog blijken, want we programmeren immers onze programma's stateless. Iedere transactie (ook web-transacties via

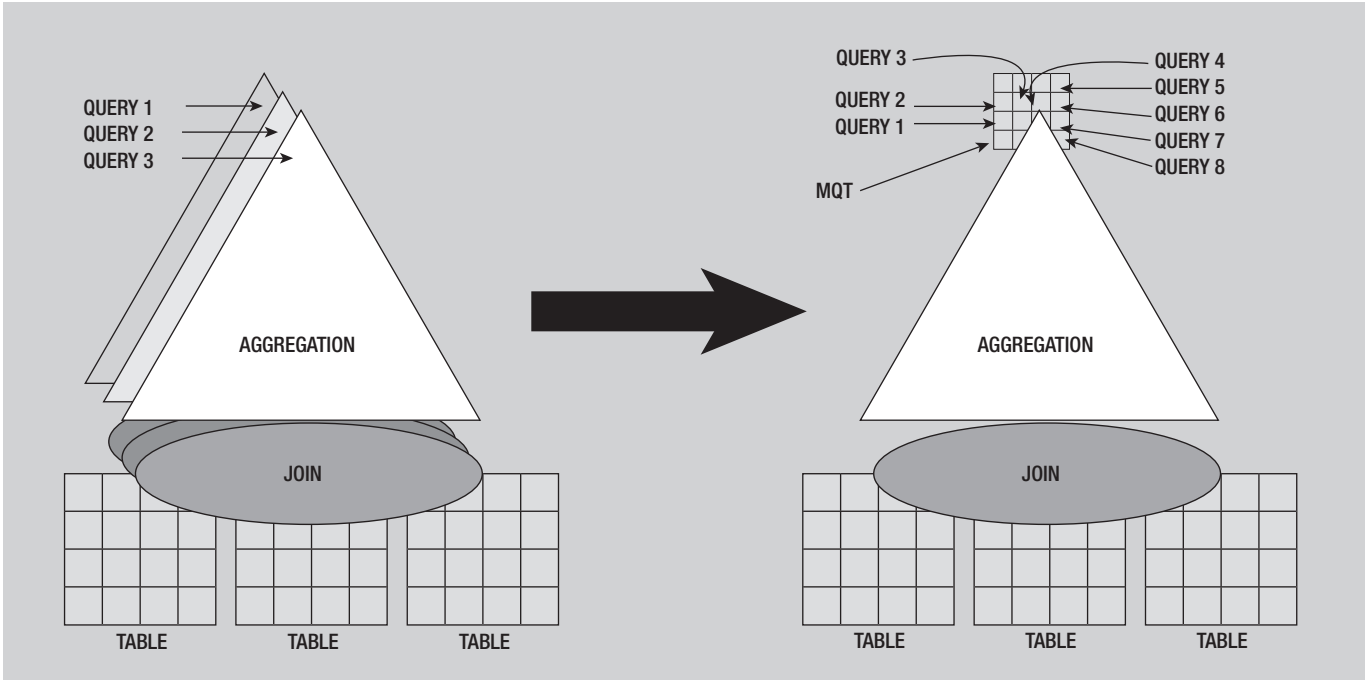


Afbeelding 2: De 'box met goodies'.

websphere) zal na deze transactie eindigen en kan dus geen cursor open houden voor de scrolling. Omgevingen die niet stateless zijn (zoals CICS conversationeel) worden in de praktijk niet gebruikt, omdat ze locking- en resource-problemen veroorzaken indien de gebruiker de interface niet afgesloten achter laat. Ook Windows applicaties halen al hun data op voor scroll boxes (met andere woorden Windows doet de scrolling zelf).
Om slim gebruik te maken van scrollable cursors moest IBM een manier verzinnen om indexes backwards te kunnen scannen. Natuurlijk had men een bi-directionele index kunnen verzinnen zoals die bestaat in de Linux/Unix/Windows (LUW) versie van DB2. Maar IBM heeft in het verleden eerder de vingers gebrand aan nieuwe fysieke indexes (type 2 index in versie 4) en men heeft gekozen voor een implementatie die iedere reeds bestaande index bi-directioneel maakt. Voor de backwards scan wordt gekozen om een pad te volgen via de hoger gelegen pages in de b-tree. Dit is net zo efficiënt als directionele pointers op leafpage niveau (zoals bij LUW), maar wel onmiddellijk bruikbaar op iedere index en er is minder diskspace nodig.

Sequences

Sequences is een van Oracle afgekeken feature en moet het ook gemakkelijker maken om van Oracle naar DB2 te converteren. Persoonlijk denk ik dat men de Oracle Procedure Language moet ondersteunen om in ieder geval een alternatief te zijn. Voor de bestaande DB2-klanten zijn de sequences een perfect alternatief voor de reeds langer bestaande identity columns, overigens ook met veel nieuwe features in versie 8. Beide zijn een soort nummerator waarbij de identity column een waarde toekent bij de insert (denk aan INSERT INTO) en de sequence via SQL om een nieuwe waarde wordt gevraagd.
Het laatste is wellicht wat meer flexibel. Vaak wil men reeds een nieuw ordernummer weten zonder een order aan te maken om maar eens een voorbeeld te noemen. Wat betreft functionaliteit ontlopen de twee elkaar niet veel. De indentity column krijgt in versie 8 veel nieuwe functionaliteit via het ALTER statement. IBM's paradepaardje blijft natuurlijk de optimizer. Ook in deze



Afbeelding 3: Gebruik van Materialized Query Tables (MQT).

release is er weer het nodige aan toegevoegd. Zoals iedere DB2-gebruiker hoort te weten kent DB2 twee niveaus waarop data geanalyseerd kunnen worden: het zeer efficiënte Stage-1 en Stage-2, dat vergelijkbaar is met andere databases. In versie 8 is het mogelijk, als van een SQL predicate de data-types van het programma en de database niet hetzelfde zijn, dit toch te promoten naar Stage-1; DB2 doet de conversie. Natuurlijk kan men zeggen dat men altijd moet zorgen dat een dergelijke mismatch niet ontstaat, maar soms heeft men nu eenmaal niet in de hand wat voor SQL er aangeboden wordt (denk aan SQ-generatoren en 4GL's).

Diegenen die in versie 7 triggers zijn gaan gebruiken zullen ontdekt hebben dat als men gebruik maakt van transition variables, de trigger niet altijd even efficiënt is. De reden hiervoor is dat de trigger een speciale transition table gaat aanmaken, ook als de trigger maar van toepassing is op één of hooguit een paar rijen. Om het geheel een performance boost te geven zal versie 8 gebruik maken van memory caching voor de transition variables en geen table meer aanmaken, als het maar een beperkt aantal rijen betreft.

Availability and Scalability

Een van de grote onhebbelijkheden van DB2 is dat men voor diverse veranderingen aan het database-ontwerp (schema management) een script moet maken. Omdat DB2 geen pending status kent van objecten, zal bij een drop van een table ook alles wat hiervan afhankelijk is (views, triggers, indexen, security etcetera) ook gedropt worden. Hierdoor ontstaat een lang en ingewikkeld script, met unload-drop-create-reload-recreate. De software-leveranciers varen er wel bij met hun tools die de scripts generen. Heeft men niet een dergelijk tool, dan is een dergelijke verandering al snel een ramp.

Hoog tijd dus om het veranderen van databases wat flexibeler te maken. Versie 8 staat zogenaamde online schema changes toe. Vanaf de dag dat dit bekend werd is het een eigen leven gaan lijden. Sommige mensen beweerden zelfs dat alle veranderingen nu on-line mogelijk waren. Ik wil geen domper op de feestvreugde zetten maar het aantal wijzigingen dat men kan doorvoeren is slechts zeer beperkt. Enkele veranderingen die nog steeds NIET kunnen, zijn: drop column, add column zonder attribute NOT NULL, rename column, conversie van column naar willekeurig datatype. Ook bestaat er geen mogelijkheid om iets in een pending status te houden tijdens een drop.

> Een catalog query via een van de standaard interfaces geeft nog steeds een EBCDIC-resultaat

Wat kan er dan wel? Men kan binnen een datatype de kolom vergroten (niet verkleinen). Ook mag de precisie niet kleiner worden (bijvoorbeeld van decimal 7,3 naar decimal 7,2). Om de ramp nog wat groter te maken gaat DB2 vanaf het moment dat men de ALTER heeft gedaan, er een soort versioning op na houden. Ergens is dit wel logisch, want de data zijn er fysiek nog steeds in het oude formaat. Maar de versioning kan veel CPU-belasting betekenen omdat er constant vertalingen zijn. De versioning wordt opgeheven door een reorganisatie die alle oude

data converteert naar het nieuwe formaat. Dus nog even voor alle duidelijkheid, een conversie van character naar datumformaat kan niet omdat dit niet van hetzelfde data-type is. Wel kan smallint naar integer of character(10) naar character(15). Het online schema change is dus verre van flexibel of een perfecte uitkomst. Maar niet getreurd, het ALTER statement heeft meer in petto in deze release. Zo is het mogelijk om de clustering van de data te wijzigen zonder een DROP. Wijzig de clustering index naar NO CLUSTER en wijs een nieuwe clustering index aan, een reorg en klaar is kees. Dit is veel beter dan de oude drop. Overigens mag men meer van on-line schema change verwachten in de toekomst. Maar men moet er rekening mee houden dat de meeste wijzigingen een REORG nodig hebben om de overhead weg te poetsen.

Partitioneren

Het partitioneren is al lang een krachtige feature van mainframe DB2. In deze release is er behoorlijk gesleuteld aan de functionaliteit van partitionering. Een van de grootste problemen was dat als een gepartitioneerde tabel nog meerdere non-partitioning indexes (NPI's) bezat, deze behoorlijk in de weg zaten bij diverse acties.

> Men moet uitkijken met DPSI's of de SQL er niet slechter van wordt

IBM beloofde in het verleden partition independance, maar kon dit alleen maar waar maken als er geen NPI's waren. Zelfs een online REORG moest door middel van een BUILD2-fase de andere partities lastig vallen als het de NPI ging bijwerken. DB2 Versie 8 kan alle indexen partitioneren. Ook de indexen die geen deel uitmaken van de partitioning key. Deze worden dan Data Partitioned Secondairy Indexes of kortweg DPSI (spreek uit: dipsie) genoemd.
Nieuw is ook dat de clustering van de partitioned table bepaald kan worden door de DPSI. Het gaat zelf zo ver dat de partitioning index niet langer nodig is. Immers de keyranges weet DB2 wel zonder index te vinden en de clustering kan dus van een DPSI komen. Moet men onmiddelijk alle partitioned tables gaan ombouwen? Liever niet, want het is niet allemaal goud wat er blinkt. De DPSI's hebben als nadeel dat een non-unique key (erg waarschijnlijk voor een secondairy index) in meerdere partities kan voorkomen, dus in meerdere DPSI's. Dit wil zeggen dat als DB2 een matching index scan wil doen, er meerdere DPSI's gescand gaan worden. Men moet dus goed uitkijken met DPSI's of de SQL er niet slechter van wordt. Dit is de reden waarom men de DPSI onder het kopje availability aantreft en niet onder performance.
Maar dit is nog niet alles voor wat betreft gepartitioneerde tabellen. DB2 kan via een ALTER ook partitions toevoegen aan bestaande partitions. Let op, alleen toevoegen, niet weghalen! Ook het lang verwachte feature Rolling Partitions is in deze release geïmplementeerd. Rolling wil zeggen dat men een data range weggooit en deze vervangt door een nieuwe datarange aan het einde. Stel, men heeft in iedere partitie de gegevens van één jaar. Men heeft 10 partities en dus de jaren 1994-2004 aan data. Als het 2005 wordt kan men nu 1994 weggooien en 2005 toevoegen door de partities te hergebruiken. Het is wel een beetje raar dat de data van 2005 in partitie 1 komen, maar echt een probleem is het niet. Immers, de applicatie weet niet waar de data vandaan komen en de DBA weet gewoon dat partitie 1 2005 en partitie 2 1995 is. Dit alles, samen met het feit dat DB2 nu 4096 partities ondersteund, maakt het partitionerings-feature tot een nog krachtiger feature dan het al was.

Cursor Type	Result Table	Visibility of own cursor's changes	Visibility of other cursors' changes	Updatability (see below about RO)
Non-Scrollable (Join, Sort, etc)	Fixed, workfile	No	No	No
Non-Scrollable	No workfile, base table access	Yes	Yes	Yes
Insensitive scroll	Fixed, declared temp table	No	No	No
Sensitive static scroll	Fixed, declared temp table	Yes (Inserts not allowed)	Yes (Not Inserts)	Yes
Sensitive dynamic scroll	Base table access	Yes	Yes	Yes

Afbeelding 4: Vergelijking Cursor typen.

Conclusie

Zouden alle nieuwe features van DB2 genoemd moeten worden, dan zou men nog een paar uur door moeten lezen en over zoveel ruimte beschikt DB/M niet. Wilt u alle in's en out's leren dan moet u op cursus gaan of de manuals en redbooks bestuderen.
DB2 versie 8 is een fantastische release met erg veel features. De voorwaarden om deze release te kunnen draaien zijn erg hoog (z/serie-machine met z/OS release 1.4). Toch hebben de meeste mainframe sites reeds plannen voor een z/serie-machine of hebben er reeds één of meerdere staan. De prijzen van de hardware gaan nog steeds naar beneden en het mainframe is en blijft daarmee een super-machine als het gaat om groot, krachtig en secure werk. Dus zullen de meeste mainframe-gebruikers wel DB2 versie 8 gebruikers zijn aan het einde van 2005.

Klaas Brant (kbrant@kbce.nl) is DB2-specialist en directeur van KBCE b.v. Meer informatie over DB2 is te vinden op www.kbce.nl en www.db2-times.com