Sybase Adaptive Server IQ – een overzicht

Peter Sap – peter@petersap.nl

OLAP met ROLAP?

OLAP applicaties worden in de praktijk vaak geïmplementeerd met een ROLAP oplossing: het vertrouwde RDBMS wordt ingericht als een query omgeving. Hier kunnen minder of meer valide redenen aan ten grondslag liggen, zoals: de kennis van het RDBMS is al aanwezig en/of de leveranciers van het relationele systeem proberen hun product zo aan te bieden dat elk probleem ermee kan worden opgelost. In de praktijk blijkt dit niet altijd zo goed uit te pakken. Veel voorkomende struikelblokken in een ROLAP omgeving zijn te vinden in het transformeren van gegevens naar het datawarehouse of de teleurstellende performance daarvan, met name bij ad-hoc selecties.
Het transformeren van ruwe gegevens naar een voor het datawarehouse bruikbare vorm gebeurt vaak vanwege performance aspecten. Immers, als een telling al tijdens de transformatie heeft plaatsgevonden is het niet nodig om dit steeds opnieuw in het datawarehouse te doen, waar we met enorme hoeveelheden data te maken hebben. Echter, hiermee wordt geen oplossing geboden voor ad-hoc queries, daarvoor blijkt een datawarehouse meestal te groot.
Sybase is een van databaseleveranciers die toegeeft dat met een standaard RDBMS, zoals met hun eigen Adaptive Server Enterprise, deze problemen niet kunnen worden opgelost. Met Adaptive Server IQ (hierna IQ) wordt een RDBMS aangeboden dat specifiek ontworpen is voor datawarehousing en decision support systemen. Hiermee wordt het mogelijk om grote hoeveelheden data ad-hoc te raadplegen en mede daardoor de transformatie naar het datawarehouse (voor een deel) te ontlasten. Vanwege enkele spraakmakende cases staat het product de laatste tijd extra in de belangstelling.
In dit artikel wordt dieper ingegaan op de onderliggende technologie van IQ en worden de meest in het oog springende aspecten voor het voetlicht gebracht.

Het terugdringen van I/O's

De architectuur van IQ is voor een groot deel ingericht op het minimaliseren van de benodigde leesacties (I/O's). In traditionele RDBMS'en wordt de data altijd per rij opgeslagen terwijl dit in IQ per kolom gebeurt. Met name bij ad-hoc queries werkt dit zeer efficiënt. Bij een puur relationeel systeem worden de gegevens rij-voor-rij gelezen en zo kunnen ongewild niet benodigde kolommen meekomen. Op het moment dat gegevens per kolom zijn opgeslagen kan dit worden voorkomen, alleen de strikt noodzakelijke gegevens worden opgehaald. In een datawarehouse waarbij met fact-tabellen wordt gewerkt, die vele miljoenen rijen kunnen bevatten en meestal een groot aantal kolommen, werkt dit sterk performance verbeterend.
Bij een relationeel systeem voor een OLTP toepassing is het volstrekt logisch om met rijen te werken. De vorm van een transactie laat zich goed in rijen uitdrukken en zo worden de gegevens ook weer opgezocht: rij voor rij. Ook de locking mechanismen sluiten hierop aan. Op het moment dat er hoofdzakelijk gegevens worden opgevraagd is het logisch om in kolommen te gaan denken, de andere kolommen in de tabel, die niet worden opgevraagd, zijn dan totaal niet interessant.

Figuur 1 - Opslag van data per kolom

	Nr	Datum	Bedrag	Aantal
Rij 1	1	1 jan 2003	100	5
Rij 2	2	1 feb 2003	150	3
Rij 3	3	1 mei 2003	75	6
Rij 4	4	1 juni 2003	100	2

Als in IQ een selectie op bedrag plaatsvindt, zoals in 'select sum(bedrag) from tabel' wordt alleen de kolom met het bedrag gelezen, ongeacht de wijze van indexering.

Comprimeren

Een tweede manier om het aantal I/O's verder terug te brengen, is het toepassen van een compressie op de data en index-structuren. Voordat gegevens op schijf worden opgeslagen vindt er een comprimering plaats en de-comprimering gebeurt voordat de gegevens vanaf harde schijf weer in het interne geheugen worden geplaatst. De mate van compressie is afhankelijk van de omvang van een pagina en deze ligt tussen de 64 en de 512 Kb. De paginagrootte kan door de DBA worden geconfigureerd.
De comprimering van gegevens levert niet alleen een performance voordeel op, er kunnen ook behoorlijke kostenbesparingen mee worden gerealiseerd. Uit de TPC-H benchmark voor de 100 Gb variant kan IQ, draaiend op een SunFire V480, goed worden vergeleken met de prestaties van Microsoft SQL-Server 2000 op een HP Proliant DL580 G2. Beide configuraties hebben een vergelijkbare performance en prijs/prestatie verhouding, maar waar IQ een disk capaciteit van 407 Gb voor nodig heeft, gebruikt Microsoft een configuratie van 1328 Gb. Dit scheelt ruim een factor drie.
Ook in de literatuur worden de comprimeringstechnieken van IQ beschreven. In [1] worden voor IQ compressie resultaten van gemiddeld 29% genoemd. Dit onderzoek is enigszins wat gedateerd en voor de laatste versie van IQ claimt Sybase inmiddels een gemiddelde van 46% compressie.

Bit-mapped indexen

Op het moment dat gegevens op basis van kolommen worden opgeslagen is het ook nodig om met indexstructuren te gaan werken die daarvoor geschikt zijn. De B-tree index die snel en efficiënt toegang kan bieden tot gegevens die op basis van rijen zijn opgeslagen en frequente wijzigingen op de data goed kan ondersteunen, is in die vorm niet meer voldoende, er moeten andere indextypen bijkomen. IQ kent maar liefst acht verschillende soorten indexstructuren, waarvan sommige bit-mapped zijn.
In de meest eenvoudige vorm is een bit-mapped index een aaneenschakeling van vectoren van bits waarbij elke bit een mogelijke waarde binnen een domein weergeeft. Voor elke kolomwaarde die wordt opgeslagen is er een vector. In figuur 2 wordt dit in een plaatje getoond. In de praktijk zijn bit-mapped indexen aanmerkelijk complexer dan hier wordt geschetst en zijn in staat om ingewikkelde queries snel en efficiënt te verwerken.

Figuur 2 - Structuur van een bit-mapped index

Bij het opslaan van de bedragen 75, 100 en 150 wordt een domein vastgesteld en voor elke waarde een bit gereserveerd.

Bedrag	Bitmap
75	001
100	010
150	100

Vier rijen met respectievelijk de bedragen 100, 150, 75 en 100 kunnen dan als volgt worden gerepresenteerd:

010	100	001	010

In IQ hebben alle kolommen een fast-projection index. Op basis van het aantal verschillende waarden die in een kolom voorkomt, de cardinaliteit, kan deze index zichzelf automatisch opschakelen naar een andere tussenvorm. Afhankelijk van het soort query, de cardinaliteit en het datatype kunnen er extra indexen worden aangemaakt. De bestaande fast-projection index kan dan blijven bestaan en de nieuwe wordt er als het ware bovenop geplaatst.
Vermeldenswaardig is de mogelijkheid om vrije tekst te indexeren. Zo kan er snel naar een bepaald woord worden gezocht. Ook kan een compare-index worden aangemaakt die de relatie zoals groter dan, gelijk aan en kleiner dan tussen twee verschillende numerieke kolommen in dezelfde tabel indexeert. Om de performance voor joins tussen verschillende tabellen te versnellen kan een zogenaamde join-index worden gebruikt. Deze kan alleen voor 1-op-n relaties worden gebruikt en er kunnen complete hiërarchieën mee worden opgebouwd.

Ondanks het grote aantal verschillende typen indexen is de keuze niet zo moeilijk, de regeltjes en aanbevelingen zijn erg overzichtelijk. Verder bepaald IQ zelf welke index voor een query het beste kan worden gebruikt in het geval er meerdere op een kolom liggen. Het is niet nodig om hiervoor statistische gegevens bij te werken.

Snapshot versioning

In een OLAP omgeving zijn er meestal meer gebruikers die gegevens opvragen dan dat er wijzigingen of toevoegingen aan de data plaatsvinden, een read-mostly situatie. Om verstoringen door (dead) locks te voorkomen, die kunnen ontstaan tijdens het bijwerken van gegevens, gebruikt IQ snapshot versioning in combinatie met een aantal eenvoudige regels met betrekking tot locking. Er kan op een bepaald moment slechts een proces een tabel wijzigen en op een manier dat er nooit verstoringen kunnen ontstaan voor een selectie, en het uitvoeren van een selectie blokkeert nooit het wijzigen van data. Deze regels zijn met een eenvoudig en effectief mechanisme geïmplementeerd. De gewijzigde gegevens worden pas beschikbaar gesteld als het lezend proces met een nieuwe transactie start.

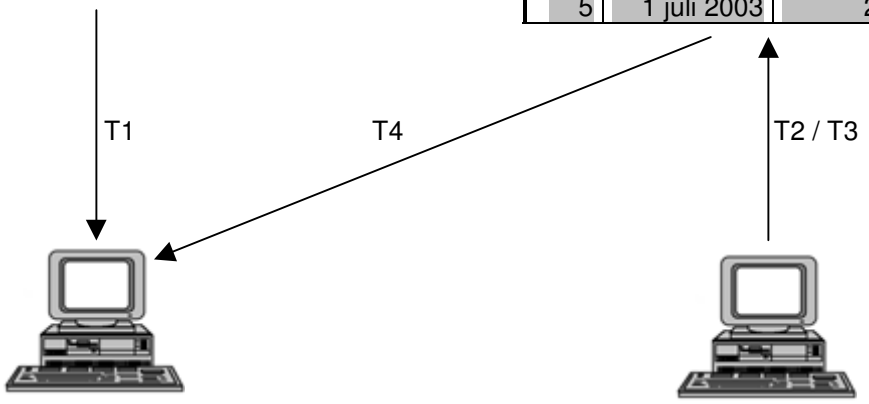
Figuur 3 – Snapshot versioning

Order tabel, versie 1

Nr	Datum	Bedrag	Aantal
1	1 jan 2003	100	5
2	1 feb 2003	150	3
3	1 mei 2003	75	6
4	1 juni 2003	100	2

Order tabel, versie 2

Nr	Datum	Bedrag	Aantal
1	1 jan 2003	100	5
2	1 feb 2003	150	3
3	1 mei 2003	75	6
4	1 juni 2003	100	2
5	1 juli 2003	25	1



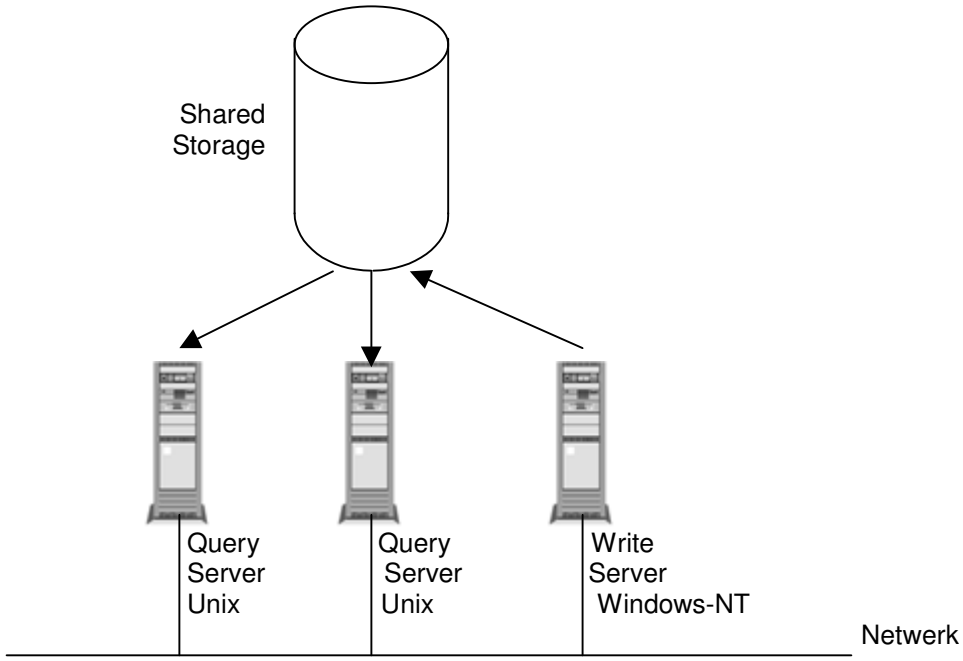
- Op tijdstip 1 (T1) start gebruiker A een transactie en leest de tabel met versienummer 1.
- T2 - Er wordt een transactie gestart door gebruiker B en ordernr 5 wordt toegevoegd. Versie 2 ontstaat. Gebruiker A ziet deze data nog niet omdat zijn transactie nog niet is beëindigd.
- T3 – Gebruiker B geeft een commit.
- T4 – Gebruiker A start een nieuwe transactie en ziet nu ook versie 2, versie 1 wordt door IQ opgeruimd.

Multiplexen

Het terugbrengen van de voor een query benodigde I/O's door gegevens te comprimeren, heeft tot gevolg dat de processor zwaarder wordt belast dan in een puur relationele omgeving. IQ kan geconfigureerd worden om met meerdere processoren in een machine te werken, maar als dat bijvoorbeeld niet meer verder kan worden uitgebreid, kan een multiplex situatie worden gecreëerd. In zo'n opstelling hebben verschillende IQ servers tegelijk toegang tot een gezamenlijke IQ database. Hiermee kan een deel van de last van een server worden overgebracht naar een andere server die zelfs met een ander besturingssysteem kan werken.

Er kunnen zo verschillende mogelijkheden worden gecreëerd: de echt zware selecties worden naar de snelste server gestuurd of het bijwerken van een database gebeurt vanaf een server die daar speciaal voor is opgezet.

Figuur 4 - Multiplex opstelling



Voorbeeld van een multiplex opstelling met twee query servers en een aparte server voor het doorvoeren van wijzigingen of aanvullingen op de IQ database.

IQ in de pers

Er is de laatste tijd redelijk wat publiciteit rondom IQ geweest. Naast de marketing van Sybase is er ook Sun Microsystems die IQ een warm hart toedraagt. Sybase en Sun proberen op deze manier gezamenlijk verder door te dringen in de Very Large DataBase (VLDB) omgevingen. Beide partijen hebben een zogenaamde Reference Architecture neergezet, een soort 'bewezen technologie'. Meer informatie hierover is te vinden op de website van Sun.
Wat minder gekleurde informatie wordt door Bloor Research beschikbaar gesteld die zo enthousiast zijn over de performance dat het rapport wordt afgesloten met 'IT managers should be ready to fall of their chairs'.
Richard Winter, een autoriteit op het gebied van VLDB, heeft een tweetal rapporten over IQ geschreven. Op zijn website is een wat verouderd exemplaar te vinden, maar via de site van Sybase is een recenter rapport te op te vragen. Hierin wordt uitgebreid op de TPC-H resultaten voor IQ ingegaan.

TPC-H Benchmarks

De Transaction Processing Performance Council (TPC) heeft een tweetal benchmarks opgesteld voor decision-support toepassingen, de TPC-H en de TPC-R. Beide maken gebruik van hetzelfde datamodel, hoeveelheid data en zelfs de queries zijn identiek. Bij TPC-R is er echter meer vrijheid om het DBMS in te richten voor de uit te voeren selecties omdat TPC-R bedoeld is om een rapportage-omgeving na te bootsen. Deze benchmark wordt echter weinig gebruikt. De TPC-H is opgezet voor ad-hoc queries. Voor IQ zijn op dit moment een vijftal resultaten bekent voor de TPC-H: 3 voor 100 GigaByte, 1 voor 300 Gb en 1 voor 1000 Gb.
De TPC benchmarks zijn opgezet met een simpel oogmerk: welk RDBMS is de snelste en welk prijskaartje hangt er dan aan. Het lijkt erop dat Sybase er voor heeft gekozen om de TPC-H benchmark voor een ander doel te gebruiken, namelijk om aan te tonen dat ze de goedkoopste zijn.

Dit maakt een vergelijking met concurrerende leveranciers op het punt van de geleverde performance haast zinloos. In de 1000 Gb benchmark gebruikt Sybase een Sun systeem met 8 cpu's, 32 Gb intern geheugen en totaal 3.6 Tb aan disk capaciteit. IBM met DB2 UDB versie 7.2 gebruikt voor dezelfde benchmark een cluster configuratie van HP bestaande uit 32 nodes met elk 4 processoren en 4 Gb intern geheugen (totaal 128 cpu's) en maar liefst 22.7 Tb aan disk capaciteit. Het zal dan ook weinigen verbazen dat DB2 zo'n 10 keer sneller is maar dat de kostprijs van IQ in combinatie met de Sun hardware 24 keer goedkoper uitvalt.

Samenvatting Sybase IQ 12.5 TPC-H resultaten per 1 augustus 2003

Database omvang	Hardware	QphH	Prijs per QphH
100 Gb	Sun SunFire V240	1124	$ 40
100 Gb	Sun SunFire V480	2140	$ 44
100 Gb	Sun SunFire V480	1760	$ 60
300 Gb	Sun SunFire V240	1026	$ 49
1000 Gb	Sun SunFire V880	2240	$ 104

QphH betekend Queries per uur voor TPC-H benchmark.

Het probleem met Sybase

De query performance van IQ is zondermeer uitstekend te noemen, dit blijkt direct uit gesprekken met gebruikers. Opvallend is dat er zeer snel een proces van gewenning ontstaat, gebruikers gaan klagen als een query over een miljard rijen langer duurt dan bijvoorbeeld zo'n 20 seconden. Ook het laden van gegevens in IQ gaat zeer snel, maar blijkt in de praktijk het beste te werken als er grote hoeveelheden data worden aangeboden. Datawarehouses die steeds rij voor rij worden aangevuld, bijvoorbeeld vanuit een replicatiemechanisme of een OLTP toepassing, moeten hier rekening mee houden. Qua architectuur biedt IQ hier weinig flexibiliteit omdat er in een multiplex omgeving maar een server kan zijn die data kan wijzigen.
Om een goede keuze te maken tussen de verschillende index structuren die IQ biedt, is het noodzakelijk om inzicht te hebben in de queries die worden uitgevoerd en het soort data, met name de cardinaliteit. Zonder deze gegevens zal niet de hoogst mogelijke performance uit IQ kunnen worden gehaald. Dit betekend eigenlijk dat IQ niet optimaal kan werken in een pure ad-hoc omgeving. De vraag is of dit in de praktijk ook zo werkt, er zal waarschijnlijk altijd een zekere kennis zijn over het attribuut en over het soort queries, maar dit kan per implementatie behoorlijk verschillen.
Het is jammer dat er nog geen TPC benchmarks door Sybase zijn uitgevoerd die gericht zijn op performance en niet op prijs/prestatie verhouding. Elke leverancier kan zeggen dat zijn product tot 100 keer sneller is dan dat van de concurrent maar het zou mooi zijn als dat ook met een onafhankelijke benchmark is aangetoond. Als er ook TPC-H resultaten voor de grotere datasets van 3 en 10 Tb worden gepubliceerd kan er een beter inzicht komen in de schaalbaarheid.
Tenslotte, het probleem met Sybase zit voor een groot deel in de naam besloten, onbekend maakt nu eenmaal onbemind. De geluiden in de markt en van tevreden gebruikers zijn echter zo luid, dat iedereen die worstelt met een datawarehouse zichzelf tekort doet door IQ te negeren.

Referenties, aanbevolen literatuur en websites:

- [1] J. Goldstein, Compressing Relations and Indexes, ICDE 1998.
- P. O'Neil, Improved Query performance with variant indexes, SIGMOD 1997.
- Sybase website voor Business Intelligence www.sybase.com/bi
- Transaction Processing Performance Council www.tpc.org
- Sun Microsytems www.sun.com
- Winter Corporation www.wintercorp.com
- Bloor Research www.bloor-research.com

Peter Sap (peter@petersap.nl) is een senior database ontwikkelaar/DBA

Augustus 2003