Official Publication of the Northern California Oracle Users Group

NoCOUG JOURNAL

Vol. 20, No. 1 · FEBRUARY 2006 $15

Adapt to Changing Times with NoCOUG

The Present and Future of Database Administration

An interview with Venkat Devraj. *See page 4.*

Hello, World!

A new column on programming languages. Guest columnist Catherine Devlin introduces us to Python. *See page 13.*

Book Review

Our regular columnist Brian Hitchcock reviews a thought-provoking book, Oracle Insights: Tales of the Oak Table. *See page 18.*

Big discount on Jonathan Lewis seminar! See page 27.

Is this your last issue? See page 27.

Much More Inside . . .

The Present and Future of Database Administration

An Interview with Venkat Devraj


Venkat Devraj

We are pleased to have the opportunity to introduce NoCOUG members to Venkat Devraj, author of *Oracle 24x7 Tips and Techniques*. He is working on a new book—on a hot new subject—titled *Building Scalable Systems with Autonomics*. Venkat is the founder and chief architect of ExtraQuest Corporation, which is based in Denver.

Tell us a little bit about your background. Where did you study, and how did you become so knowledgeable about Oracle?

I started working full time when I was 14 and ended up in a job that required me to program heavily in C. I worked my way through school, acquired a bachelors' in business and finance, and then signed up for a masters' program in software engineering. During that time, I found myself going from flat-file systems built in C and COBOL to Xbase environments and then to RDBMSs. I started off in Oracle 5.1 as a developer. The mid-'90s saw me migrating to the systems, network, and database administration side of things.

Wanderlust eventually took over, and I ended up in a project in the Middle East. After that project, I moved to the U.S., working as a performance and disaster recovery consultant for several companies.

You also seem to have some entrepreneurial spirit, founding a couple of companies. How do you account for this? Is it in your blood?

For me, it's never been about building a company for putting lots of money in my pocket. I repeatedly saw huge inefficiencies in the way database administration was done and decided to do something about it.

DBAs are busy fighting fires and, unfortunately, ignoring what they really need to be doing. They work hard, but often end up directing their energies at the symptom rather than the cause. If an application has database problems after it goes live, that to me clearly means the DBA didn't have the resources to get the project done right. In the end, this is not a DBA problem—it's a managerial problem.

DBAs are busy fighting fires. In the end, this is not a DBA problem—it's a managerial problem.

Tools vendors provide a plethora of "point solutions" catering to the low hanging fruits in the DBA workload—such as monitoring and alerting— which only foster reactive work patterns—repairing mistakes of the large DBMS vendors—mistakes that should never have been there in the first place. Companies worldwide pay for these mistakes every day by retaining an army of DBAs to baby-sit the database.

I saw that this prevailing juggernaut could not be effectively controlled via the existing DBA methods and tools. If the typically reactive database administrator (let's call this role "DBA-1") needs to get ahead of the curve and transition into more of a proactive database architect (call this "DBA-2"), they need to separate and offload the lower-level mundane tasks and firefighting that the DBA function had ended up morphing into and, instead, focus on higher-value tasks.

Tasks such as business needs analysis, data type and volume analysis, systems design and architecture, configuration optimization, performance management, business continuance planning and testing, workload analysis and segregation, application/database interplay design, security and compliance audits, and what-if analyses are creative things that are better done by these thinking individuals that have the capability to absorb the institutional knowledge in their environments and use it to build better databases.

The lower-level functions could then be outsourced or, better still, automated so the business is not impacted in the short term by this role change.

Sometimes I see an even worse problem. DBA managers make the DBAs take on system, network, or application administrative tasks or even function as developers. This only compounds the problem such that instead of just dealing with fires on the database front, they now have to choose which fire is the largest and, in the process, go even farther away from any proactive environment optimization.

Many of the people subscribing to this ineffective structure have since been rudely awakened when the new economy bubble burst. IT fell off its pedestal and came under intense scrutiny. Because the bar was low, the babysitting jobs started going offshore, where any trained engineer with a couple of years' experience could take over the job that our DBA-1 had been doing for the last decade—without a noticeable impact to quality. There are companies, both small and large, that have recognized this and have taken the right measures to optimize their structure and job functions; however, they are exceptions to the norm.

Let's talk about your book* Oracle 24x7 Tips and Techniques. *What led you to write it?

I found myself tasked multiple times with building a highly available database environment. I stumbled my way through this process many times, until I got pretty good at it. I learned that building a highly available database environment is not just about deploying a single magic bullet such as Parallel Server/RAC or Advanced Replication. It is a series of processes and configuration techniques that you need to embed in your approach.

While many portions of that book continue to remain relevant even today, what really excites me today is the ability to use features built within the DBMS kernel, well-written DBA-1 scripts and techniques and advanced analytics to take database administration to the next level—a more creative manner in which DBAs can get engaged to add more value to the business.

My next book discusses the momentum that we are gaining in this new area and what DBAs can do differently in their day-to-day jobs to leverage that momentum more effectively, rather than continue fire-fighting and watch their jobs being offshored, often to less experienced, but equally qualified individuals overseas. They need to creatively raise the bar by leveraging newer technology that is waiting in the wings to be adopted to enhance individual job satisfaction, as well as value to the business. My new book attempts to explain how to get from DBA version 1 (DBA-1) to DBA version 2 (DBA-2).

Is 24x7 really a myth?

24x7 is not about throwing together a system and then doing maintenance on it to keep it running forever. On the contrary, achieving 24x7 takes a good amount of prep time to understand and quantify business requirements and expected service levels, define standards that are appropriate for that environment, build a business case to identify the right budget based on such requirements versus the cost of downtime, assembling the right software and hardware components, then configuring them correctly to meet the business needs, and then testing the heck out of them to ensure they meet expected standards. 24x7 is largely established even before a system goes live. Getting a sub-optimally constructed live system to be 24x7 is a difficult endeavor and often results in failure and job loss.

Isn't 24x7 very expensive and out of the reach of most organizations?

I would argue to the contrary. There are many simple things that DBAs can do and attain much higher availability, performance, and scalability than they thought possible—if only those DBAs had the time to take a step back and think about it. Unfortunately, they are too busy dealing with the mundane. So 24x7 often seems daunting.

The lack of standard operating procedures in database administration astounds me.

At the end of the day, "very expensive" is a subjective term. If you are a financial clearinghouse and you stand to lose a couple of million dollars for every hour of database downtime, would you think spending $3M in robust infrastructure is "too expensive"? Anyway, expensive synchronous geo-mirroring is no longer the only option to get to 24x7. With some of Oracle's (and DB2 and SQL Server's) newer capabilities, 24x7 is within the reach of most companies—provided they have a legitimate business need to attain 24x7 and are willing to invest the time and resources to do it right the first time.

Isn't human error the major cause of outages? What advice do you have in this regard?

Yes, unfortunately in my limited experience, human errors do seem to account for the majority of outages. The lack of standard operating procedures in database administration astounds me. What astounds me even more is that people are content to keep DBA work shrouded as a black art. But business just cannot continue to operate that way. Oracle Corp. had an interesting observation in one of their recent ads, where they said a huge percentage (80%, I think) of outages are caused by human errors. That's why we need to work harder and smarter to automate our day-to-day tasks. As DBAs, whether we know it or not, whether we accept it or not, the fact is, most of our day-to-day workload is mundane and is automatable. The sooner we embrace the facts and actually get these low-level tasks standardized and automated, the less human intervention we would have and the lesser the chances of human error. Anything that's repeatable or complex needs to be automated, so we can move on to projects that have been on the back burner.

Last year, you made a presentation on "autonomics" to the Denver chapter of the IEEE. It is also the topic of your next book. Would you briefly explain the subject to our readers?

Autonomics brings together an expert system with the operational procedures to interpret the current state of an environment and automate its care and feeding both proactively and reactively. At an abstract level, it is meant to simulate an experienced administrator capable of gathering and analyzing facts and thinking on her feet.

Let's take some common examples to understand this better: when you get a cut and start bleeding, the blood begins to thicken and clot to avoid further loss of blood. When you haven't eaten in a while, your body goes into conservation mode. These are natural reactions to stimuli built inside every normal human body and different parts work in tandem to achieve the desired response. Autonomics attempts to replicate this model to proactively carry out tasks that nurture and nourish the system, as well as select the best way to respond to a given stimuli caused within the environment (inside or outside the database). The operational actions and reactions are hardwired in the DNA. A good autonomics system attempts to build a baseline of environmental patterns by collecting key performance indicators and understanding acceptable deviations. This allows it to identify real problem patterns as significant deviations from the baseline emerge.

The thing that really hurts our company is employee turnover in India.

What support does Oracle have for autonomics, if any? What about other database engines such as DB2/UDB and SQL Server?

As far as I can see, newer versions of Oracle including 10g are getting better and better at one type of automation—self-repair—and so are DB2 UDB v8+ and SQL Server 2005. Such self-repair automation also allows the DBMS to repair itself when it runs into certain types of predetermined problems. They are also reducing the need for maintenance and downtime and, overall, are heading in the right direction.

For a long time, Larry Ellison has been predicting that database administrators will become extinct. Will autonomics hasten that day?

Yes and no. DBA work in the U.S. as we know it today (i.e., the DBA-1 work) is sure to be offshored more and more in the short run and will taper off in the mid to long run because autonomics will replace it. It's a natural evolution.

However, that will still not eliminate the need for a different breed of DBAs—the database architect (DBA-2)—because we still need people to use the newer database technologies and features to shorten deployment cycles and enhance service levels. That kind of creative DBA work is not likely to go away in the foreseeable future. Also, it will be a long time before the need for development DBAs and to a lesser extent, application DBAs, goes away.

What are some of your pet peeves about Oracle products? What new features would you like to see in Oracle?

Let me start out with a positive comment. In many ways, Oracle still has a much better product than many others out there. For instance, I haven't seen any other database provide the kind of granular performance metrics that Oracle's instrumentation layer provides—it pervades the database and goes deep into the network, hardware, storage sub-system, and application layers. It makes advanced troubleshooting so much easier.

My biggest peeve about it is that Oracle products tend to be much more bulky (in terms of resource consumption) and complex than required.

Database administration jobs are being sent to India (or Romania or the Philippines) all the time. Is the trend increasing or moderating and even declining?

In the short term, I expect the trend to accelerate more until a good chunk of DBA-1 work is offshored. Corresponding short-term challenges such as security concerns and false patriotism will evaporate as DBA outsourcing further matures. But as salaries and employee turnover rates continue to go up in India, the window of opportunity will become larger for other places like China, Romania, and the Philippines. If they play their cards right and sufficiently polish up their English language skills, these places will be in a good position to replace India as the offshoring destination of choice. However, real business issues such as lack of standard operating procedures resulting in inconsistent quality of work, problems with communications and cultural differences, different work ethics and customer expectations, and eventually rising salaries in China, Romania, etc., will lower the offshoring curve again. Autonomics will begin to bridge the gap at lower costs and better quality.

Meanwhile, human DBAs everywhere will continue to innovate and climb upstream, they do not have a choice. Corporations will always eventually do what's in the best interest of the shareholders. We need to embrace technology

You should be ensuring that each task worked on by a DBA has a corresponding ticket—no exceptions!

to continue adding higher value. As long as we do that, we will always have a job.

What is your experience with the quality of offshore personnel?

Overall, offshore personnel in the DBA area tend to be not as experienced as the quality of talent we find in the U.S. and require more training. But that is understandable because countries like India have been catapulted from an environment that had very little IT to the forefront of being the IT custodians for some of the world's largest organizations. Companies that have successfully utilized offshore resources have invested significantly in training, building standards, and improving communications. The salary difference is still enormous and the time differences make it ideal to leverage offshore locations for follow-the-sun support models. However, the thing that really hurts our company is employee turnover in India. Sometimes this starts even before people start on the job—they accept an offer, request a start date that is a month away (in India, people often are required to give a month's notice) and then, they don't show up on the day they are supposed to—no calls or explanations come forth!

India especially is going through a boom that we went through in the late '90s where a DBA could walk across the street and get a 30–40% raise. Once overseas HR policies strengthen and the demand settles and salaries level out, I expect this will be less of a problem in the next 3–4 years. Either that, or increasing levels of individual greed will cause India to lose its advantage to other countries, like China and Romania, that are further behind as far as IT salaries go.

What career advice do you have for database administrators and wannabe database administrators in this country? (Should we all switch to bee-keeping and other jobs that cannot be outsourced to cheap offshore locations?)

Not at all. On the contrary, I would say there is a shortage of good DBAs in the U.S. One of the reasons we use offshore resources is because we can't find enough good people here. More people should get into DBA work with both guns firing! Most importantly though, they need to be innovative and business-savvy. The DBAs here often have the luxury of being on-site with the business users. Spend time with them to understand how you can use your knowledge of the database to better serve their current and future needs.

Focus more on being database architects rather than database administrators. Be willing to embrace newer technology and automation. Be the technology champion in your organization by segregating workload into mundane and non-mundane. Don't be worried about losing your job if the mundane work gets outsourced to firms that specialize in that and can bring economies of scale to bear. Instead, figure out how not to drop the ball with the non-mundane work. Read and understand the new features that your DBMS product gives you; play with those features in a lab or development environment, and work on projects that utilize those features to enhance your environment.

Go beyond the command line and the GUI to understand the concept behind the technology. Don't worry that if you don't remember the commands, they will become antiquated. On the contrary, with your institutional knowledge, proximity to your users, deeper product knowledge, willingness to embrace outsourcing and automation, and your newly nurtured architectural capabilities, you can add more value to your company than ever before.

There is a shortage of good DBAs in the U.S.

Finally, what kind of advice would you have for managers in our field?

Beyond asking those puzzling interview questions to screen your new hires, you should be doing your bit to make day-to-day DBA work a science in your environment. You should be ensuring that each task worked on by a DBA has a corresponding ticket—no exceptions! You should be analyzing the tickets on a weekly basis. The tickets should describe the exact problem worked on (e.g., it should read "rebuilt index NDX_tblname_02" rather than "did maintenance on the DB") and the amount of time the person worked on it. If any ticket pertains to a recurring problem, you should expect your seniormost DBA to build a standard operating procedure for it that everyone can follow to produce consistent results when dealing with that issue. That procedure will also be the blueprint for any downstream automation.

You should attempt to ensure that once such standard operating procedures and automation routines are built, they are being used by every DBA in the team. You should attempt to ensure that workload is distributed evenly and average resolution time is brought down. When you do quarterly comparisons, you should see a reduction in time for tasks that took up the bulk of DBA time the previous quarter. Periodically, you should have a trustworthy consultant come in to do a "discovery" to ensure you are not missing any opportunities for further optimization. If you run a shop based on these ground rules, you can rest assured that your team members will have high morale, you can get a lot more done, and your senior management will value you—without having to worry about the optimal database-to-DBA ratio. Head count is less relevant; process is the king! ▲

Interview conducted by Lisa Loper and Iggy Fernandez.