



Rezension zu:

Mogens Nørgaard et al.: Oracle Insights: Tales from the Oak Table

Sprache: Englisch

Broschiert – 420 Seiten

Apress 2004

Preis: 35,10 €

ISBN 1590593871

Normalerweise ist die Besprechung von Sammelbänden ein schwieriges Geschäft, nicht aber im vorliegenden Fall, weil es sich bei den Autoren durchweg um Verwandte im Geiste handelt. Sie sind nicht allein seit vielen Jahren im Oracle-Umfeld tätig, sondern sie eint auch mehr oder weniger leidvoll Erfahrung aus diversen Projekten. Das ist einer der Gründe, warum sie sich im *Oak Table Network* zusammengeschlossen haben. Anliegen, Geschichte und Aktivitäten dieses informellen Zusammenschlusses von Oracle-Experten erläutert Mogens Nørgaard in seiner Einführung. Es ist nun nicht der Platz, hier im Detail die 12 durchweg interessanten Beiträge zu besprechen, weswegen ich mich auf einige prinzipielle Feststellungen beschränken muß.

Eine der großen Stärken des Bandes ist, daß nicht einfach Features nebeneinander gestellt, sondern diese in ihren historischen Kontext gestellt werden. Ohne eine Kenntnis der Geschichte kann man die Gegenwart nicht verstehen. Dieser Allgemeinplatz bleibt naturgemäß in den meisten stark technisch orientierten Oracle-Büchern unberücksichtigt. Dave Ensor legt in seinem Beitrag *A Brief History of Oracle* nachvollziehbar dar, wie sich Oracle im Kontext mit der Durchsetzung des relationalen Datenmodells entwickelt hat. Von besonderem, nicht allein theoretischem Interesse ist dabei ein Abriß zur Umsetzung der Codd'schen Regeln durch Oracle. Den größten Raum in diesem längsten Beitrag des Bandes nimmt die durch praktische Probleme induzierte Abfolge der Oracle-Versionen ein. Das ist durchaus mit Gewinn zu lesen, auch wenn man nicht jeder Einschätzung unbedingt folgen muß. Letzteres gilt vor allem für die eine oder andere kritische Bemerkung zu neuen Features wie Real Application Cluster oder Flashback.

Der überwiegende Teil der Beiträge ist den Konsequenzen gewidmet, die die Protagonisten aus häufig eskalierenden Projekten für die Methodik des Performance Tuning gewonnen haben. Das liest sich teilweise sogar recht amüsant, was es aber für die Beteiligten sicher nicht immer war. Durchgängig konzentriert sich die Diskussion darauf, eine Methode für die Diagnostik der Performance zu entwickeln, die das vermeidet, was G. K. Vaidyanatha *Compulsive Tuning Disorder* nennt. Gemeint ist das Insistieren auf irgendwelchen Benchmarks und Hit Ratios. Hier kristallisieren sich zwei Schwerpunkte heraus: Erstens muß man in der Lage sein zu erfassen, wo die (Antwort)Zeit wirklich bleibt. Das erfordert vor allem den Zugriff auf das Oracle Wait Interface. Wenn Oracle schon wie keine andere Datenbank so umfassend dafür instrumentiert ist, dann sollte man diese Daten auch für die Diagnostik nutzen (können). Zweitens darf man sich nicht auf isolierte Bereiche oder Aktionen konzentrieren, sondern muß auf eine „End-to-End“-Analyse orientieren. Das heißt einerseits, die Benutzeraktionen in ihrem logischen Zusammenhang als größeres Ganzes zum Ausgangspunkt der Untersuchung zu machen, und andererseits, nach Möglichkeit die verschiedenen Schichten der komplexen Infrastruktur einzubeziehen.

Einige wichtige Momente dieser Diskussion sollen hier noch speziell herausgestellt werden.

1. wird an verschiedenen Stellen die Bedeutung von Extended SQL Tracing für die Performancediagnostik herausgestellt. (Vgl. dazu im Detail bei C. Millsap und J. Holt: *Optimizing Oracle Performance*, O'Reilly 2003.) Trotz der neuen Möglichkeiten (z.B.

mit dem Active Session History) und dem erweiterten Zugriff auf das Wait Interface (z.B. durch die Warteklassenstatistik) in Oracle 10g sollte diese Methode ihren Platz bei der Analyse komplizierter Performanceprobleme behalten. Nicht umsonst wird durch Oracle 10g mit dem Package DBMS_MONITOR eine unterstützte Schnittstelle für das Extended SQL Tracing geboten.

2. Performancediagnostik, sei es mit Hilfe der v\$-Views oder des Extended SQL Tracing, ist immer mit zusätzlicher Last für das System verbunden. Die Abfrage der Views z.B. ist mit dem Parsen und Ausführen von SQL verbunden und führt unter Umständen zu internen Konflikten beim Zugriff auf den Library Cache. Bei Datenbanken, die ohnehin schon hängen bzw. extrem langsam sind, wird das wahrscheinlich keine brauchbaren Resultate bringen. Einen Ausweg eröffnet der Direktzugriff auf die SGA (Direct Memory Access). Die Grundlagen dafür werden in dem Beitrag von Kyle Hailey sehr gut nachvollziehbar dargestellt. (Vgl. zu diesem Gegenstand auch den Appendix D von R. Shee, K. Deshpande und K Gopalakrishnan: Oracle Wait Interface: A Practical Guide to Performance Diagnostics & Tuning, McGraw-Hill/Osborne 2004.) Natürlich braucht man dann noch eine Applikation, die diesen Zugriff realisiert. Zu diesem Zweck ist in Oracle 10g der direkte Zugriff auf die SGA als Option in die Performancediagnostik mittels des Enterprise Managers integriert.
3. Jeder, der in einer frühen Phase zu einem Oracle-Projekt hinzugezogen wurde, hat sich bestimmt mit der Frage konfrontiert gesehen, ob man nicht mit RAID 5 das teure Storage-System effizienter ausnutzen kann. In dieser Hinsicht sind die Autoren ganz entschieden: Die Einsparung an Plattenplatz erkaufte man in der Regel mit so gravierenden Nachteilen für die Performance, daß die Geschichte im Endeffekt sogar teurer kommt. Da die Protagonisten des Bandes der mit jedem Projekt immer wieder neu zu führenden Diskussion überdrüssig sind, haben sie unter dem Wahlspruch „Enough is Enough“ eine Partei gegründet – BAARF = **B**attle **A**gainst **A**ny **R**AID **F**ive. Was auf den ersten Blick so scherzhaft daherkommt, ist mit einer Vielzahl harter Argumente untersetzt, die unter <http://www.baarf.com> nachzulesen sind. Die Partei steht offen für alle Gleichgesinnten.

Insgesamt läßt sich feststellen, daß es sich um ein sehr lesenwertes Buch handelt, das jedem wärmstens empfohlen sei, der sich für mehr als die technischen Details dieses oder jenes Features oder einer bestimmten Datenbankversion interessiert.

Dr. Frank Haney
Freier Consultant
info@it-haney.de