

Hotsos Symposium 2008

Performance, performance

Het jaarlijks terugkerende Oracle performancegerelateerde Hotsos Symposium werd dit jaar voor de 7e keer gehouden in Dallas. Dit driedaagse evenement, dat dit jaar rond de 400 internationale inschrijvingen kende, is uitgegroeid tot het performancesymposium voor Oracle-architecten en databasebeheerders wereldwijd.

De laatste jaren wordt dit symposium ook aangevuld met een volledige trainingsdag. Dit jaar werd de extra trainingsdag ingevuld door Thomas Kyte van de firma Oracle. Afgezien van trends en nieuwe inzichten in performancemethodieken ten aanzien van allerlei Oracle-software en hardwaregerelateerde onderwerpen, schenkt het symposium ook aandacht aan algemenere onderwerpen zoals database change management. Het Hotsos symposium is de laatste jaren snel in aanzien gegroeid, voornamelijk vanwege het goede onderbuikgevoel van de organisatoren ten aanzien van de performancetrends in de markt. De gedrevenheid van de organisatoren en het vasthouden aan hoge kwalificaties met betrekking tot inhoud van een presentatie en de vakbekwaamheid van de presentator spelen hierbij een grote rol.

De presentatoren staan vaak onder druk, doordat de toegestroomde symposiumdeelnemers een gedegen kennis van hun vakgebied hebben en/of zodanig gedreven zijn in hun vakhouding dat slecht onderbouwde en/of niet getoetste uitspraken al tijdens de presentatie ter discussie worden gesteld. Wat op alle presentaties van toepassing is, is de unieke en vernieuwende, vaak praktisch ingestelde en toepasbare zienswijze van het gepresenteerde materiaal. Zelfs bij presentaties die gedreven worden door een hoog theoretisch conceptgehalte, was het goed mogelijk om een brug te slaan naar business- of databasegerelateerde situaties om uiteindelijk hierdoor meer rendement uit de 'thuis'-situatie te halen. Wat Hotsos ook uniek maakt is niet alleen de informele sfeer, maar ook dat sprekers en toehoorders aan elkaar gewaagd zijn. Vaak nadat een spreker zijn specialisme of onderwerp ten toon had gespreid, nam hij of zij weer plaats in het publiek. Door zowel presentatoren als toehoorders werd er in de wandelgangen nog vaak uitvoerig gediscussieerd over de aangedragen ideeën en demonstraties tijdens de presentaties.

Vanuit Nederland waren er dit jaar zes mensen aanwezig, waarvan er drie ook behoorden tot degenen die een presentatie mochten houden. Tot ons genoegen en met gepaste trots konden wij Nederlanders aan het eind van Hotsos opmaken dat alle drie Nederlandse presentaties (Anjo Kolk, Toon Koppelaars, Gerwin Hendriksen) een volle zaal hadden getrokken en veel enthousiaste opmerkingen en discussies te weeg brachten, maar hierover later meer. De Engelse begrippen 'innovation' (vernieuwing) en 'inspire' (inspiratie) staan hoog in het vaandel bij de organisatoren met betrekking tot de doelstellingen voor het symposium. Een van de directeurs van het bedrijf Hotsos, Cary Millsap, behoort zelf tot het selecte internationale groepje dat vernieuwend bezig is geweest met betrekking tot Oracle performancemethodieken. De door hem ontwikkelde "Method-R" is de basis van het bedrijf Hotsos en zijn boek over Oracle performanceanalyse en -verbetering, "Optimizing Oracle Performance" uit 2003 is al jarenlang 'het boek' als het gaat om hoe om te gaan met Oracle performanceproblemen.



Afbeelding 1. Een van de twee presentatiezalen



Afbeelding 2. Toon Koppelaars, RuleGen

Toon Koppelaars – Semantic Query Optimization

Toon Koppelaars mocht maandagmorgen de spits afbijten met zijn presentatie over “Semantic Query Optimization”. De presentatie die een stuk uit zijn en Lex de Haan’s boek “*Applied Mathematics for Database Professionals*” behandelt, ging over het toepassen van semantische query-logica ten behoeve van SQL statement optimalisatie. De theorie gaat er van uit dat twee selectiestatements semantisch identiek zijn als zij hetzelfde antwoord teruggeven, voor elke databasestatus waarin zij zich bevinden, mits zij hierbij voldoen aan een gegeven set van integriteit-constraints. Toon gaf het volgende simpele voorbeeld om deze semantische logica te verduidelijken. Stel dat je een tabel hebt die een ‘geslacht’-kolom bevat. Op deze kolom is van kracht dat deze alleen gevuld mag zijn met de waarden “M”, voor mannelijk en “V”, voor vrouwelijk. Als gevolg van deze conditie mag de volgende semantische transformatie plaatsvinden en zijn beide statements identiek in elkaars semantische betekenis.

```

-- check(geslacht in ('M','V'))

select count(*)
from   tabel
where  geslacht != 'V'

select count(*)
from   tabel
where  geslacht = 'M'
  
```

Het grote voordeel dat hierbij optreedt, is dat een statement getransformeerd kan gaan worden in een semantisch gelijk

ander statement. Deze transformatie, op basis van constraints, kan dan als voordeel hebben dat deze economischer en slimmer kan worden uitgevoerd en/of zelfs helemaal niet meer wordt uitgevoerd daar deze, bijvoorbeeld, NULL als resultante oplevert. Toon demonstreerde, door gebruik te maken van de Optimizer query-engine van Oracle 10g en 11g databases, meerdere semantische methodieken, zoals onder andere: “commutativiteit”, “associativiteit”, regels van “De Morgan” en “involutie”.

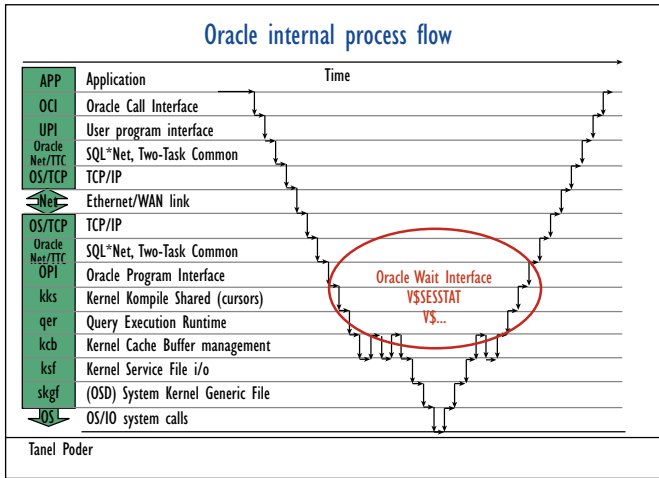
Uit de demonstraties bleek dat de query Optimizer engine van de Oracle databaseversies de meeste semantische logicaregels al volgt en hier ook gebruik van maakt. Soms laat de Oracle Optimizer echter ook verstek gaan, vaak met negatieve gevolgen voor het verwerken van de vraagstelling en het hierna, ophalen van de gevraagde data.

Tanel Pöder - Advanced Oracle Troubleshooting

Tanel Pöder presenteerde zijn verhaal over “Advanced Oracle Troubleshooting”. De ondertitel van deze presentatie was: “No magic is needed, systematic approach will do” en deze weerpiegelde exact de inhoud. Aan de hand van drie volledig uitgewerkte live demonstraties, nam Tanel zijn publiek aan de hand door te demonstreren dat als zelfs de standaard Oracle-methodieken falen, een systematische aanpak zeer effectief kan zijn om alsnog de oorzaak van het probleem te vinden. Tanel had ook een aantal extreme voorbeelden waarin de standaard methodieken, zoals het gebruik van het zetten van een 10046-event of het kijken naar logfiles, je totaal op het verkeerde been zetten of je geen enkel inzicht opleveren. Eén voorbeeld was erg sprekend waarbij de statistieken van de Oracle Wait Interface een compleet ander verhaal vertelde (“het proces is honderd procent bezig”) dan de verzamelde gegevens op het operationele systeem (“het proces doet helemaal niets”).



Afbeelding 3. Tanel Pöder



Afbeelding 4. Het systematisch doorlopen van de proceslagen

Which Index Would You Choose

1 M-C Semifat 2 M-C Fat 3 Ideal ***

Index	ET(QUEBE) Worst input	Side effects
LNAME, FNAME	100 s	
1 LNAME, FNAME CITY	0,3 s	UPDATE CITY +10 ms or +20 ms
2 LNAME, FNAME CNO, CITY	0,1 s	UPDATE CITY +10 ms
3 LNAME, FNAME FNAME, CNO	0,01 s	INSERT, DELETE +10 ms UPDATE often +20 ms

New Index

Afbeelding 5. Verwachte responsetijden op basis van QUBE-methodiek

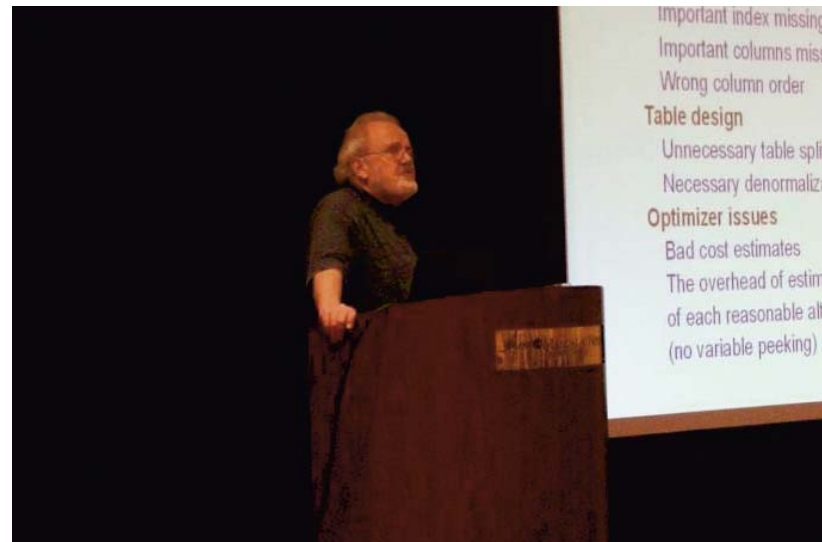
Door het proces te analyseren op elk programmeertechnisch niveau (van database tot op het niveau van het operationele systeem) en door gebruik te maken van het juiste gereedschap, is elk probleem met behulp van zijn methodiek tot in de details te ontleden. Op databaseniveau werd er gekeken naar “profile session wait” en “CPU breakdown”, in verhouding tot de tijd eenheid, met de door hem ontwikkelde Oracle SQL scripting: “sw.sql” en “snapper”. Hierna werden de “profile session performance counters” in verband gebracht met het gegeven “hoe vaak komt het voor” (de “occurrences”), door te kijken naar de Oracle databaseview V\$SESSTAT. Aan de hand van operationele sessie process ‘stack’-informatie werd hierna verder ingezoomd en alles in verhouding geplaatst door middel van programmatuur zoals “pstack”, “procstack”, “procmon.exe”, “procexp.exe” en de “oradebug errorstack” commando’s. Het systematisch doorlopen van alle deelnemende processtappen geeft hierna een volledig beeld van het performanceprobleem en wordt het eenvoudig om de pijnpunten te achterhalen.

Tapio Lahdenmäki – Rethink Your Indexes

Tapio Lahdenmäki had op de volgende dag twee presentatiesesies achter elkaar (ruim twee uur) met als thema “Rethink Your Indexes”. De presentaties gaven een samengevat beeld van een masterclass waarvoor hij normaliter drie dagen reserveert. De presentatie, gebaseerd op zijn en Mike Leach’s boek “Relational Database Index Design and the Optimizers”, gaat ruim in op thema’s zoals adequate indexering, tabelontwerp en Optimizer-afhankelijkheden, dit alles in het licht om zo de performance te optimaliseren op basis van indexering. De presentatie begon met het ter discussie stellen van een aantal heilige huisjes, zoals de stelregel dat er maximaal maar vijf indexen per tabel gedefinieerd mogen worden.

Na een theoretische onderbouwing ten aanzien van fysieke performance-implicaties bij het gebruik van indexen, het afspreken van enkele definities, zoals wat zijn ‘fat’ en ‘semifat’ indexen, werd Tapio’s definitie “Quick Upper-Bound Estimate” (QUBE) behandeld. Deze definitie geeft een richtlijn met betrekking tot de te verwachten ‘Service’ en ‘Waittime’, tezamen verantwoordelijk voor de uiteindelijke ‘respons’tijd bij het inzetten van indexen. Dit is een maatstaf of een index wel of niet de verwachte resultaten zal behalen.

Nog even terugkomend op de begrippen ‘fat’ en ‘semifat’. Een geconcateneerde index wordt ‘semifat’ genoemd als alleen de gevraagde kolommen worden geïndexeerd in de WHERE-sectie van het SELECT-statement. Een geconcateneerde index wordt ‘fat’ genoemd als de index voldoet aan de criteria van ‘semifat’ en hierin de kolommen worden meegenomen die ook in de ORDER BY-sectie worden genoemd. Een geconcateneerde index wordt een ‘Three Star’-index genoemd als de index voldoet aan de definitie ‘fat’ en hierin ook nog eens alle kolommen meeneemt die genoemd worden in de SELECT-sectie. Op basis van deze drie definities, in combinatie met QUBE, kan een uit-



Afbeelding 6. Tapio Lahdenmäki

spraak worden gedaan over de wenselijkheid van een (nieuwe) index ten behoeve van het verbeteren van de datatoegan-lijkheid.

Bij het toepassen van Tapio's methodiek met betrekking tot indexoptimalisatie, tegenwoordig al vaak liefdevol het creëren van "Tapio Indexen" genoemd, wordt performance tot bijna een maximum geoptimaliseerd met betrekking tot uitvoer van het SELECT-statement. In de optimale situatie resulteert dit er zelfs in dat de tabeldata niet eens meer geraakt wordt. De index-structuren worden doorlopen en de gevraagde data die op 'leaf'-niveau verankerd is, wordt hierna rechtstreeks geretourneerd. De presentatie ging verder in op de "wat als...?" vragen met betrekking tot de effecten van indexering bij: datamanipulatie, de invloed van de ordening van de kolommen in de WHERE-sectie, als de ordening van de index afwijkt van de ordening in de tabel en de vraagstelling uiteindelijk een leeg resultaat oplevert. Aan de hand van de gedefinieerde QUBE-richtlijnen werd uiteindelijk bepaald of de datatoegang geoptimaliseerd kon worden en/of een nieuwe geconcateneerde index een beter resultaat op zou leveren. Aan het eind van de presentatie werd nog even stil gestaan bij de performancekosten van een INSERT, UPDATE en DELETE, die grotendeels veroorzaakt worden het groter worden van de indexopbouw ("split") en/of het verplaatsen van indextakken ("move").

Dominic Delmolino - Release Management for Database Application

Dominic Delmolino begon zijn presentatie "Release Management for Database Applications" met zich te verontschuldigen naar al degenen die een RAC-, performance-, optimalisatie- of geheugendump trace-presentatie verwachtten. Zijn presentatie was er echter niet minder boeiend om, in tegen-deel. Zijn presentatie behandelde het vaak in de praktijk verwaarloosde database change management-deel, zoals het vastleggen en het versiebeheer van database DDL (Data Definition Language) statements ten behoeve van bijvoorbeeld tabel- en indexcreatie. Zijn onderbouwing om juist meer te gaan doen aan database change management is het verbeteren van de onderhoudbaarheid van de databaseomgeving met hierdoor positieve effecten met betrekking tot minder performanceproblemen, beter inzicht in de eventuele effecten van wijzigingen en een langer minder stressvol leven voor alle betrokkenen. Of de positieve effecten anders samengevat: sneller, goedkoper, stabiel en efficiënter.

Dominic is een groot voorstander voor het hanteren van één testdatabase. Hierdoor worden eerder codeproblemen zichtbaar met betrekking tot eventuele database-instance synchronisatieproblemen, locking-verschijnselen, het per ongeluk overschrijven van elkaars data, problemen met globale wijzigingen en hierdoor is het vaak mogelijk om met veel grotere hoeveelheden data te werken.



Afbeelding 7. Dominic Delmolino, Agilex Technologies

Een ander voordeel is hierbij de inzet van zijn "global code audit trail" methodiek. Deze methodiek maakt gebruik van de audit-mogelijkheid in de Oracle-database om uitgevoerde DDL-statements van gebruikers en programmeurs automatisch op te slaan. Daar vanwege Sarbanes-Oxley ingezette regelgeving steeds vaker tot gevolg heeft dat auditing-functionaliteit in databases wordt aangezet, is de negatieve performance-impact voor het aanzetten en toepassen van deze auditing-functionaliteit al onderbouwd. Dit maakt de weg vrij voor een snelle inzet van zijn methodiek. Via een slim SQL scriptingsysteem kan men hierna niet alleen alle uitgevoerde DDL- en SQL-statements op de test- of productiedatabase achterhalen, maar ook de historie en wijzigingen van deze statements. Dominic's methodiek blinkt uit vanwege schoonheid en eenvoud. In combinatie met een in de praktijk gehanteerde naamgevingmethodiek en de gepresenteerde DDL- en SQL-statement-auditing wordt het nu mogelijk om gemakkelijk installatie- en upgrade-scripting te genereren. Zoals hij het verwoordde, wordt hierdoor nu de aandacht verlegd richting de gewenste, de te programmeren functionaliteit en minder op de change-management-boekhouding van code.

Gerwin Hendriksen – Another Way of Thinking

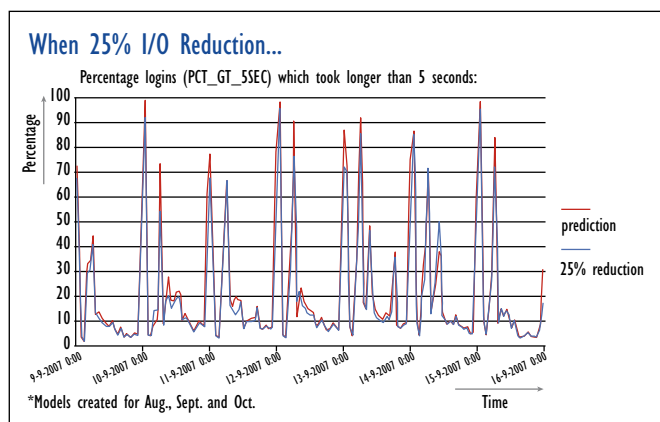
Gerwin Hendriksen presenteerde zijn presentatie "Another Way of Thinking", waarin hij een nieuwe performancemethodologie presenteerde. De methodiek "GAPP" genaamd, is een afkorting van "General Approach Performance Profiling". De methodiek kan door middel van datamining performanceproblemen aantonen in complexe architecturen. Het voordeel van deze methodiek is dat zij softwareleverancieronafhankelijk kan worden uitgevoerd. Deze generieke performanceprofilering kan worden ingezet zonder dat extra op het systeem belastende performance trace-functionaliteit moet worden aangezet en/of geautoriseerd.



Afbeelding 8. Gerwin Hendriksen, AMIS Services

Gerwin legde uit dat de gedachte is gestaafd op het businessproces. Dit betekent dat de responstijd van het businessproces centraal staat. Op basis van metrische gegevens, verzameld uit de totale architectuur, kan bepaald worden welke factoren de meeste impact hebben op de responstijd van het businessproces. De data die worden verzameld, kan bestaan uit allerlei informatie over CPU, I/O, geheugen, netwerk, queues (wachtrijen), SQL-statements, ASH (Active Session History) data, applicatiedata, enzovoort.

De naam “GAPP” werd gekozen daar in gevallen dat daadwerkelijke performance-tracing en/of codewijzigingen in de praktijk niet zijn toegestaan, er een ‘gat’ in de informatie is waardoor het moeilijk is om een performanceprofilering te kunnen maken van het businessproces. Op basis van een zogenaamde “Factorial Analyses” kan een model worden gemaakt om de werkelijke responstijd te voorspellen. Het is hierna mogelijk om, op basis van de gevonden factoren die het meest van invloed zijn, deze te verbeteren en door middel van het model het effect ervan te kunnen voorspellen; zie tekening.



Afbeelding 9. Een “What if...?” scenario bij 25% I/O-afname

“GAPP” maakt op dit moment gebruik van de Oracle database-package DBMS_PREDICTIVE_ANALYTICS, die onderdeel is van de Oracle Data Mining (ODM) functionaliteit. Deze database-package heeft twee procedures: EXPLAIN en PREDICT. De EXPLAIN-procedure maakt het mogelijk de “Factorial Analyses” te doen, en de PREDICT-procedure maakt het mogelijk de voorspellingen op basis van het model te doen. De syntax om de procedures te gebruiken is erg eenvoudig.

Hieronder een voorbeeld voor het maken van een “Factorial Analyses”, met behulp van de database-package.

```
BEGIN
    DBMS_PREDICTIVE_ANALYTICS.EXPLAIN
    (
        data_table_name => 'ghh_model',
        explain_column_name => 'PCT_GT_5SEC_
CHD',
        result_table_name => 'ghh_expl'
    );
END;
/
```

Hieronder een voorbeeld voor het maken van een model, met hulp van de database package.

```
BEGIN
    DBMS_PREDICTIVE_ANALYTICS.PREDICT
    (
        accuracy => v_accuracy,
        data_table_name => 'ghh_model',
        case_id_column_name => 'TIME',
        target_column_name => 'PCT_GT_5SEC_CHD',
        result_table_name => 'ghh_predict_mod'
    );

    DBMS_OUTPUT.PUT_LINE('Accuracy = ' || v_accuracy);
END;
/
```

Voor het toepassen van de methodologie, vermelde Gerwin, is het niet noodzakelijk om deze database-package te gebruiken, hiervoor zou ook een open source-variant kunnen worden ingezet. Het voordeel hiervan zou kunnen betekenen dat “GAPP” gemakkelijker in de praktijk kan worden ingezet, daar deze niet op specifieke Oracle databasefunctionaliteit is gebaseerd.

Cary Millsap – Measure Once, Cut Twice

Cary Millsap hield tijdens zijn presentatie “Measure Once, Cut Twice” een betoog voor het toepassen van de softwareontwikkelmethodiek “Agile” op databasepraktijkhandelingen. Van huis uit is Cary een enthousiaste houtbewerker en de titel van zijn presentatie is afgeleid van de standaardnorm in

deze wereld: "Measure twice, cut once", vrij vertaald: "meet twee keer voordat je hout zaagt". Twee ideeën uit de 'Agile'-methodiek waren zeer toepasbaar volgens hem: incrementeel ontwerp en wekelijkse oplevering van releases. Incrementeel ontwerp was volgens hem zeer toepasbaar en uiteindelijk effectiever, omdat een compleet ontwerp van tevoren vaak gebreken bevat en deze achteraf moeilijk te herstellen zijn. Hij demonstreerde zijn gelijk via een houtbewerkinganalogie - het maken van een houten tafel - en trok deze vergelijking hierna door naar de computerwereld. Het is voor programmeurs nog lastiger, de mensen die volgens Cary echt invloed kunnen uitoefenen op de performance van het geheel, om programmatuur te maken zonder dat zij deze tegen een omgeving kunnen toetsen die nog niet de werkelijkheid weerspiegelt.

Cary Millsap is een voorstander van wekelijkse oplevering om zodoende effectiever toe te kunnen werken naar een eindontwerp. Pas tijdens het ontwerpproces wordt duidelijk wat men, bijvoorbeeld de klant, echt wil. Door het ontwerp in zijn geheel van tevoren vast te leggen, heeft dit vaak als effect dat het eindproduct niet gelijk is aan dat wat de klant wou, er overmatig veel overleg nodig is om alles in banen te leiden en het bijvoorbeeld erg moeilijk is om een stuk interfacing te maken als het eindproduct nog niet is opgeleverd. Een ander voordeel van een wekelijkse oplevering is dat het ontwerp een versimpelijking zou kunnen toestaan, als dit gewenst is. Een totaal uitgewerkt ontwerp maakt dit onmogelijk.



Afbeelding 10. Cary Millsap, Hotsos

Cary beargumenteerde dat deze methodiek veel dichterbij ons dagelijkse leven en hierdoor ook met meer succes ingezet kan worden, ongeacht of dit nu geldt voor het maken van software, performance-eisen bij aankoop van hardware of het maken van een houten ladekast.

Hotsos liet dit jaar zien op eenzame hoogte te staan als Oracle performancesymposium en de goede kwalitatief hoogstaande presentaties. De symposiumganger kreeg een compleet scala aan onderwerpen voorgeschoteld. Het wel en wee van Oracle Real Application Clusters-omgevingen, het demon-

streren van innovatieve performance-tooling, tot aan het demonstreren en uitleggen van nieuwe concepten en methodieken. Terugkijkend had het symposium in 2007 als grootste gemene deler: 'best practices' bestaan niet. Men doet de klant en zichzelf te kort als men te lang vasthoudt aan standaard 'trucjes'. Elk systeem is uniek. Gezond verstand en een goede kennis van zaken leiden tot betere resultaten. Er werden dan ook meerdere heilige huisjes omvergeworpen.

Dit jaar, 2008, stond het symposium veel meer in een conceptueel daglicht met als hoogtepunten het in het zonnetje zetten van twee auteurs met uitzonderlijke boeken die de laatste tijd behoorlijk hun weerklink vinden in de Oracle performancewereld: Toon Koppelaars' met "Applied Mathematics for Database Professionals" en Tapio Lahdenmäki's met "Relational Database Index Design and the Optimizers". Opmerkelijk is dat het materiaal van beide heren generiek toepasbaar is op relationele databases en niet alleen Oracle-database-specifiek materiaal bevat.

Performance is een probleem dat zijn weerslag heeft op de gehele architectuur. Steeds vaker moeten er grenzen worden overschreden: een programmeur die over de schutting meekijkt bij een databasebeheerder en een databasebeheerder die meedenkt en leert van de programmeur of de wensen van een klant/eindgebruiker. Performance is een thema voor iedereen, in alle lagen van de van de organisatie.

Referenties

Toon Koppelaars	http://www.rulegen.com
Tanel Pöder	http://www.tanelpoder.com
Tapio Lahdenmäki	http://www.tapio1.com
Cary Millsap	http://www.hotsos.com
Gerwin Hendriksen	http://www.amis.nl
Dominic Delmolino	http://www.oraclemusings.com

"Relational Database Index Design and the Optimizers",
Tapio Lahdenmaki, Mike Leach
Uitgever: Wiley
ISBN: 978-0-471-71999-1

"Applied Mathematics for Database Professionals",
Lex de Haan, Toon Koppelaars
Uitgever: Apress
ISBN 13: 978-1-59059-745-3
ISBN 10: 1-59059-745-1

Marco Gralike, Amis