Weather Tough Times with NoCOUG

Spotlight on Oracle

Candid questions get candid answers from Dr. Bert Scalzo.
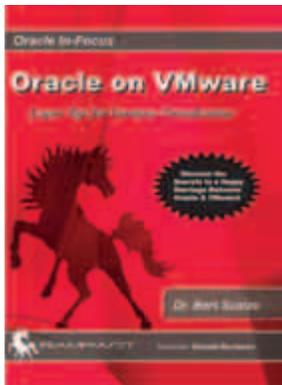
See page 4.

Oracle on VMware

Brian Hitchcock reviews Dr. Bert Scalzo's latest book

See page 6.

Bolt-On Power

Chris Lawson discusses the perils and pitfalls of RAC.

See page 18.

Much more inside . . .

Spotlight on Oracle

Bert Scalzo

Bert Scalzo has worked with Oracle databases for well over two decades, starting with version 4. He has an extensive academic background, including a PhD in computer science. He has written six books on Oracle: Oracle DBA Guide to Data Warehousing and Star Schemas, TOAD Handbook, TOAD Pocket Reference, Database Benchmarking: Practical Methods for Oracle & SQL Server, Advanced Oracle Utilities: The Definitive Reference *and* Oracle on VMware: Expert Tips for Database Virtualization.

Why Oracle? Enterprise Edition with Partitioning now costs $59,000 per CPU. The annual support fee is 22% of the purchase price. MySQL has partitioning and it's free.

I always answer this question using a hopefully clever wordplay of an old *Saturday Night Live* skit: "Oracle been very, very good to me." I chose databases and Oracle way back, and mostly stick with what I think I know. I've presented papers at the MySQL conference—but until their SQL optimizer matures, I'll stick with Oracle's proven track record, maturity, and scalability. But I'm watching MySQL, SQL Server and even PostgreSQL—because things can always change.

My experience with RAC has been bumpy. Is it just me? Is RAC a bane or a boon?

RAC is a great technology. But RAC requires teaching us "old dogs" some new tricks. We cannot simply apply SMP-based Oracle rules of thumb to RAC and expect the same results. I've seen numerous RAC "proof of concept" tests fail from this, and thus people sticking with their comfort zone.

But I've yet to see a successful RAC deployment need to revert. Done right, RAC rocks. Yes it's more complex and requires mastering new things. Just think of it as job security. I do, however, recommend Quest Software's Spotlight for RAC, because it makes RAC architecture visualization and the optimization identification process more straightforward.

A company I know has been using Oracle 10g for a long time. Is it time for them to upgrade to Oracle Database 11g? Their database is very stable and performance is rock solid. They don't have a support contract anymore and don't seem to need one. The application was written in-house.

That's a tough question. The technologist in me wants to say 11*g*—because I like to stay current. The pragmatic consultant in me says don't fix what ain't broke. But I never advise anyone to skimp on Oracle support—and no, I'm not an Oracle stockholder. MetaLink access alone can save many hours' time and real dollars. While Internet searches are very powerful and can find useful stuff—critical wisdom and knowledge resides exclusively within MetaLink. It's worth every penny. Furthermore, as long as Oracle offers new patch sets for your database, it's worth being able to evaluate and deploy them.

In your latest book, Oracle on VMware, you make a case for "solving" performance problems with hardware upgrades.[1] Is Oracle too hard to tune, then?

No—Oracle is a great database with extensive flexibility. Instead, it's the cost of tuning (whether for database, operating system, and/or application) that has become somewhat cost ineffective. Let me give an example. A decade

---

[1] Editor's Note: Here's the full quote from Dr. Scalzo's book: "*Person hours cost so much more now than computer hardware even with inexpensive offshore outsourcing. It is now considered a sound business decision these days to throw cheap hardware at problems. It is at least, if not more, cost effective than having the staff [sic] tuned and optimized for the same net effect. Besides, a failed tuning and optimization effort leaves you exactly where you started. At least the hardware upgrade approach results in a faster/better server experiencing the same problem that may still have future value to the business once the fundamental problem is eventually corrected. And, if nothing else, the hardware can be depreciated, whereas the time spent tuning is always just a cost taken off the bottom line. So, with such cheap hardware, it might be a wiser business bet to throw hardware at some solutions sooner than was done in the past. One might go so far as to make an economic principle claim that the opportunity cost of tuning is foregoing cheap upgrades that might fix the issue and also possess intrinsic value. Stated this way, it is a safe bet that is where the business people would vote to spend.*"

ago, tuning the SGA memory allocation and usage was critical—and it still is today. But with dirt-cheap memory, servers that can accommodate tons of memory, and Oracle automatic memory management—just buy more memory. It's often far cheaper to add 16 GB RAM than pay two days' expert database consultation fees. Yes, there are some scenarios where simply adding hardware might make a database bottleneck situation worse. But the odds and cost are now such that it's a worthwhile first step—since you can always tune if that fails. I'm just preaching to try cheap and easy solutions first.

I make a good living adding hints to Oracle queries to improve performance. Does the optimizer really need a hint or is it something the developer or DBA is doing wrong in the first place?

That's a loaded question. People generally fall into one of two camps: hints are okay and hints are the devil's spawn. So picking one side or the other means 50% of the people will disagree with me, and trying to find a happy middle ground will mean 100% of the people think I'm nuts. Personally, I'm not a huge fan of hints—I'd rather rely on 11*g*'s outstanding plan management features (profiles and baselines). For me, even 10*g*'s stored outlines are often preferable. But I nonetheless often have to use hints to find the needle in the SQL optimization haystack for whatever reason. When I do, I rely on Quest Software's SQL Optimizer. Okay, now I sound like a sales guy—I've mentioned two products from the company I work for. But we used to do a booth drawing where we displayed a SQL statement on a banner and asked people to guess how many rewrites could be done. No ones' guesses ever came close to the 39,000 actual rewrites possible. Any tool that helps to generate and filter through all those SQL coding possibilities and performance ramifications is worth a look. It even suggests structural changes to further extend that research. Do note that I did not mention my tool—Toad. So while I may have mentioned some Quest tools, I did not mention my favorite and my baby. I hope that this helps to retain some of my credibility!

Is 24x7 really possible? Personally, I've never seen a site that was truly 24x7, and I'm inclined to believe that it's a myth. Is MAA the answer? If it is, it sure looks expensive.[2] Is there a cheaper alternative?

This is an easy question—heck, no. Nothing is life is absolute. There are few universal truths. So 24x7 is not possible—not even with RAC, Data Guard, and redundant hardware. Remember, the Titanic was unsinkable—and look what publicizing that got them. But we technologists work for the business people—they are our customers. Bear in mind that we (technologists) are just overhead. The customer (who is always right) says they are a 24x7 business. So we have to find an SLA that can satisfy their demands/needs. Look at Internet hosting companies. They generally advertise something like 98.5%

uptime. That's what we too should strive to offer. Then based on their reply, we know what hardware, software, Oracle options, etc., we need to deploy to meet those requirements. Then monitor and improve as needed.

My manager keeps nagging me to get certified. Certification requires the purchase of at least one instructor-led training course. For the price of a five-day training course, I can buy dozens and dozens of good books. I tried to convince my manager to give me a week off to spend at the beach, reading a good book like Effective Oracle by Design *by Tom Kyte, but he wouldn't bite. My manager doesn't take my opinion seriously but he'll listen to you. Is he right about this or am I?*

If you look at my resume, I don't really have any current Oracle certifications. So my answer may seem like sour grapes—but I'm not a fan of certifications. When I got my vision corrected (i.e., Intra-Lasik), I looked for someone who had done it a lot and that pro athletes and other doctors went to. Thus my personal selection criteria were competence and acknowledged experience—not cost or academic credentials. I believe the same in our industry. I have a PhD, but that means nothing to people who hire me. It's what I've done and what I know that counts—not the paper on the wall. But meet your manager halfway, agree to attend the next expert training session for certification held in Hawaii or on a cruise ship. Don't laugh, such events do exist. As for Tom Kyte, you are spot on—his books and blogs are pure gold.

Thanks for answering all my cheeky questions today. I'm always interested in buying a good Oracle book. Do you have some personal favorites that you can recommend?

There are tons of people writing Oracle books these days—and four whom I've worked with personally and respect (i.e., I learned a lot). For anything PL/SQL related, Steven Feuerstein is the man. For general DBA texts, Mike Ault is a safe bet. For RAC specific, Murali Vallath is second to none. And for anything performance or tuning related I look to Guy Harrison. In fact, Guy has a new book coming out next year that will become the best single reference book for any Oracle professional. I also like Cary Millsap—especially for his Method R trace file tuning approach. But I've not had the privilege of working with Cary; I've just attended his sessions and classes, and have had numerous lunches with him. There are lots of other great Oracle authors—so don't restrict yourself to my short list. ▲

---

[2] **www.oracle.com/technology/deploy/availability/htdocs/ maaoverview.html**.