FEATURE

The Making of Oracle Database 10*g*

By Barbara Assadi

A behind-the-scenes look at how the world's most sophisticated database was built

In the software development world, designing, building, and releasing a new version of the Oracle database is about as interesting as it gets. What's behind this massive effort? We took a look at the industry's brightest developers, managers, and marketers; Oracle's cutting-edge technology; and the detailed planning process that brought it all together.

Early Direction

Planning a release as large as an Oracle database is no small task. For Oracle Database 10*g*, the planning phase commenced about two years before coding began, when Oracle developers and managers gathered to brainstorm changes to the database. "We're able to be so creative in this stage because we have the industry's most stable database kernel," explains George Demarest, senior marketing director for Oracle Database. "This gives our developers a strong base to work from. The system doesn't unravel when changes are made or features are added."

Not surprisingly, the biggest challenge isn't coming up with great ideas to build into the database—it's narrowing them down. "Within any given week, any Oracle development group is capable of coming up with enough great ideas to keep the division busy for years," says Carol Sexton, Oracle's database release manager. "The tricky part is picking the right projects."

The team agreed that database manageability was one key area that needed improvement. Not only had managing the database become increasingly complex, but the evolution of outsourced computing and new technology paradigms such as enterprise grid computing demanded a more manageable database. "Within a decade, enterprise IT will have to decide whether to outsource its IT or continue to manage computing power in-house," explains Demarest. "Either way, software that's completely self-managing and self-healing is critical, both for the enterprise that continues with hands-on IT management, and for providers like Oracle who manage IT services for their customers."

Another major product initiative came directly from Oracle Chairman and CEO Larry Ellison, who thought customers relied too heavily on third-party software to run their Oracle database. "One of the goals of Oracle Database 10*g* is to deliver a complete, integrated stack of software," explains Andy Mendelsohn, senior vice president in Oracle's Database and Application Server Technologies group. "This lowers customer costs, improves system reliability, and gives customers a much better out-of-the-box experience."

Based on these directions, Oracle's development managers identified individual features that would ensure the new database met these goals and divided them into 12 focus areas, each with its own leader. Explains Prem Kumar, vice president of Oracle Server Technologies: "I was the focus area leader for Windows, someone else took responsibility for business intelligence, and someone else handled clusters. Each leader identified projects that we needed to work on to meet our customers' performance and manageability needs, and projects we needed to tackle to stay ahead of the curve. Once a set of projects was approved by management, that set the ball in motion." Approved development projects were entered into the Project Database.

Documents and More Documents

Writing the code that comprises the Oracle database is a disciplined activity that begins with the creation of a Functional Specification Document—a document that provides details about each new feature of the database, complete with architectural diagrams—that was distributed to all Oracle Database 10*g* development teams to review and approve. "The Oracle database code is large and complex, so there's a lot of interaction among various groups," says Kumar. "The Functional Specification Document is taken very seriously."

After the Functional Specification Document was finalized, developers drafted a Design Document, containing the actual algorithms used to write code. A Test Specification followed, which detailed how the database would be tested.

Next, individual product owners entered information into the Project Database about their projects, such as the number of developers required for the project, length of time to complete, and so forth. "We also entered information about the exit criteria," notes Sexton. "So we knew up front how we would measure and track the quality of the release."

The emphasis on planning significantly shortened the Oracle Database 10*g* development cycle and helped to condense the amount of time developers spent writing code. "If you look at the entire release cycle, the actual time developers spent writing code is about 25 percent of the overall project," says Kumar. "That may sound surprising, but it makes sense, since the rest of the time you're either planning for the release, writing specs, or testing."

All for One and One for All

Imagine telling 800-plus Oracle developers that they'll have to crank out a major release of the Oracle database—without their workstations. That's exactly what the company's senior management did at the beginning of the Oracle Database 10*g* project.

"In the past, developers used local, four-CPU machines for development and testing," says Kumar. "But the number of tests has grown from 30,000 to more than 100,000, which is more than a desktop machine can handle." So, Oracle pooled the developers' individual computing power to create a server farm of more than 1,000 machines. This development grid represented a major breakthrough for Oracle, giving developers on-demand computing from any location. It also offered them far more horsepower than a single dedicated machine could provide. As a result, they could edit source code not only more safely and accurately but also more quickly.

Kumar explains how the development grid increased productivity: "If I change my code, I might have to run it through 250 hours of testing before I can check it into the source control system," he says. "In the old days, I'd have to tie up my workstation. Now, I just send the test request to the development grid, which takes that 250 hours of testing and breaks it down into 100-plus streams. If 100 machines are available in the server farm, the tests will run simultaneously on those 100 machines. So, instead of waiting for 250 hours, I could potentially get the test results back in a couple of hours. The server farm is a cost-effective, shared entity that is being fully utilized. It's not lying idle in someone's office or cube."

With the server farm, the Oracle Database 10*g* developers became early users of grid computing, one of the product's focus areas. "Oracle is among the first users of this grid concept, especially for software development and testing," comments Kumar. "And we benefited tremendously from the development grid."

Building and Testing Software Every Day

Once coding began, developers checked in code daily. Making sure code works is no trivial process: A set of 100,000 regression tests ran nightly against the daily build, using the 1,000-machine development grid.

"We did a daily sanity test—the short regress—to make sure that whatever we compiled and built was OK," describes Kumar. "After that, we kicked off a series of tests called the long regress. It takes about 3,000 CPU hours, and we did that every single night."

When developers returned to the office the next morning, they reviewed the long-regress test results from the previous night. The results provided a snapshot of the product, showing where successes and differences were found. A success meant the code was behaving normally; a difference (or "diff") meant something was broken. "Once diffs were identified, they were automatically assigned to a developer for resolution," says Kumar. "We tracked this work closely throughout the course of the release."

The daily build is protected in a way by the source control system running on Oracle Real Application Clusters (RAC), and to date it has never failed. The combination of the server farm, with its massive testing resources, and the RAC-enabled source control system has made Oracle developers productive and has greatly reduced Oracle's development and related IT costs.

Keeping track of daily code testing was the job of the Release Management Group, led by Sexton. "We had checkpoint meetings every six weeks to make sure the development managers were on top of the project and on schedule," she says. "Things don't always go as planned, and these meetings give us a chance to anticipate problems and respond quickly to events."

Sexton says that closely managing the release kept it from spinning out of control. "Many developers in the industry have experienced teams just throwing a whole bunch of code into the mix, and then spend months just trying to get it to work. At Oracle, the principle is simple: We want the code to be stable every day from day one. The only way to do that is with a daily build, and by making sure that your regression tests look OK."

Release management was also tracked in an Oracle database. "This database stored project information including their status and their test results and bug counts; the latter was gathered directly from testing and updated daily," says Sexton. "We could see if any groups in the last build had a lot of regressions or if bugs were getting out of control. We examined bug counts every week and expected development teams to keep bugs to a manageable number so there wasn't too much lag time."

Beta Testers Get an Early Start

Beta testing usually doesn't begin until alpha testing is done and a relatively stable version of an application can be distributed to beta testers. But to get beta testing started earlier, Oracle created a hosted environment for beta testing. "Hosted beta testing for Oracle Database 10*g* let us start testing early in the process," says Debbie Migliore, director of Oracle beta programs. "In the hosted environment, we gave each customer remote access to an exclusive instance of the database for testing." Test platforms included Solaris, HP-UX, Linux, and Windows 2000.

In addition to the hosted beta testing, Oracle invited individuals for a week of test activities at Oracle headquarters. "Developers sat down right next to customers and had them run tests on the release," says Migliore. "We'd have 10 or 12 customers in the room with 20 developers. This gave us incredible feedback from a product perspective. What was great for developers was that not only did they get a chance to understand how their products were being used in the real world, but if problems were found, the developers often fixed and tested them right there."

The interaction illustrated the different ways developers and customers test a system. "What is so valuable about the beta program is that customers don't test Oracle Database 10*g* in a vacuum," says Migliore. "Rather, they test the new release against their existing applications in a diverse environment using a large set of existing functionality. We found that the first thing our beta sites test is often the last thing we test. They test data loading, migration from older releases, and compatibility with other applications, while we initially focus on making sure our new features are good to go. They often identify things we miss because we haven't gotten that far in our full-scale, high-performance testing."

Migliore's group also works with third-party developers. "Once the database release is close to production quality, we will release it to thousands of ISVs for testing. Our goal is to ensure a significant selection of third-party products in the marketplace supporting Oracle Database 10*g*," she says.

Spreading the Word

The best database release in the world isn't worth much if no one knows about it. Getting the word out was the job of Oracle's marketing staff. "Marketing doesn't typically get too deep into the product design process," says Oracle's Demarest. "The feedback we provide is more on the macro level. While individual features may not create a crucial purchasing decision for customers, a group of features that cuts downtime in half is a different matter. We look for the stories that will really affect DBAs, developers, IT managers, and even CEOs.

"Early on, we developed a core document that distilled the myriad features of the release into a statement of the release's key messages," adds Demarest. "We spent a lot of time on the language. How do you reduce a complicated, far-reaching release into a relatively small number of messages?"

What are the top messages for Oracle Database 10*g* according to Demarest? "First, it's about making things easier, or even better: automatic. This is one of the key ideas around the emerging phenomenon of enterprise grid computing, or IT as a utility. Oracle Database 10*g* is the first database designed for grid computing. In grid computing, it's essential that infrastructure software such as databases and application servers are self-managing and self-healing. As a result, this release was designed to reduce the database's management time by half, or more. It's the easiest database out there to manage. And that's the second message: Oracle Database 10*g* is a self-managing database.

"The third message is that our current customers will get immediate benefits from updating from earlier Oracle Database releases. By simply installing Oracle Database 10*g* they'll immediately experience better performance, easier management, better reliability, and a raft of new capabilities."

***Barbara Assadi** is a cofounder and principal of BayCreative, a San Francisco-based marketing agency, and has authored several books on computer software.*

Source: http://otn.oracle.com/oramag/oracle/03-sep/o53making10g.html