

Sybase ontwikkelde geheel nieuwe kernel

# ASE 15: vrachtwagen vol performance items

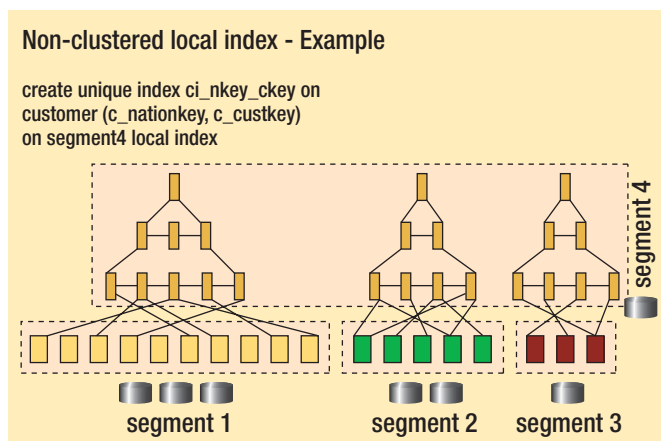
Hans Lamboo

**Eind 2005 introduceerde Sybase versie 15 van de vooral in OLTP-kringen populaire Adaptive Server Enterprise, kortweg ASE. Versie 15 is niet zomaar de opvolger van ASE 12.5.**

Sybase ontwikkelde een geheel nieuwe kernel, met als doel de performance tot ongekende hoogte op te stuwen. ASE 15 bevat niet alleen een geheel nieuwe infrastructuur voor query-processing, maar ook sterk verbeterde eigenschappen voor rapportage en datawarehousing.

**We hebben een compleet nieuwe execution engine en een query optimizer gebouwd**

Performance, dat was de afgelopen jaren het 'Leitmotiv' bij Sybase' R&D-divisie. Vele manjaren aan engineering werden geïnvesteerd in dat wat de opvolger van ASE 12.5 moest worden. Om de gewenste verbeteringen te realiseren, kwam men al snel tot de conclusie dat de hele kernel op de schop moest. ASE 15 is dan ook bijna een geheel nieuw product. Dat is echter niet de verklaring voor het overslaan van de nummers 13 en 14; de reden daarvoor is het ongewenst oproepen van bijgeloof in de voor Sybase belangrijke Westerse (13) en Chinese (14) markt. ASE 15 moet het 'lucky number' van Sybase worden.



Afbeelding 1: Partitionering.

## OLTP

ASE is van huis uit ontworpen en geoptimaliseerd voor OLTP en is dan ook wereldwijd vooral succesvol bij de top van de financiële bedrijven. In de loop der tijd zijn de OLTP-requirements geëvolueerd en er wordt nu ook gevraagd naar rapportages en datawarehousing. Het verschil in toegangspatroon – OLTP raakt slechts aan twee of minder rijen, rapportage aan alle – maakte dat Sybase wel gedwongen was een nieuwe kernel te ontwikkelen. Niet alleen is de ASE 15 dus geschikt voor snelle rapportage, bovendien presteert de OLTP-kant aanmerkelijk beter. "ASE 12.5 had voor een rapportage-query vier minuten nodig en 17 miljoen I/O's. ASE 15 doet hetzelfde in 0,3 seconden en kost 3500 I/O's. Dat komt puur en alleen door de herimplementatie van de hele query processing-infrastructuur", demonstreert Rob Verschoor, engineer bij Sybase EMEA, op zijn laptop. "Maar er is heel veel veranderd. ASE 15 is een vrachtwagen barstensvol nieuw spul."

## Semantische partitionering

Een van de nieuwe features in ASE 15 heet 'semantische partitionering', waarmee met name voor OLTP-toepassingen een grote winst wordt behaald.

"Met name met hele grote tabellen beslaat de hoeveelheid I/O nodig om door de hele indexboom heen te lopen voor elke insert, een relatief groot deel van de hele query," licht Verschoor toe. Om aan de hand van afbeelding 1 nog verder te verduidelijken: "Stel, je hebt een tabel waar we rijtjes inserten. De index is een boomstructuur; elke keer als je een rij insert moet je door de hele index heenwandelen, in de index een rij inserten en dan in de tabel. Tegen de tijd dat je een hele grote tabel hebt van pakweg 200 miljoen rijen is de index misschien wel acht lagen hoog. Wat kun je nu met partitionering doen: je knipt de tabel op in stukjes, elk in een eigen partitie. Door nu een index te bouwen op iedere individuele partitie, blijven de indexboomjes in principe laag. Op het moment dat we met de partitionering beginnen hebben we enerzijds 'het oude spul', in dit voorbeeld 195 miljoen rijen. Vervolgens maken we voor elke kalendermaand een nieuwe partitie aan waarin alle nieuwe transacties zitten. Zo'n partitie wordt niet groter dan, zeg, 1 miljoen rijen. De indexboom die je nodig hebt om die 1 miljoen rijen te indexeren is misschien maar vier of vijf hoog. Dat betekent dat je in plaats van negen, misschien maar vijf I/O's hoeft te doen. In OLTP-termen is dat een enorm verschil."



Rob Verschoor, engineer bij Sybase EMEA: "ASE blijft een transactie-database die hoog volume OLTP kan doen".

Verschoor noemt nog een voordeel van partitionering. "Het draaien van maintenance-commando's op een tabel met 2 miljoen rijen kan dat wel even duren. Met toepassing van partitionering is het duidelijk dat er – in ons voorbeeld – 195 miljoen oude rijen zijn, die niet meer onderhouden hoeven worden. Bovendien kunnen de maintenance-commando's per individuele partitie worden gedraaid. Dat alles scheelt veel tijd in het maintenance window." Ook de optimizer houdt rekening met de partities; aan de opgegeven waarden kan hij zien op welke partities in elk geval niet gezocht hoeft te worden. De individuele partities kunnen ook onafhankelijk van elkaar worden opgeschoond. "De semantische partitionering grijpt direct aan op de wijze waarop de data opgeslagen wordt op disk. Daar merk je met upgrades niets van, want op dat moment hoeft niet alle data te worden geherformateerd. Ga je na de upgrade partities gebruiken, dan is de data-indeling technisch gesproken anders."

Sybase heeft een grote klant in de VS, die deze semantische partitionering aan het toepassen is. "Men gaat daar werken met 367 partities, 366 voor de dagen van het jaar, en één spare", vertelt Verschoor. "Je maakt dan als het ware een soort cirkel-

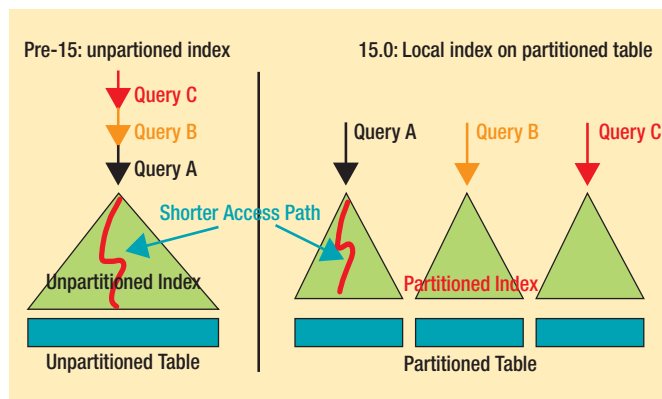
algoritme, de nieuwe data voor vandaag is de partitie van de dag van het jaar van vandaag. De zaken die je niet kwijt schuif je naar nummer 367."

## Datawarehousing en rapportage

Voor deze nieuwe functionaliteit in ASE 15 gebruikt Sybase nieuwe types joins, hash joins, en een slimme optimizer. "Je kunt ook meerdere indexen per tabel gebruiken en dat soort zaken meer", zegt Verschoor. "De tijdwinst zit in het feit dat we twee tabellen joinen op een onmogelijke manier, zonder dat er indexen zijn. In 12.5 moest je dan voor elke rij in de buitenste tabel van de join een scan doen en dan aan de binnenkant. Nu hebben we hash joins, waarbij we één scan op zowel de buitenste als de binnenste tabel doen en deze met een hashtabel aan elkaar knopen. We investeren dus wat memory in die hashtabel, maar daar besparen we miljoenen I/O's mee. Dat is met name voor de rapportagekant heel handig."

## Individuele partities kunnen ook onafhankelijk van elkaar worden opgeschoond

In afbeelding 2 is het verschil duidelijk gemaakt tussen de oude en de nieuwe query-infrastructuur. "In ASE 12.5 had je eerst de normalisatiestappen, permissiechecks en dat soort zaken, de query optimizer maakte vervolgens een query-plan, dat ging naar de execution engine, vervolgens werd de data opgeslagen. Daarbij was voorzien in transactiemangement, join-methodes en speciale commando's. Dat is zoals het altijd gegaan is. Dat plaatje ziet er nu helemaal anders uit", legt Verschoor uit. "We gebruiken nu nieuwe types joins, hash joins, merge joins, nieuwe hashing-trucs en nog meer. We hebben een compleet nieuwe execution engine en een query optimizer gebouwd. Ook op het laagste niveau, de data, is er heel wat veranderd."



Afbeelding 2: Oude en nieuwe query-infrastructuur.

---

Verschoor geeft een voorbeeld. "Stel, je hebt een query 'select color' op 100.000 rijen en tien verschillende kleuren. Wat doet ASE 12.5? Die haalt eerst alle 100.000 waarden op, die worden in een werktabel gestopt. Dan volgt een sorteerslag, die ook de duplicaten eruit gooit. Het resultaat is een rijtje van tien kleuren. Vooral die sorteerslag kost ontzettend veel I/O en dus performance. ASE 15 maakt veel gebruik van hashing algoritmes. Voordat een rij in de werktabel wordt gestopt, wordt hij eerst tegen een hashtabel aangehouden om te kijken of hij er al in zit. Zo ja, dan wordt hij direct verwijderd om duplicaten te voorkomen. Bovendien bevat de query een 'distinct', bijvoorbeeld geformuleerd als: 'delect distinct color from my\_table'. Op die manier wordt de werktabel dus nooit groter dan die tien waarden. De hele I/O verslindende sorteerslag is nu alleen maar nodig om die tien in volgorde te zetten. Dat scheelt dus een karrevracht I/O."

## Webservices, XML en Computed Columns

ASE 15 biedt veel op het gebied van XML en webservices. "SQL, bijvoorbeeld een stored procedure in de database, kan met één commando als webservice worden aangeboden aan de buitenwereld. 'Create service', je geeft 'm een naam en wat parameters, en de webservice kan worden aangesproken via een url", demonstreert Verschoor.

Sybase beschikte vanaf ASE 12.5.1 weliswaar over een goed schaalbare XML processing engine, die echter Java-gebaseerd was en dus niet zo snel. "ASE 15 maakt gebruik van een in C++ geïmplementeerde XML engine. Daardoor kunnen XML documenten worden gevalideerd, op vorm maar ook op DTD", aldus Verschoor. "Die DTD bevindt zich ergens op een webserver. Je kunt een soort syntax aangeven, waarna ASE 15 naar het webadres gaat, de DTD ophaalt, die in de server laadt, en het XML document valideert tegen de DTD. Op die manier wordt gecontroleerd of de XML documenten in overeenstemming zijn met de DTD, voordat ze in de database worden opgeslagen." Iedereen wil graag XML documenten in een database opslaan, maar ze wel kunnen blijven benaderen, bijvoorbeeld op basis van de inhoud van tags. Dat levert nogal wat problemen op, onder meer door de afwijkende structuur van XML documenten. Verschoor: "ASE 15 heeft daarvoor een feature 'Computed Columns'. Zo'n column gedraagt zich als een normale kolom in een tabel maar heeft zelf geen waarde, het is een expressie. De kolommen kun je indexeren en vervolgens kun je die index gebruiken om met normale klassieke OLTP-snelheid de inhoud van het XMLdocument te benaderen. Dus eigenlijk kun je met ASE 15 automatisch een XML document 'shredden'." Voorheen waren XML en webservices-features bij Sybase altijd licentie-opties; vanaf ASE 15 horen deze functionaliteiten bij de standaard-uitrusting.

## Pijnloze migratie

"Het probleem bij een database is natuurlijk, dat veranderingen altijd volledig backwards compatible moeten zijn", stelt Verschoor.

"Een upgrade van 12.5 naar 15 is heel simpel. Dan kan al vanaf de nog ondersteunde versies 12.0 en 11.9, maar ook 11.5 en 11.0 kun je gewoon migreren naar 15. Het migratiepad is pijnloos. We zijn nu ongeveer driekwart jaar verder sinds de introductie, en we hebben inmiddels een aantal klanten die productie draaien met ASE 15. We hebben een klant in Noorwegen die een Business Objects applicatie draait, dat zijn van die query's die dynamisch in elkaar worden gesleuteld en dan op een database worden losgelaten – eigenlijk de nachtmerrie van iedere DBA. Die Noren zijn laaiend enthousiast, die zijn van uren naar seconden teruggegaan in de performance. En dat is puur te danken aan onze nieuwe infrastructuur, want ze hebben aan de applicatie niets veranderd. In Nederland hebben we een telco, die een gedeelte van hun billingsysteem draait op ASE 15. Daar zit een rapportage-stuk in, dat eerst 8 uur of meer nodig had, dat is nu minder dan een kwartier geworden. Ik ben zelf bij een investmentbank in Londen betrokken, waar we bezig zijn met een migratietraject van 12.5 naar 15. Zonder dat er nog veel aan tuning en afregeling is gebeurd, zijn we nu al factor twee tot factor tien sneller.

## Iedereen wil graag XML documenten in een database opslaan maar ze wel kunnen blijven benaderen

Dat varieert, want met name bij BI-applicaties en DWH's kan je veel grotere verbeteringen zien, soms wel factor 100 tot 1000, of zelfs meer. In Zuid-Afrika hebben we een bedrijf dat analyses loslaat op internet tracing patronen, dat is ook typisch iets dat je in een klassieke database niet echt goed kan doen. Ze zijn zeer tevreden over wat ASE 15 daar biedt", aldus Verschoor.

## Tot slot

Betekent dit nou dat ASE 15 Sybase IQ overbodig maakt? "ASE blijft een transactie-database die hoog volume OLTP kan doen", zegt Verschoor daarover. "Aan de andere kant van het spectrum kunnen we nu ook heel behoorlijk omgaan met rapportages, en zaken zoals DSS. Sybase IQ is vooral geoptimaliseerd voor dát stuk en zal nooit echt optimaal OLTP kunnen verrichten. Het gat tussen IQ en ASE is kleiner geworden, maar het is er nog wel. Want IQ kan dingen zoals bijvoorbeeld met een enorme snelheid grote volumes data laden. Ook de indexerings- en opslagstructuur van IQ en ASE verschillen principiële. Ze bestaan dus echt naast elkaar."

**Hans Lambou** is hoofdredacteur van Database Magazine.