

“Pitajte Toma” na HROUG-u

HROUG, udruga hrvatskih korisnika Oraclea, svaku jesen začini svojom konferencijom koja je stekla ugled u Hrvatskoj, regiji, ali i šire. Ove godine 18. HROUG konferencija održat će se od 15. do 19. listopada na Crvenom otoku kod Rovinja, a jedan od najzvučnijih gostiju bit će i Tom Kyte iz Oraclea

■ OLEG MAŠTRUKO
i ZORAN JANKOVIĆ

Tom Kyte je Senior Technical Architect u Oracle-ovoj Server Technology Division. Prije nego što je počeo raditi za Oracle, Tom je radio kao sistemski integrator, slažući velike heterogene baze podataka i aplikacije, ponajviše za korisnike iz vojnih i državnih krugova. Tom provodi mnogo vremena radeći s Oracleovim bazama podataka i, nešto specifičnije, s ljudima koji rade s Oracleovim bazama podataka. Dodatno, Tom održava kolumnu AskTom u Oracle Magazineu, gdje odgovara na pitanja o Oracleovim bazama i alatima (<http://asktom.oracle.com/>).

Tom Kyte je i autor knjiga Expert Oracle Database Architecture (Apress, 2005), Expert One on One Oracle (Wrox Press, 2001/Apress 2004), Beginning Oracle Programming (Wrox Press, 2002/Apress 2004) i Effective Oracle by Design (Oracle Press, 2003). Riječ je o knjigama o općem korištenju baza podataka i o tome kako razvijati uspješne Oracle aplikacije.

Nakon ove službene biografije za uvod, kažite neki detalj koji je namjerno ili nenamjerno preskočen?

Jedna od stvari koje malo ljudi čak i u Oracle zajednici zna o meni jest da nisam nikad išao na tečajeve ili škole za programiranje, ili bilo što vezano s računalima, sve

do nakon fakulteta. Diplomirao sam matematiku, a programiranje, baze podataka i sve ostalo što danas znam naučio sam kasnije, uz posao.

Duga veza

Kako je započela vaša veza s Oracleom?

Počeo sam programirati 1987. godine, radeći na IBM *mainframe* računalima i koristeći jezik PL/I. Ubrzo počeo sam pisati aplikacije koristeći SQL/DS na jednom *mainframeu* i DB2 na drugom. Kako sam postajao sve vještiji sa SQL-om, proširio sam svoje korištenje baza podataka na Gupta SQLBase, Ingres, Informix, Sybase i još neke. Igrao sam se Oracleom na IBM DOS sustavima nakon što sam dobio bazu podataka, SQL*Net, forme i reportse na *floppy* diskovima za 99 dolara. Kada je verzija 7.0 objavljena 1992. godine, to je za mene bilo to. Bio sam zaluden. Malo kasnije, u kolovozu 1993.

godine, počeo sam raditi za Oracle.

Oraklašima ste dosta poznati preko usko specijaliziranog Oracle Magazinea i rubrike AskTom. Većina pitanja i odgovora na AskTom.oracle.com sadrži frazu “the best way”, najbolji način. U čemu je trik s pronalženjem tog najboljeg načina?

Točno je da velik broj pitanja sadrži zahtjev ili molbu za najboljim načinom. Sve što mogu reći u vezi s tim načinom jest - ovisi o situaciji. Da postoji samo jedan najbolji način da se nešto napravi, ne bismo izmišljali niti korisnicima nudili više mogućih načina. Npr. ponekad je najbolji način za procesiranje upita u bazu korištenjem indeksa; u nekom drugom slučaju i pod drugim okolnostima najbolji način za procesiranje tog istog upita bit će korištenjem *full scan*. Ovisi, u ovom slučaju, o tome treba li vam prvih par redaka

iz upita što je brže moguće, ili želite doći do zadnjeg retka iz upita što je brže moguće. Najbolji način za optimizaciju tog upita ovisi o tome. Pogledajte <http://bit.ly/12Kmt6B> za puni takav primjer.

Ove godine punite 20 godina rada za Oracle. Što su po vašem mišljenju bile najveće promjene u tih 20 godina?

Najveće promjene dosad bile su s verzijama 7.3, 10g, i 12c. Verzija 7.3 bila je tako velika objava - s mnogo novih mogućnosti - da ima kompanija koje ju drže u operativi i danas, 18 godina nakon izlaska na tržište. Sa 10g implementirali smo AWR i ASH. Bio je to divovski skok naprijed u smislu monitoringa i upravljanja performansama. S Oracle Database 12c vidimo prve velike arhitekturne promjene u implementaciji baza podataka u 25 godina (od verzije 6), korištenjem tehnologije Oracle Multitenant.

Zabavna stvar u radu s bazama podataka jest u tome da se stalno dešavaju promjene - nikada nije dosadno.

Uvijek iste pogreške

Jedna od vaših poznatijih prezentacija zove se “Što radimo krivo?” Možete li dati par kratkih primjera za naše čitatelje, prije predavanja na samom HROUG-u?

Iste osnovne pogreške ponavljaju se opet i uvijek iznova, godinu za godinom. Računalni tečajevi i studiji - čak i oni posvećeni bazama podataka - ponekad ne poučavaju osnovama, posebno kad je riječ o bazama podataka. Velika greška koju još uvijek vidim i u 2013. godini jest

nedostatak *bind* varijabli u aplikacijama. Nekorištenje ili neispravno korištenje *bind* varijabli, po mom iskustvu, uzrok je broj jedan velikih problema s performansama. Dodatno, nekorištenje *bind* varijabli neizostavno vodi problemima sa SQL *injection* u aplikacijama, što je pak sigurnosni bug broj jedan u današnjim aplikacijama.

Još je jedan primjer postavka da će sve uspjeti. Primjećujem da ljudi pogrešno rade *exception handling* uvijek iznova, što dovodi do problema s integritetom podataka i

TRANSPARENTNO PREŽIVLJAVANJE

Application Continuity u verziji 12c posve je nova mogućnost koja se dosta tiče developera?

Da, Application Continuity omogućuje transparentni *failover read-write* transakcije u klasteriranom okruženju. U verzijama 11g i ranijima baza podataka mogla je obaviti transparentni *failover read only* aplikacije u slučaju pada čvora, no ako ste provodili *read-write* operaciju, vaša bi se aplikacija transparentno spojila na drugi čvor, ali bi tada dobila poruku o pogrešci. Vaša bi aplikacija bila prisiljena restartirati transakciju.

Počev od Oracle Database 12c, vaša aplikacija može biti konfigurirana tako da umjesto primanja poruke o pogrešci po padu čvora baza podataka može obaviti *failover* transakcije, transparentno je restartirati, i tako omogućiti da transakcija aplikacije preživi pad transparentno.

jednostavno pogrešnih rezultata aplikacija. Ovi su problemi vrlo teški za debugiranje jer je programer, da tako kažemo, sakrio problem.

Oracle 12c - posljednja generacija tehnologije baza podataka - postoje li novosti i poboljšanja zbog kojih zaslužuje naziv esencijalnog koraka naprijed?

Oracle Database 12c velik je korak naprijed. Imamo stotine novih mogućnosti (v. http://docs.oracle.com/cd/E16655_01/server.121/e17906/toc.htm, odnosno New Features Guide). Tu je i nova arhitektura baze - Oracle Multitenant. Godinama smo radili na tome.

Među novim mogućnostima spominje se i "još bolji PL/SQL iz SQL-a", kako ste vi to kazali; što mislite pod tim?

Još od verzije 7.2 imamo mogućnost pozivanja PL/SQL-a iz SQL-a - u principu, možemo proširiti SQL jezik korištenjem PL/SQL-a, implementirajući funkcije koje Oracle nije implementirao. Ovaj postupak uključivao je određeni dodatni teret (*overhead*), ali bio je moguć. Također, trebao vam je *create procedure* i trebali ste instalirati svoj kod u bazu podataka. Počevši od Oracle Database 12c, možete inkorporirati PL/SQL izravno u SQL *statement*. Ovo smanjuje dodatni teret pozivanja PL/SQL-a iz SQL-a i ne zahtijeva da predinstalirate svoju PL/SQL funkciju u bazu podataka.

Još jedna zanimljiva nova mogućnost jest Adaptive Execution Plan?

Da, ovo omogućava bazi podataka da iznađe "mnogo planova u jednom planu" - tako da može pronaći pravi plan u prvom pokušaju.

Oracle Multitenant Architecture u par riječi?

Teško je odgovoriti na ovo u par riječi - riječ je o prilično velikoj novosti. Kada je predstavljam u prezentacijama, primijetio sam da mi treba 45 do 60 minuta da je detaljnije objasnim! Stoga ću vas uputiti na Patricka Wheelera, *product managera* za Multitenant i njegov odličan tekst na ovu temu: https://blogs.oracle.com/Multitenant/entry/introducing_oracle_multitenant. @

