SELECT Journal

For the Complete Technology & Database Professional

Q4-16

Agile Upgrades

Modern and progressive updates keep pace with change.

Features:

Migration to 12*c* Made Easy Using Replication Technology

DBA 201: Mitigating Risk in Database Upgrades

Using RAT Features in Oracle Database 12*c* to Replay Previously Captured Workloads

IOUG independent oracle users group

www.ioug.org

SELECT Contents

Volume 23 ■ No. 4 ■ Fourth Quarter 2016

Features

Columns

IOUG Press Corner

How can I contribute to *SELECT Journal*?

Write us a letter. Submit an article. Report on Oracle use in all corners of the globe. We prefer articles that conform to AP guidelines. Send to select@ioug.org.

Headquarters

Independent Oracle Users Group
330 N. Wabash Ave.
Chicago, IL 60611-7621
USA
Phone: +1.312.245.1579
Fax: +1.312.527.6785
Email: ioug@ioug.org

Editorial

Alexa Schlosser
Managing Editor
IOUG Headquarters
Phone: +1.312.673.5791
Fax: +1.312.245.1094
Email: select@ioug.org
editor@ioug.org

How do I get the next one?

SELECT Journal is a benefit to members of the Independent Oracle Users Group. For more information, contact IOUG Headquarters at +1.312.245.1579

SELECT Journal Online

For the latest updates and addendums or to download the articles in this and past issues of *SELECT Journal*, visit www.selectjournal.org.

Users Group Calendar

For the most updated calendar, please visit www.ioug.org/events.

Click calendar icons for more information.

December 2016

2 — **The Oracle OUGPY Day Paraguay 2016**
Hotel Excelsior Asuncion, Paraguay
Chile esquina Manduvira
Asunción, Central 1209
Paraguay

7 — **NYOUG Winter 2016 General Meeting**
Borough of Manhattan Community College
199 Chambers St.
New York, NY 10007

8 — **WEBINAR: Using DevOps for an Oracle Data Warehouse**
Thursday, December 8, 2016

April 2017

2-6 — **COLLABORATE17**
April 2-6

Advertiser's Index

The Indepedent Oracle Users Group and *SELECT Journal* would like to thank the following members of the community for their support. For more information about their products and services, visit their websites. And be sure to let them know you saw them in *SELECT Journal*!

Get Oracle BI Support When You Need It Most

24x7 Business Intelligence Solutions

Datavail's unique managed services model now includes Oracle Business Intelligence and analytics services via our more than 600 DBAs, analysts, and developers.

For companies running applications across the Oracle stack including Hyperion, OBIEE, OBIA, GoldenGate, WebLogic, and more, our expert support in project consulting, database administration, data warehousing, and enterprise performance management is available to you 24x7.

- Oracle Database Upgrades
- Report & Dashboard Building
- Application Configuration
- GoldenGate
- Performance Tuning
- OBIEE
- Hyperion

datAvail

BI/Analytics • Applications • Databases

www.datavail.com | 877.634.9222

From the Editor

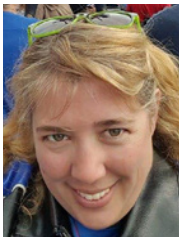
Farewell... And Welcome

By Gary Gordhamer, Executive Editor, IOUG *SELECT Journal*

Hello everyone, and welcome to the Q4 2016 issue of *SELECT Journal*. My term as executive editor for the Journal has come to an end. Let me start by saying thank you to all the great authors, editors, staff and volunteers who have worked with me over the past six years. Without lots of help from many, many people, I never would have risen to this role. As Stan Lee always reminded us, "Nuff said".

As with most volunteer work, I won't be far away. I'm serving as the COLLABORATE 17 IOUG Conference Chair. The team is hard at work right now to bring over 200 great technical sessions to Las Vegas in April 2017. The herculean effort of bringing together DBAs, developers, and architects for five days started over three months ago and will continue well into April. See you in Vegas!

I feel confident passing the SELECT Journal mantle to our new Executive Editor Michelle Malcher, who is no stranger to the IOUG community. She will help usher-in the next generation of *SELECT Journal* as we move to a full web-based experience in 2017. Take it away, Michelle!



From the Editor

Upgrades & Opportunities with the Arrival of 12*c*R2

By Michelle Malcher, Executive Editor, IOUG *SELECT Journal*

For all those waiting for Oracle database 12*c* Release 2, it is time! 12*c*R2 has been showcased and is already available in the public cloud, which means it is time to start planning those upgrades. This release brings new features and provides the stability over the R1 version. The structural changes to 12*c*, with container and pluggable databases, can cause reasons for concern about the upgrades. The user community has been involved in beta testing and still more are early adopters of 12*c*R1. This is a great opportunity to learn from the community.

Successful upgrades and using the new technology features are definitely in store for those in the community.

It is also not just the database that continues to have new releases, but the tools we use for managing them. Oracle Cloud Control updates include features to provision databases and manage the large hybrid and cloud environments that we are dealing with. Big Data, clustered databases and managing tools all have enhancements and new features – new features that are becoming must haves for the applications and data services to perform well, be secure and highly available.

I always view upgrades as an opportunity for new features to benefit the system and provide a solution for a current issue or project needs. These benefits do help justify the required testing, maintenance windows and effort for the upgrades. However, just as there have been enhancements to providing always-available databases, there are tools and steps to minimize the database downtime and maintenance windows for upgrading.

As with any database upgrade, standards and baselines should be reviewed to validate that proper security and feature changes for the database to meet your company's minimum requirements. For example, the non-container database is being deprecated so the standard build would now be a container database with one pluggable.

This quarter, *SELECT Journal* brings you other upgrade processes and testing plans. Gary Gordhamer provides a DBA 201 article on risk and testing database upgrades. Jim Czuprynski continues his series on Real Application Testing features. Donna Zehl informs us on how to use replication to minimize the downtime during migration to database 12*c*. These articles along with others in this *Journal* should provide steps and valuable information to help prepare for the new releases. In upcoming *SELECT Journal* issues, we will continue to examine the new 12*c*R2 database features and where the value is being added.

Successful upgrades and using the new technology features are definitely in store for those in the community. As always, reach out to us with your questions, comments, or let us know about your favorite new feature in Oracle database 12*c* at select@ioug.org.

10 MINUTES

That’s how long it takes to deploy a fully configured, highly-available Oracle Database on the AWS Cloud

Quick Provisioning. Easy Management. Reduced Cost.

Amazon Relational Database Service for Oracle

Try it today for free - aws.amazon.com/rds/oracle/try

amazon web services

Upgrading Your Big Data Clusters Can Be Challenging

By Ian Abramson

The promise of Big Data is to provide data in a way which is unencumbered and available when needed. The key to providing a reliable and secure environment is to ensure that your software is current and up to date. Depending on how you installed your Hadoop cluster and the distribution you are using, this can either be easy or very difficult. Most distributions like MapR, Couldera and Hortonworks support a guided and supported upgrade path. If you are using Apache Hadoop, then you will find the challenge to be significant.

As an experienced Oracle professional, I realize that database upgrades have improved over the years to a point where they are almost simple to perform. The task is well documented and the problems you encounter tend to be ones with which others have seen and to which Oracle would either provide a workaround or a patch. With Hadoop, this is not always an option, and, often, you need to develop fixes on your own or wait for the community to develop things.

From a core computing perspective, the upgrade can be done with a reasonable amount of confidence. You can install a new version of the software and then copy or re-assign your cluster to the new version of Hadoop. If you are using a cloud implementation of Hadoop, this can be done quite effectively, with two separate clusters. If you are upgrading in-place, then I suggest you back things up, if possible. But just like upgrading your database, you must be careful. With the right planning and testing, things from a foundational level should be fine.

The biggest challenges and problems I have seen is the secondary products which you install along with Hadoop (HDFS). Upgrades to products like Spark have caused issues with backward compatibility. Due to changes in libraries, we have found that we must retest all of our code to ensure that a new software version supports our current code. You will find that in the Big Data ecosystem the pace of change often introduces problems that previously did not exist or a function you previously used is no longer supported or has changed. This is where you need to focus your upgrade energy. We are seeing significant change in areas of processing, security and presentation, and if you want to take advantage of this new functionality, you may also know that you may need to rework your effort. As more stability comes to Hadoop, you can expect that these problems will be reduced and that better backward compatibility is maintained.

Ultimately, you must keep your Hadoop cluster current; just as you would with your relational databases, you need to consider keeping your cluster current through a carefully planned and executed upgrade plan. S

Ian Abramson is an Oracle and data warehousing expert with more than 20 years of hands-on experience. Based in Toronto, Canada, he is a Data Architect for Walmart Canada.

Technologies that Enable Change

By Seth Miller

The IOUG membership is a diverse group holding many different titles. Regardless of whether you are a developer, manager, C-level or administrator, your organization's databases are an integral part of or your company, which is why *SELECT* devotes so much content to the Oracle Database. My conversations with professionals about Oracle Database 12*c* are always interesting — ranging from, "It's the most stable release in a decade," to "I'm not touching it until release two," to "I'd rather go back to spreadsheets than install that."

Oracle Database 12*c* Release 1 became generally available June 2013. Despite having been an early adopter of 12*c*, even I do a double take when I realize it's been out for almost three-and-a half-years already. It's no mystery that customer adoption of Oracle Database 12*c* has been slow, and it's also no secret that part of that hesitation has to do with the negative experiences many had with the first release of 11*g*. While the first release of 12*c* is notably better than 11*g*R1, that stigma will only fade with time and experience.

Oracle is 100 percent invested in cloud architectures and is encouraging customers to move into public, private and hybrid cloud computing.

If you have been following the announcements from Oracle OpenWorld this year, you have seen that Oracle announced the release of Oracle Database 12*c* Release 2 – well, kind of. For the first time in Oracle's history, they are releasing a version of their flagship product in their public cloud first. According to the announcements and a number of conversations between myself and Oracle Product Managers, there is no official release date for the full installable version of 12*c*R2. It is clear, that Oracle is 100 percent invested in cloud architectures and is encouraging customers to move into public, private and hybrid cloud computing.

In hopes of fast migrations to the newest release, Oracle sometimes gets a little ahead of itself. According to Oracle Support documents prior to October 2015, premier support for 11*g*R2 would end in January 2015, just more than five years since its release and extended support would end January 2018. Given that 12*c*R1 was released more than three years after 11*g*R2, far longer than any of the previous five releases, the extended support fee was waived until February 2016. The adoption of 12*c* was likely much slower than

For the first time in Oracle's history, they are releasing a version of their flagship product in their public cloud first.

Oracle had anticipated — so much so that the support for 11*g*R2 was revised. In October 2015, Oracle announced that not only would the fee for extended support for 11*g*R2 be waived until June 2017, but extended support would now be offered for an additional two years, ending December 2020. This extra time and waived fee should ease customers' transition into 12*c* but don't sit back for too long, four years isn't that far away.

By far the largest architecture change with 12*c* is the multitenant option, also referred to as container database (CDB). With the release of 12.1.0.2, Oracle revealed that the non-CDB (non-multitenant) architecture is deprecated.

This language in the documentation was recently updated to reflect that non-CDB is not only deprecated but will no longer be available after 12*c* Release 2. It says, "The non-CDB architecture is deprecated in Oracle Database 12*c*, and may be de-supported and unavailable in a release after Oracle Database 12*c* Release 2. Oracle recommends use of the CDB architecture."

The move to a multitenant is one of the most difficult transitions for administrators that have been using the same database architecture for decades. A clear example of this is not being able to log into a pluggable database using operating system authentication (/ as sysdba). A pluggable database can only be accessed by connecting to the container database and running an alter session command or by connecting directly to the PDB over OracleNet. A change this big is nothing to fear. It becomes second nature so quickly that you won't even realize you are doing it. Once you are connected, you might be surprised at how similar everything looks.

While change is inevitable and seems to be increasing in velocity and scope, it is still difficult for companies and their administrators to transition between software releases, especially when those releases flip the traditional architecture on its head. IOUG is committed to helping its membership overcome these obstacles through content like the articles in *SELECT Journal*, presentations at conferences and local events, webinars held regularly throughout the year, subject

(Continued on page 34)

Revamping the Water Deionization Industry with the Oracle IoT Cloud

By Amit Mehetre, Ashish Thakkar and Ankit Thakker, L&T Infotech ■ Eric Mader, Editor

We live in an increasingly connected world, where the physical and digital spaces are converging.

Technology puts this convergence to work, in the form of the Internet of Things (IoT), which leverages automation, analytics and exchange of data and transform processes, increasing efficiency and creating value like never before.

We live in an increasingly connected world, where the physical and digital spaces are converging. Technology puts this convergence to work, in the form of the Internet of Things (IoT), which leverages automation, analytics and exchange of data and transform processes, increasing efficiency and creating value like never before.

Using the power of IoT, we at L&T Infotech have developed a smart solution for the water deionization industry using the Oracle IoT Cloud. Water deionization is the process of removing dissolved impurities such as salts from water using deionization cylinders. Keeping track of the health of these cylinders is one of the critical components of the entire deionization process.

Our solution connects collected industrial data with customers, technicians and service providers to provide predictive and preventive maintenance, thereby increasing the efficiency of equipment and the overall deionization process.

Service Provider: The organization that provides cylinders for water deionization.

Technician: Field service engineer; the employee of a service provider.

Customer: Organization which avails service from service provider.

Traditional Deionization Process

Traditionally, water deionization uses Electrical Conductivity (EC), temperature and flow sensors to monitor the process on premise. These sensors gather data as water flows out of a treatment facility.

The EC sensor measures the conductivity of the water, which is directly proportional to amount of impurity in water and inversely proportional to the health of cylinder. Signals from these sensors are collected by a panel, which stores and transfers readings to a USB storage device. The panel throws an alarm when the conductivity value crosses a certain threshold — indicating that the cylinder is exhausted.

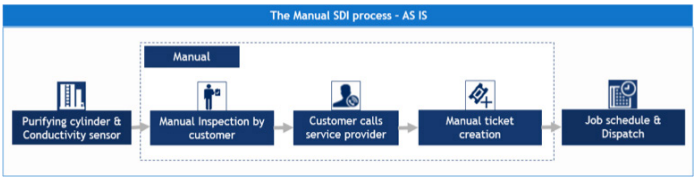


Figure 1: Manual Deionization Process

The sensor readings are manually checked once every 15 days, and a technician, based on the data, estimates the health and remaining life of the cylinder. After a floor supervisor determines that a cylinder is exhausted, the cylinder is replaced with the spare one and the replacement of the exhausted cylinder is scheduled with a service provider. Later, the service provider replaces and collects the exhausted cylinder and sends the exhausted one for regeneration process, in preparation for reuse by a different customer.

The manual interventions in the traditional deionization process are prone to errors. For instance, impure water could result from the process if the exhaustion of the cylinder is not noticed immediately. Also, there is no way to know precisely when the cylinder will be

exhausted, thus 33 percent extra inventory is maintained at a facility, at all times.

It is also typical for an end customer to send cylinders to a competing vendor for cheaper regeneration or for some other reason. Due to this, there is a critical need for service providers to track whether or not the cylinder is on a customer's premises. The tracking helps preserve the technology and ensure continuous business.

Smart Water Deionization Solution – Technical Insights

L&T Infotech's Smart Water Deionization Solution uses Oracle's powerful IoT cloud platform, which gives simple and secured device management, rule engine and real time data exploration, cloud storage and seamless integration with various enterprise applications.

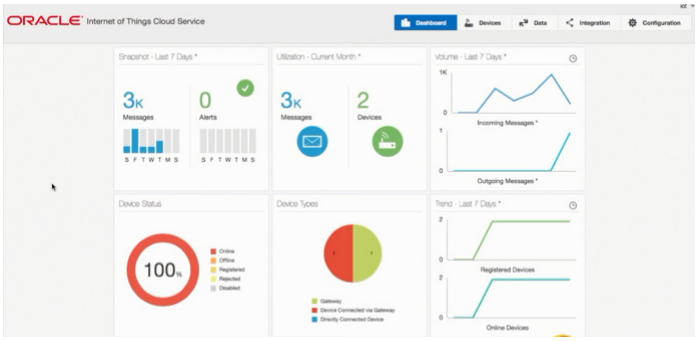


Figure 2: Oracle IoT Cloud Service Console

The solution also uses other products from the Oracle stack including Oracle Service Bus, Mobile Application Framework and integrations with Oracle JD Edwards Enterprise One.

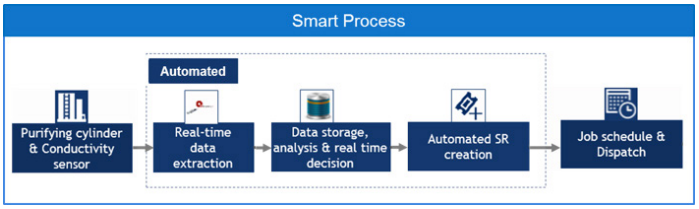


Figure 3: Smart Water Deionization Process

Figure 3 shows the flow of smart water deionization solution. Instead of relying on an on-premise panel, signals from the sensors are collected by a gateway device — a Raspberry Pi — via conditioning circuits. The gateway device collects signals from different sensors from multiple deionization setups on-site. In this solution, temperature, electrical conductivity of water and cylinder tracking data is captured using sensors with custom ZigBee based tracking tags. Since the electrical conductivity sensor is analog in nature, an Arduino Uno is used to convert data into a digital message format. The digital temperature sensor and tracking data is also attached to the Arduino. The Arduino converts all readings into the required digital message format and sends it to the gateway device using a serial protocol.

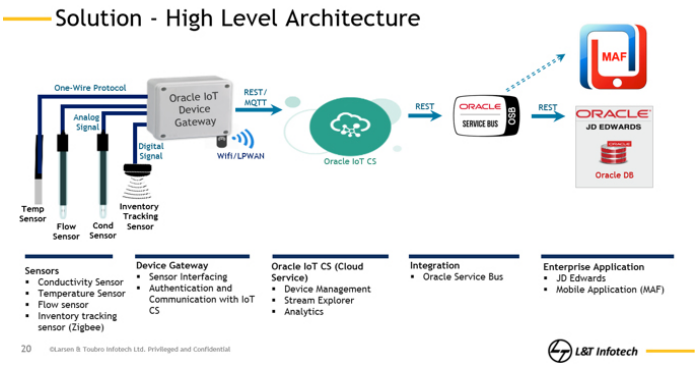


Figure 4: High-level solution architecture

The gateway device runs the Oracle IoT Gateway software, which establishes a secured communication with Oracle IoT Cloud service over the Internet. The solution connects to the internet over Wi-Fi, but can alternatively use other communication channels like GSM/ GPRS, 3G and Sigfox. The Arduino, which acts as an end node, is managed as a device connected via a gateway configuration in the Oracle IoT Cloud service.

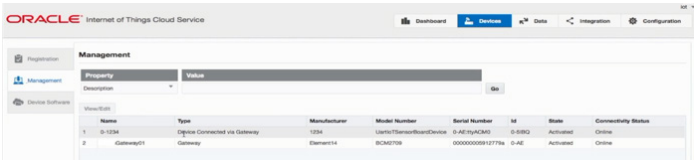


Figure 5: Device management – Oracle IoT CS

Once the device is registered and the data starts reaching the IoT CS, a data stream can be created using the out-of-the-box powerful rule engine Stream Explorer (SX). In SX, explorations can be created to analyze the data stream based on rules. Rules can be defined in SX, and are based on which data is analyzed in real time.

The smart water deionization solution consists of three explorations.

1. **All data exploration:** This exploration captures all the data emitted by devices and received by IoT CS, and analyzes live readings of temperature and electrical conductivity. This data is later stored in an on-premise database via Oracle Service Bus. This data is utilized by the mobile application to show the real-time monitoring.
2. **SOR exploration:** The rules within this exploration look for the temperature or electrical conductivity to cross a specific predefined threshold. For instance, if electrical conductivity crosses a threshold of 8 uS/cm, the rule in this exploration is trigged and the data is captured. This exploration changes a flag for service order request in local database via Oracle Service Bus, which will raise a service order request in JDE.
3. **Tracking Exploration:** This exploration tracks whether or not the cylinder is in the premises. Whenever a cylinder is moved, the exploration changes a flag in Oracle Service Bus, which raises an alert.

These explorations are also visible in IoT CS console, under the Analyzed Data screen. The Integration tab within the IoT CS web console provides features to integrate with different enterprise applications. For instance, all the above explorations are integrated with Oracle Service Bus.

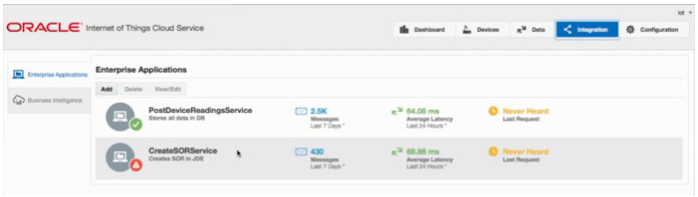


Figure 6: Integration console: Oracle IoT CS

In this solution, data is routed from the IoT CS to a mobile application and JD Edwards Enterprise One via Oracle Service Bus. The mobile application allows the end customer to monitor the performance of the deionization system from anywhere in the world. The customer also receives an alert if the health of the cylinders degrades, based on the data coming through sensors and the rules configured in the IoT CS. After receiving these alerts, the customer can immediately schedule a replacement of the cylinder from the mobile application. The request is generated in JD Edwards and a service provider can take the required action. This allows the service provider to receive a higher visibility over demand and supply in advance.

Likewise, if a customer fails to notice the health alert and the cylinder goes into a critical condition, the system automatically generates a request for the replacement. In this case, the service provider contacts the customer to schedule a visit for the replacement work, which helps put the customer's mind at ease and improve the overall service experience.

In this way, the system provides both preventive and predictive maintenance capabilities, and solves the challenge of maintaining spare inventory by an overall reduction of around 33 percent. The solution also provides cylinder-tracking capabilities for the service provider. Cylinders are equipped with ZigBee tracking tags, which generate tracking data that is also leveraged in the IoT CS. In the event that a cylinder is moved off of a customer premises without prior approvals, an alert is sent to the service provider and necessary actions can be taken.

The Field Service – Mobile Application

The Smart Water Deionization application offers a mobile platform for service providers, customers and service engineers. The application fetches data from on premise database as well as JDE using Oracle Service Bus services.

The application for customers features a dashboard view. This dashboard contains critical data provided by the Oracle IoT CS, including the ability to view and monitor all installations across multiple locations, and track real-time operational parameters. In addition, customers are also provided with alerts and notifications, critical business information, analytical reports, and the ability to easily request service from anywhere.

Figure 7: Customer dashboard - mobile application

The top left section of the dashboard shows the live readings of conductivity and temperature to the customer, providing live monitoring capability of a remote plant. The dashboard also shows the trend of electrical conductivity, which is inversely proportional to the health of cylinder. Thus, on the dashboard itself, a customer can visualize how the plant has been performing over time.

In the top right corner of the dashboard, the customer is provided with a real-time prediction of the next service order. Based on current electrical conductivity data, the algorithm predicts the number of days left before the next replacement of the cylinder will be required. The dashboard also fetches important business-related information from JDE, such as the invoice amount due, the contract expiration date and SOR history.

Service providers also have a similar dashboard view, which allows for the tracking of installation jobs and both connected and disconnected cylinders based upon customer type and geography. Similar to the customer application, service providers can also receive real-time alerts and notifications for pre-defined conditions, such as the unauthorized movement of a cylinder outside of a predefined area.

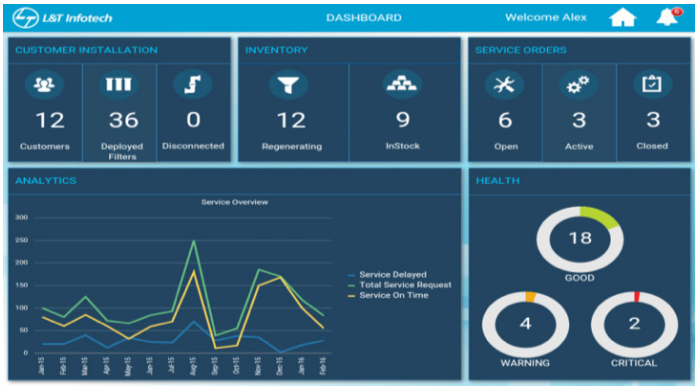


Figure 8: Service provider dashboard - Mobile application

Service providers can also track key metrics about active cylinders, such as conductivity, and view trends and perform

analysis of this data. The ERP data can also be pulled into the mobile application, for situations such as real time asset tracking and service order management.

To provide prompt service to customer, it is essential to efficiently and intelligently manage human resources. The field service engineer's mobile application does this job smoothly.

Figure 9: Open Jobs – mobile application

When a work order is created and assigned in JDE, the assignment immediately appears in the mobile application of the field service engineer. The engineer can accept and schedule the work order right from the mobile application. The application also keeps him up-to-date with upcoming jobs that he has been assigned. Upon completion of the job, the application can also capture the customer's signature and feedback on the service.

Additionally, the mobile application provides an "Ask the Expert" feature that allows the engineer to take a video call with the expert from field in order to showcase the fault and solve it from anywhere. Engineers are also provided with a gamification feature that tracks skill levels, time of completion of jobs, and maintains rankings for the jobs completed by the engineer.

JD Edwards Enterprise One

The solution utilizes JD Edwards 9.1 with Tools Release 9.1.5.2 as the ERP platform. Modules like Manufacturing, Inventory Management, Sales Order Management and General Accounting (G/L) were implemented. For the Water Deionization Solution, the Inventory Management and Manufacturing modules are linked with the overall Oracle IoT CS architecture to achieve the digitization of the water deionization process.

During the architecture phase of the project, two options were considered for integrating with JDE. The first option was to utilize the IoT Orchestrator using Application Interface Services (AIS), which was an alternative to the implemented solution that involves OSB connecting to the JDE Business Services (BSSV). In this case, since the client already had Oracle Service Bus (OSB) as a part of their existing technology stack, it was decided to leverage this platform for the JDE integration.

JDE provides web services for querying the inventory from the database and to create the service orders. When the conductivity or temperature crosses a threshold, the data is routed via OSB to an on premise database. To achieve this integration, the OSB server running on the database calls the JDE BSSV via a web service call. The new service orders are then processed further from within JDE.

The BSSV query retrieves the complete details of a customer such as contract information, invoice details, last active or new service order details, and parts information for specific equipment. This information is displayed in mobile application via web services.

For example, consider the details for a specific contract, which are fetched from table F1721 based on customer number (AN8) and equipment number (NUMB). The order details are fetched from table F4801 based on Equipment Number (NUMB) and SRST as SX, SV for last added records and active orders for that equipment. For each order fetched from F4801, the invoice details are filled from table F4812H. Parts details are fetched from table F3111 based on the equipment number. The standard BSSV - JP170000 CustomerServiceManager web services call then triggers the creation of a service order. The processServiceOrder method of BSSV is then used for creating the actual service order.

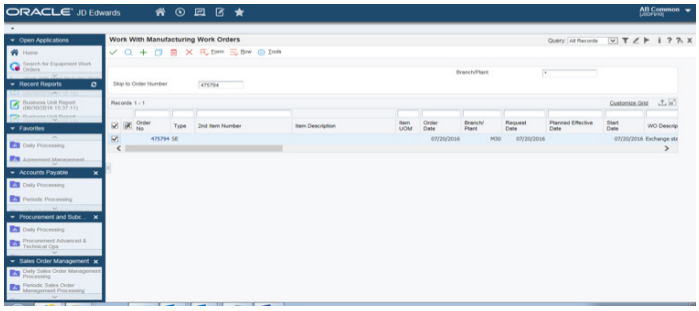


Figure 10: Work Order Console – JDE

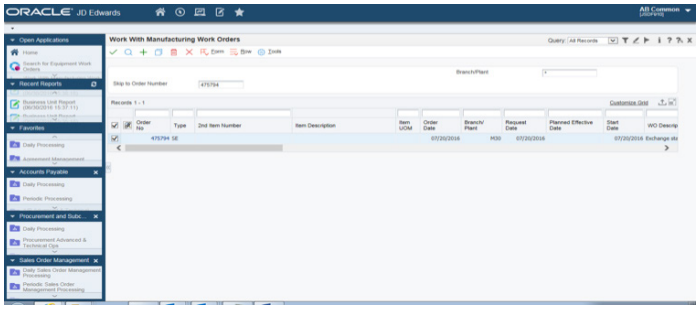


Figure 11: Work Order Console – JDE

Business Benefits of the Oracle IoT Cloud Service

By using the Oracle IoT Cloud Service with Oracle Service Bus, the Mobile Application Framework and JD Edwards Enterprise One, the smart water deionization solution simplifies the maintenance process while enhancing the overall customer experience. Customers are

able to reduce inventory by 33 percent, while gaining 24/7 real-time monitoring, alerting and predictive and preventative maintenance for their plants. The cylinders at customer premises are tracked in real time, which prevents from events like theft and reverse engineering by unauthorized source. The mobile application provides higher visibility over demand and supply for service providers, as well quick service responses powered by process automation.

Overall, manual intervention is greatly reduced, which leads to fewer errors and higher process efficiency. Users are also able to leverage data analytics to gain greater insights into their business.

Amit Mehetre is an Oracle FMW and PaaS Solution Architect with L&T Infotech for digital platform. Responsible for enabling enterprises on digital journey.

Contact

Ashish Thakkar is an Oracle IOT Solution Architect with L&T Infotech. Responsible for designing and implementing digital solutions for enterprises.

Contact

Ankit Thakker is an Oracle IoT and Digital Consultant, specialized in Device - Cloud Integration and designing IoT solutions.

Contact

Migration to 12*c* Made Easy Using Replication Technology

By Donna Guazzaloca-Zehl ■ Michelle Malcher, Editor

It has become critical for businesses to keep up with the latest release for their software assets.

Performance, security and vendor supportability are three motivators that are front and center. Businesses that don't protect their sensitive data become targets of malicious individuals, ultimately putting individuals, as well as the entire company, at risk.

Realizing this risk, Oracle makes security a priority to its customers by also releasing Critical Patch Update Advisories and Security Alerts quarterly for all supported releases. It contains a collection of patches for multiple security vulnerabilities.

Why have organizations been hesitant to migrate their databases?

Upgrades and migrations are considered to be business as usual for IT departments, but to businesses, they can be seen as risky and problematic. If the database is down, this could result in the inability to service customers, which ultimately leads to loss of revenue for the company.

For some industries, this could mean SLA violations and high financial penalties. Data of all kinds is collected, analyzed and used to make critical business decisions 24/7. The days of the regular maintenance windows that were used to perform these types of activities are gone. Availability requirements of the database and contractual SLAs are higher than ever. This has reduced a database administrator's time to upgrade or migrate their database to minutes or less per year depending on the nines availability (Figure 1) they have to do the work in. Businesses see this as a pain point from many angles, including time and dedicated resources from multiple specialized technology teams.

Lack of time, but also the intangible costs, are part of why these upgrades and migrations are put off. Costs to maintain adequate staff and IT infrastructure have been increasing over time, while budgets have been slashed repeatedly every year. 'Do more with less' has pushed the envelope for businesses to come up with a solution that doesn't include adding headcount or incurring additional expenses.

Companies with technology savvy management are now coming on board with cloud technology such as Oracle Cloud, AWS and others to be another tool in their wheelhouse to address these challenges. Many businesses are finding that migrating to the cloud is an attractive option as long as security can be ensured and data protected from loss and from exposure. The costs to maintain a full DR site or a hosting center can be very expensive. Though the adoption of cloud technology has been slower than expected, the time may be here for consideration.

Availability %	Downtime per year
99.8%	17.52 hours
99.9% ("three nines")	8.76 hours
99.95%	4.38 hours
99.99% ("four nines")	52.56 minutes
99.999% ("five nines")	5.26 minutes
99.9999% ("six nines")	31.5 seconds
99.99999% ("seven nines")	3.15 seconds

Source: Wikipedia, "High Availability"
http://en.wikipedia.org/wiki/High_availability

Figure 1

Staying current with the most recent versions of hardware, operating systems and databases is key to reducing business operational risks. This is important not only for improved functionality, but also for data security and for being able to pass more stringent audits for critical systems. A few of the Oracle 12*c* new security features include the inclusion of data redaction, support for secure hash algorithm SHA-2 for DBMS_CRYPTO, and more granular privilege hierarchy for separation of duty to reduce the dependence on SYSDBA.

Can I use a physical standby database to migrate from on-premise to cloud?

Physical replication has always best been known for Disaster Recovery. Examples of this technology in use are Oracle Enterprise Edition Data Guard and Dbvisit Standby for Oracle SE, SE2 and XE Editions. Physical replication is a binary copy of the primary database whereby changes are applied at the lowest available level within the DBMS. A standby database is kept up to date by recovering archive logs that were shipped from the primary and applied to the standby database.

Physical replication can be utilized to migrate to the cloud, data center or a hosting service by building a standby database and performing a graceful switchover — provided they are the same operating system and version of Oracle. This makes the transition between locations or cloud simple to configure and manage with minimal risk.

What about using logical replication?

Logical replication is a technology that has been around for around 15+ years. Logical replication has two methods of implementation. The first is a trigger-based solution which requires additional changes on the primary database and implications of those changes to the application need to be understood.

The second, is the Oracle-preferred method, of logical replication that reads or mines the redo log of a running production system in real time. This requires no code changes or additions to the primary database and requires no interruption in service for your business application. The mine process runs outside of the database in the background on the primary database and an apply process in the background on the target database. Examples of standalone products that use this mining technology are Oracle GoldenGate and Dbvisit Replicate. These products maintain real-time or near real-time data synchronization.

How does logical replication function?

Logical replication enables you to configure replication on a running production system to move data to another database with no downtime. Besides requiring no changes to the primary database or source database, logical replication allows you much more flexibility in your definition of your replication set. You can do some cleanup of temporary schemas during the process. Logical replication has the flexibility to populate schemas or tables, and rename tables or schemas on target database, e.g., scott.dept to tom.department. It also allows you to exclude tables from a schema or exclude a sensitive data column from a replicated table if it is no longer required.

With logical replication, Oracle stores changes to the data or structure via DML (Data Manipulation Language) or DDL (Data Definition Language) in the Oracle redo log. Oracle uses a redo structure to store all these changes made to the database as they occur. The logs are scanned for transactions on the objects in the replication set. A background mining process is used to extract them and place them in a trail or parsed log file for example. The file with the extracted transactions is then sent to the target database. The file can be encrypted as well as compressed to mitigate risk if the file is intercepted while sending across the WAN or to the cloud. Once these files are present on the target server an apply process reads the file and converts it to native SQL. The native SQL is then run on the target database directly or via a SQL*Net connection. Because logical replication uses SQL statements to replicate data, the Oracle version as well as the operating system of the server do not have to be the same.

Logical replication is hardware, operating system and Oracle-version agnostic. This provides a nice alternative if you are changing hardware, server operating systems or, in this case, a version of Oracle. For example, a primary or source database may be on Oracle 10*g* for Windows and the target database may be Oracle Linux 7 with Oracle database version 12*c*. Logical replication provides a solution to eliminate down time required with other alternative strategies and reduces risk overall.

Sounds like logical replication has it all, so why aren't more people using it to solve their business challenges? A flexible technology is more complex to configure, and requires care in setting up and a skilled DBA to oversee this process. If one takes the time to understand how this works and explains the advantages to management, they will be be able to better understand the benefits of this solution.

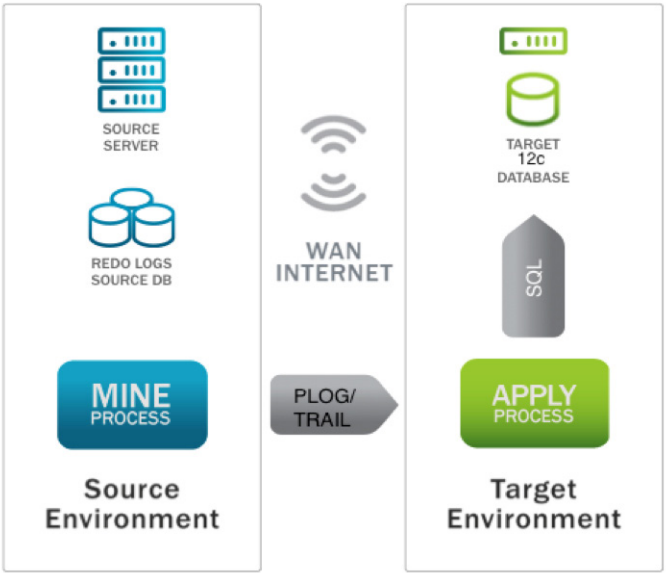


Figure 2: Logical Replication Architecture

Is any preparation needed for logical replication?

Yes, there are some preliminary steps you need to complete during your planning phase. First and foremost is to understand how the logical replication method works. Additionally, here is a checklist for prescreening your source database that must be included.

1. **Check for data types.** The tool may not support all data types. Develop strategies for working around them to get data on the target.
2. **Understand your Redo Volume.** Is there a quieter time where activities on the database are less to instantiate data in the target database?
3. **Check for triggers.** You don't want/need them to fire on the target during the migration. This means they will need to be disabled prior to cutover and enabled at cutover time.
4. **Identify use of Oracle Sequences.** Does your application use Oracle Sequences? These will need to be recreated on your 12*c* target database at time of cutover. Dynamic SQL scripts can be prepared in advance for this during your testing.
5. **Verify space of primary database.** Does your primary database have enough space on it to keep the archive logs on disk during the instantiation period of the target?
6. **TEST, TEST, TEST.** This includes performing migrations in your lower environments first to work out the process and prepare for production migration.

What happens if some other failure occurs at cutover to 12*c* target?

Myriad issues may occur when connecting to the new 12*c* target database resulting from not testing and validating connection and connectivity to the database. One cannot overstress the importance of a solid and comprehensive test plan. The better the test plan, the better results you will receive. Since the logically replicated target database is open, you can connect through your application and run reports and validate before cutover.

You can replicate your production database to multiple servers on premise, to another data center or to the cloud. This would be a one-source-to-many-target database configuration where one target database may be on premise and the other database running in the cloud. The transactions are configured to replicate using one MINE process and two APPLY processes, one running on each target server. When both target databases are in sync, you can shutdown replication to this second target and perform regression testing with your application with the new version of Oracle.

> Logical replication is hardware, operating system and Oracle-version agnostic. This provides a nice alternative if you are changing hardware, server operating systems or, in this case, a version of Oracle.

What happens on the day of cutover if you have issues? Your current production system is intact and, therefore, all you need to do is repoint your application back to your original system. Nothing has changed and there is no risk to your data. Few technologies can provide this level of protection.

Summary

The key takeaway is that your production data is safe and there is no impact to availability implementing a logical replication migration. Tools that utilize logical replication are very effective for migrating to Oracle 12*c* and offer zero or near-zero downtime for your business. You can keep an Oracle 9*i* instance in sync with a 12*c* database for example until time of cutting over the application.

With logical replication, it is efficient to migrate to Oracle 12*c*, change hardware platforms and change operating systems. It supports migrating Oracle databases to a cloud service provider, a hosted service provider or a new data center. Logical replication can provide added value by offloading operational reporting, information sharing, and population of other databases such as MySql, and Sql*Server. It can feed Big Data platforms and Snowflake cloud data warehouse targets, too. It is also a viable use for replacing depreciated technologies, such as advanced replication and streams. This is multi-use technology that offers and addresses the challenges businesses face today with their database mitigating risk and in remaining operational 24/7.

Donna Guazzaloca-Zehl is a senior pre-sales database engineer at Dbvisit. She has a 30+ year career implementing Oracle solutions and managing Oracle technology teams. She is pasionate about Oracle database technology and works with a team of leading experts. Visit www.DBvisit.com for additional information.

Contact

Regression Analyzed, *Tout de Suite:* Leveraging Oracle 12*c* Database Real Application Testing (RAT) Suite, Part 2

By Jim Czuprynski ■ Nick Marcovecchio, Editor

The first article in this series discussed the new features of RAT in 12*c*R1 and illustrated how to capture an actual application workload from a pre-12*c* Oracle database for eventual replay and analysis against an Oracle 12*c* database.

In this second and final part of this series, we illustrate how to use the latest features of RAT in Oracle Database 12*c* to replay the workloads previously captured to determine if there will be any significant variance in application performance, as well as how to detect those variances and determine possible solutions.

To recap where Part 1 of this article series ended, we finished using Database Workload Capture to collect two different workloads from an Oracle 11.2.0.3 database — one OLTP workload, and one DSS workload — and saved the resulting shadow capture files for eventual transference to a new platform for Database Workload Replay testing and evaluation. The results of several queries against statistics for those Database Workload Capture sessions are available here.

Replaying a Workload: Preparations

To prepare for Database Replay operations against the new database system — an Oracle Database running the latest beta version of Release 12*c*R2 — I transferred the generated Database Workload Capture files via SFTP to two separate directories on the new destination database's server. I also created a separate directory to hold just the results of the consolidated workloads' execution and any Database Replay activity.

I then set up the destination database environment for replay using the code shown in Listing 1 to create corresponding directory objects for the DSS and OLTP workloads.

```
-----
-- Create directory objects
-----
DROP DIRECTORY DBRCONS;
CREATE DIRECTORY DBRCONS AS '/home/oracle/DBRControl';
GRANT READ, WRITE ON DIRECTORY DBRCONS TO PUBLIC;

DROP DIRECTORY DBCOLTP;
CREATE DIRECTORY DBCOLTP AS '/home/oracle/DBRControl/OLTP';
GRANT READ, WRITE ON DIRECTORY DBCOLTP TO PUBLIC;

DROP DIRECTORY DBCDSS;
CREATE DIRECTORY DBCDSS AS '/home/oracle/DBRControl/DSS';
GRANT READ, WRITE ON DIRECTORY DBCDSS TO PUBLIC;

-----
-- Create WRC user account for Workload Replay Client isolation
-----
DROP USER wrc CASCADE;
```

```
CREATE USER wrc
    IDENTIFIED BY wrc
    DEFAULT TABLESPACE users
    TEMPORARY TABLESPACE temp
    PROFILE DEFAULT
    QUOTA UNLIMITED ON users;

GRANT CONNECT, DBA TO wrc;
```

Listing 1: Preparing for Database Workload Replay

Note that I also created a new Oracle database user account named WRC that I'll use to invoke multiple Workload Replay Client sessions. Using this separate account is a recommended best practice to make it easier to isolate any session(s) that are only involved in replaying workloads against the destination database.

Combining Database Replay Workloads

Next, I'll combine the OLTP and DSS workloads into a single workload schedule using Oracle 12*c*R1's new BEGIN_REPLAY_SCHEDULE procedure of package DBMS_WORKLOAD_REPLAY. The code to perform their combination is detailed in Listing 2.

```
DECLARE
    cpt_id NUMBER;

BEGIN
    DBMS_WORKLOAD_REPLAY.PROCESS_CAPTURE(
        capture_dir => 'DBCOLTP');
    DBMS_WORKLOAD_REPLAY.PROCESS_CAPTURE(
        capture_dir => 'DBCDSS');

    DBMS_WORKLOAD_REPLAY.SET_REPLAY_DIRECTORY(
        replay_dir => 'DBRCONS');

    DBMS_WORKLOAD_REPLAY.BEGIN_REPLAY_SCHEDULE(
        replay_dir_obj => 'DBRCONS'
       ,schedule_name => 'TEST2DESTRUCT');

    cpt_id :=
        DBMS_WORKLOAD_REPLAY.ADD_CAPTURE(
            capture_dir_name => 'DBCOLTP'
           ,start_delay_seconds => 0
           ,stop_replay => FALSE
           ,take_begin_snapshot => TRUE
           ,take_end_snapshot => TRUE
           ,query_only => FALSE
        );

    cpt_id :=
        DBMS_WORKLOAD_REPLAY.ADD_CAPTURE(
            capture_dir_name => 'DBCDSS'
           ,start_delay_seconds => 0
           ,stop_replay => FALSE
           ,take_begin_snapshot => TRUE
           ,take_end_snapshot => TRUE
           ,query_only => FALSE
        );

     DBMS_WORKLOAD_REPLAY.END_REPLAY_SCHEDULE;
END;
/
```

Listing 2: Combining OLTP and DSS Workloads for Database Workload Replay

Let's break down exactly what this step accomplished:

- The PROCESS_WORKLOAD_CAPTURE procedure prepares both the DSS and OLTP workloads — which were recorded against an 11.2.0.3 database — to be replayed against the destination 12.2.0.1 database.
- The SET_REPLAY_DIRECTORY procedure defines exactly where the results of the Database Workload Replay activity will be recorded.
- The call to BEGIN_REPLAY_SCHEDULE defines the Database Workload Replay schedule named TEST2DESTRUCT which includes the DSS and OLTP workloads captured against the source system.
- The ADD_CAPTURE procedure defines several key aspects of the impending Database Workload Replay session for both workloads:
  - CAPTURE_DIR_NAME defines the location of the directories that contain the existing DSS and OLTP workload capture files.
  - Setting START_DELAY_SESSIONS to a value of zero (0) specifies that the workload sessions should begin without any delay. Note that it's also possible to enable a replay to start after time has elapsed — say, to enable a database to attain a desired steady state.
  - Setting STOP_REPLAY to FALSE means that the replay should continue until all captured workloads have been consumed completely.
  - The calls to procedures TAKE_BEGIN_SNAPSHOT and TAKE_END_SNAPSHOT specify that an AWR snapshot should be taken just before the beginning and just after the end of consolidated workload activity.
  - Setting QUERY_ONLY to FALSE specifies that the database should replay all DML statements as well as all queries. If I had set this to TRUE, then only read-only queries would be replayed.
- Finally, the END_REPLAY_SCHEDULE procedure defines the end of the scheduled replay activity.

Initializing a Consolidated Database Workload Replay

Now that I've finished building my Consolidated Replay schedule, my next step is to make sure my destination database is ready to accept that workload schedule. For these tests, I simply reset my database by reloading the data into the AP schema via a series of stored procedures that generated exactly the same data that was present in the source database before I began workload capture.

Of course, in the real world and for a larger database, I could use any number of methods to reset my destination database, including:

- Rewinding the database back to a prior point in time using FLASHBACK DATABASE.
- Employing a snapshot standby version of the original source database.
- Restoring the database from earlier backups using Recovery Manager (RMAN).
- Reloading the database completely via DataPump Import.

To initialize the database for a consolidated replay, I used the code in Listing 3. Procedure INITIALIZE_CONSOLIDATED_REPLAY assigns the task name of DBR_CONS_100 to the TEST2DESTRUCT replay schedule created earlier. Note the use of the REMAP_CONNECTION procedure to remap the database services that user sessions used to connect to the source database during the original Database Workload Capture session to just two database services (OLTP and DSS) during the consolidated Database Workload Replay to come.

```
-----
-- Create . . .
-----

BEGIN
    DBMS_WORKLOAD_REPLAY.INITIALIZE_CONSOLIDATED_REPLAY(
        replay_name => 'DBR_CONS_100'
       ,schedule_name => 'TEST2DESTRUCT');
     DBMS_WORKLOAD_REPLAY.REMAP_CONNECTION(
         connection_id => 1
        ,replay_connection =>
           '(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=ora12201)
(PORT=1521))
            (CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=DSS)))');
     . . .
     << other REMAP_CONNECTION calls removed for sake of brevity >>
     . . .
     DBMS_WORKLOAD_REPLAY.REMAP_CONNECTION(
         connection_id => 6
        ,replay_connection =>
           '(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=ora12201)
(PORT=1521))
            (CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=OLTP)))');

     . . .
END;
/
```

Listing 3: Initializing Database Workload Replay

Starting a Consolidated Database Workload Replay

Next, I'll start up two Workload Replay Client (WRC) sessions using the script shown in Listing 4. Note that I'm using the WRC user account to log into the destination database so that any additional activity generated by WRC sessions is extraneous to the workloads being replayed.

```
#####
# Start Workload Replay Client(s)
#####

$> wrc wrc/wrc mode=replay replaydir=/home/oracle/DBRControl
```

Listing 4: Starting Workload Replay Client Sessions

Now that the Workload Replay Client sessions are running, I'll start consuming the captured workloads by kicking off the Consolidated Replay with a call to procedure START_CONSOLIDATED_REPLAY, as shown in Listing 5. Note that setting START_REPLAY_TIMEOUT to FALSE turns off the default action of the WRC, which is to terminate any user call that may inadvertently cause a replay to run slowly or possibly cause it to hang.

```
-----
-- Start the Consolidated Replay
-----

BEGIN
    DBMS_WORKLOAD_REPLAY.SET_REPLAY_TIMEOUT(FALSE);
    DBMS_WORKLOAD_REPLAY.START_CONSOLIDATED_REPLAY;
END;
/
```

Listing 5: Starting a Consolidated Workload Replay

Monitoring Consolidated Database Workload Replay Progress

As the Consolidated Database Workload Replay unwound, I monitored its progress through several panels via *Oracle Enterprise Manager 13c Cloud Control* (EM13*c*CC). As its name suggests, the Database Replay panel (see Figure 1) displayed the overall progress of the ongoing Database Replay operation, including a progress bar reflecting the time it took to play back the original workloads as compared to the current consolidated workloads. It also provided tallies of any regression observed so far — for example, a variance in the number of rows queried or modified, or the number of errors observed — between the captured and replay workloads.

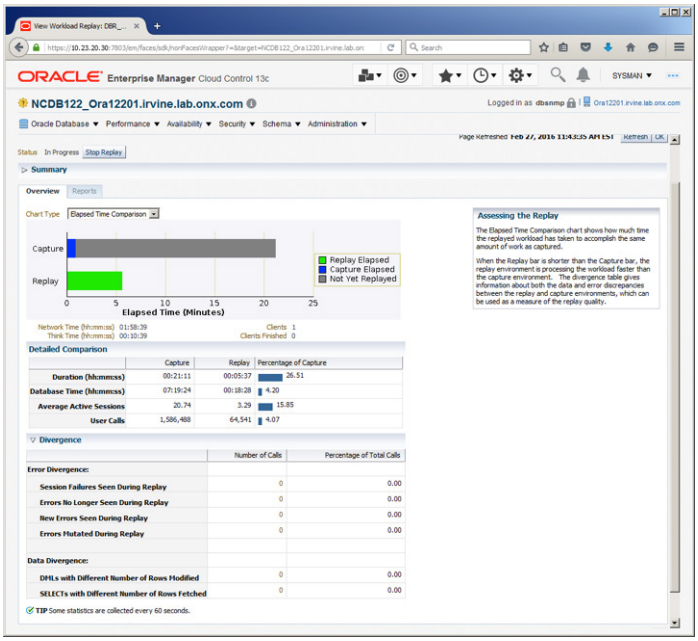


Figure 1: Database Workload Replay Progress Monitoring

Figure 2 shows a snapshot of the Database *Performance Monitoring* panel as the workload was replayed. Note that the bottom half of this panel allowed me to see how the different database services were consuming the captured workload and replaying it in the new environment.

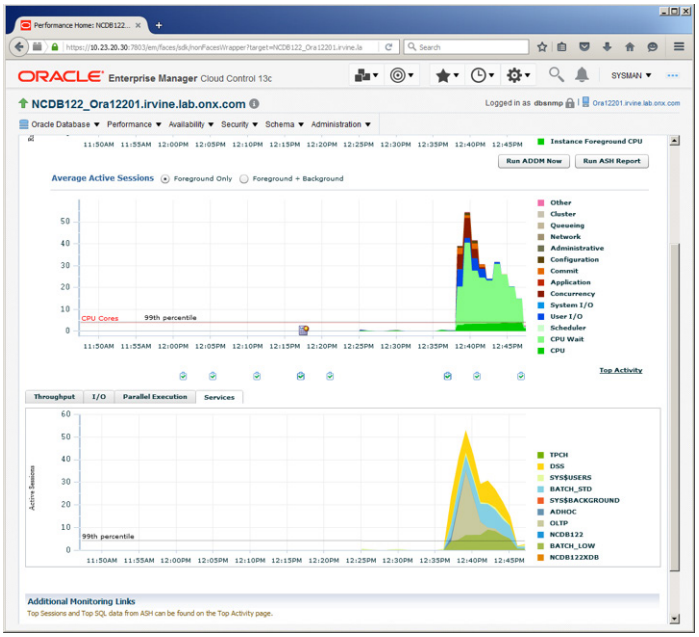


Figure 2: Database Workload Replay: Database Performance Monitoring

Finally, I kept an eye on the state of my database's Database In-Memory (DBIM) memory utilization using one of the newest features of OEM13*c*CC, the *In-Memory Central* panel. This screenshot reflects the fact that during this particular run of Consolidated Workload Replay, I had enabled population of several key tables into *In-Memory Column Store* (IMCS) so that I could evaluate the impact of enabling Database In-Memory features as part of my beta-testing activities for Release 12*c*R2. Figure 3 shows the results of monitoring that aspect of the consolidated replay.

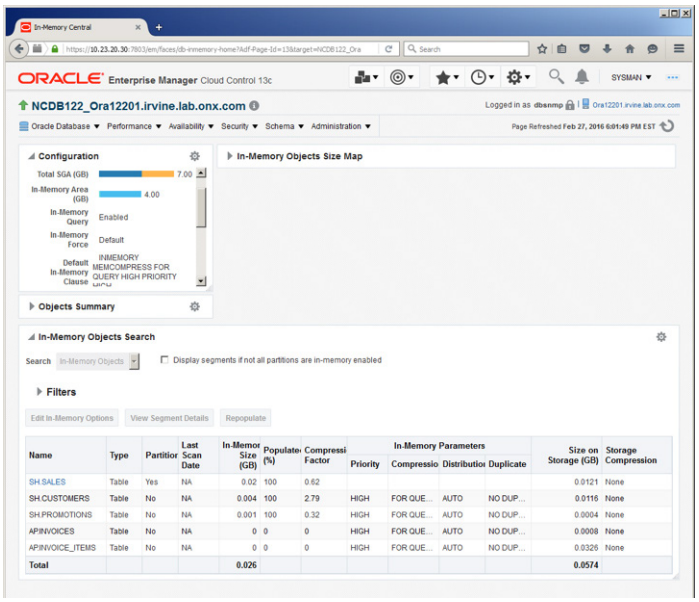


Figure 3: Database Workload Replay: Monitoring In-Memory Area Via In-Memory Central

Analyzing Consolidated Database Workload Replay Results

After the Consolidated Database Workload Replay completed its execution, I captured some statistics from the results via the queries in Listing 6; their output is shown in Listing 7.

```
SELECT
     id
    ,name
    ,schedule_name
    ,capture_id
    ,user_calls
    ,status
    ,divergence_load_status DLS
    ,TO_CHAR(start_time,'yyyy-mm-dd hh24:mi:ss') start_dtm
    ,TO_CHAR(end_time,'yyyy-mm-dd hh24:mi:ss') end_dtm
    ,duration_secs
    ,num_clients
    ,num_clients_done
    ,(dbtime/1000000) dbtime_ss
    ,(think_time/1000000) think_time_ss
    ,(network_time/1000000) network_time_ss
    ,pause_time
FROM dba_workload_replays;

SELECT
     id
    ,name
    ,dbname
    ,dbversion
    ,directory
    ,capture_id
    ,user_calls
    ,status
    ,TO_CHAR(prepare_time,'yyyy-mm-dd hh24:mi:ss') prepare_dtm
    ,synchronization
    ,connect_time_scale
    ,think_time_scale
    ,think_time_auto_correct TTAC
    ,scale_up_multiplier SCM
    ,default_action
  FROM dba_workload_replays;
```

Listing 6: Tracking Consolidated Workload Replay Progress

Database In-Memory (DBIM) memory utilization using one of the newest features of OEM13cCC, the In-Memory Central panel.

```
                                                                     Think
                   # of              Replay                    Think Time      Scale
Rply Replay        User              Prepare     Synch       Connect  Time Auto        Up Default
 ID# Operation    Calls Status       Time        ronized?    TimeScale Scale Corre Multipler Action
---- ------------ ------- ------------ ----------- ---------- --------- ------ ----- --------- ----------
 167 DBR_CONS_100 1943044 COMPLETED    2016-02-29  SCN               0      0 FALSE         1 INCLUDE
                                       18:22:04

 157 DBR_CONS_100 1943044 COMPLETED    2016-02-29  SCN               0      0 FALSE         1 INCLUDE
                                       16:17:57

 113 DBR_CONS_100 1943044 COMPLETED    2016-02-28  SCN               0      0 FALSE         1 INCLUDE
                                       17:36:42

 102 DBR_CONS_100 1943044 COMPLETED    2016-02-28  SCN               0      0 FALSE         1 INCLUDE
                                       17:00:50


          # of            Replay       Replay       Replay  Total  Clnt                 Think   Netwk  Pause
Rply      User            Start        End            Time   Clnt  Sess     DB Time      Time    Time   Time
 ID#     Calls Status     Time         Time            (s)   Sess  Done         (s)       (s)     (s)    (s)
---- -------- -------------------- ----------- ------ ------ ------ ---------- ---------- ------- ------
 167   1943044 COMPLETED 2016-02-29  2016-02-29    629      2      2   19667.10  162448.47 3486.92      0
                         18:23:04    18:33:33

 157   1943044 COMPLETED 2016-02-29  2016-02-29    872      2      2   35494.17  201475.82 4222.84      0
                         16:18:33    16:33:05

 113   1943044 COMPLETED 2016-02-28  2016-02-28    563      2      2   18411.01  174882.85 3701.86      0
                         17:37:12    17:46:35

 102   1943044 COMPLETED 2016-02-28  2016-02-28    537      2      2   16963.72  168811.75 3852.33      0
                         17:01:22    17:10:19
```

Listing 7: Consolidated Workload Replay Statistics

In actuality, these reports show the results from several tests I had run over a period of 24 hours with varying results, but I'll focus in on the latest run (#167) in the following sections as we analyze the results from the consolidated replay activity. Listing 8 shows the code I used to report against that last Consolidated Database Workload Replay run. Note that the output from that report (Listing 9) shows that a consolidated replay task definitely occurred, that it ran against an Oracle 12*c*R2 database, and that it was originally captured from an 11.2.0.3 database. Of course, this information is also available from the Database Workload Replay panel in HTML format.

Real Application Testing in Oracle Database 12c to replay previously-captured workloads concurrently, even when they were recorded during completely different timeframes.

```
VARIABLE replay_report CLOB;
VARIABLE capture_id NUMBER;
VARIABLE replay_id NUMBER;
BEGIN
    :replay_id := 167;
    :replay_report :=
               DBMS_WORKLOAD_REPLAY.REPORT(
             replay_id => :replay_id
                         ,format => 'TEXT');
    END;
END;
/
SET LONGCHUNKSIZE 1000
SET LONG 30000000
SET LINESIZE 2000
SET PAGESIZE 0
PRINT :replay_report
```

Listing 8: Workload Replay Report Generation

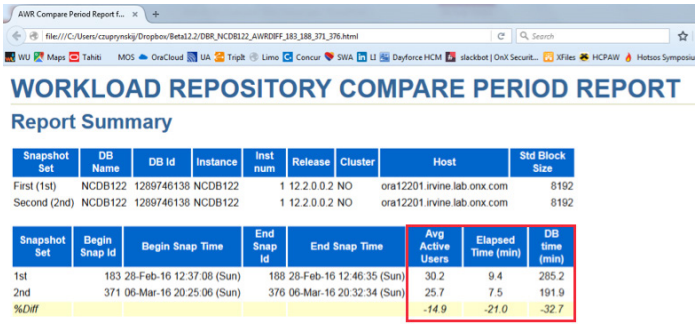
```
-------------------------------------------------------------------------------
| DB Name | DB Id      | Release    | RAC | Replay Name  | Replay Status |
-------------------------------------------------------------------------------
| NCDB122 | 1289746138 | 12.2.0.0.2 | NO  | DBR_CONS_100 | COMPLETED     |
-------------------------------------------------------------------------------
Replay Information
-----------------------------------------------
|     Information      | Replay               |
-----------------------------------------------
| Name                 | DBR_CONS_100         |
-----------------------------------------------
| Status               | COMPLETED            |
-----------------------------------------------
. . .
-----------------------------------------------
| AWR DB Id            | 1289746138           |
-----------------------------------------------
| AWR Begin Snap Id    | 371                  |
-----------------------------------------------
| AWR End Snap Id      | 376                  |
-----------------------------------------------
| PL/SQL Capture Mode  | TOP_LEVEL            |
-----------------------------------------------
| Replay Schedule Name | TEST2DESTRUCT        |
-----------------------------------------------
. . .

. . .
Replay Schedule Information Schedule TEST2DESTRUCT
      contains 2 captures:
-----------------------------------------------------
|       Information       | Capture 1               |
-----------------------------------------------------
| Name                    | DBR_CAPTURE_100         |
-----------------------------------------------------
| Status                  | COMPLETED               |
-----------------------------------------------------
| Database Name           | ORCL                    |
-----------------------------------------------------
| Database Version        | 11.2.0.3.0              |
-----------------------------------------------------
| Start Time              | 24-02-16 12:27:13       |
-----------------------------------------------------
| End Time                | 24-02-16 12:32:48       |
-----------------------------------------------------
| Duration                | 5 minutes 35 seconds    |
-----------------------------------------------------
| DB Time                 | 7363.499 seconds        |
-----------------------------------------------------
| User Calls              | 356556                  |
-----------------------------------------------------
| Average Active Sessions | 21.98                   |
-----------------------------------------------------
| Directory Object        | DBCOLTP                 |
-----------------------------------------------------
. . .
<< remainder of report removed for sake of brevity >>
. . .
```

Listing 9: Workload Replay Report Output

Now let's take a deeper look into whether this new configuration of my database was able to handle the consolidated workload that unwound during this Database Workload Replay task. The easiest way to tackle that topic is with tools we've been using since Oracle 11*g*R1 — running an *Automatic Workload Repository (AWR) Compare Periods Report.*

Figure 4.1 shows some interesting initial results: The consolidated workload appears to have completed with significantly diminished Database Time (285.2 minutes vs. 191.9 minutes), a 32.7 percent improvement. Also, it shaved almost 21 percent off of the original elapsed time for the two separate captured workloads. So far, so good — we're not seeing any apparent performance degradation at a database level.



WORKLOAD REPOSITORY COMPARE PERIOD REPORT

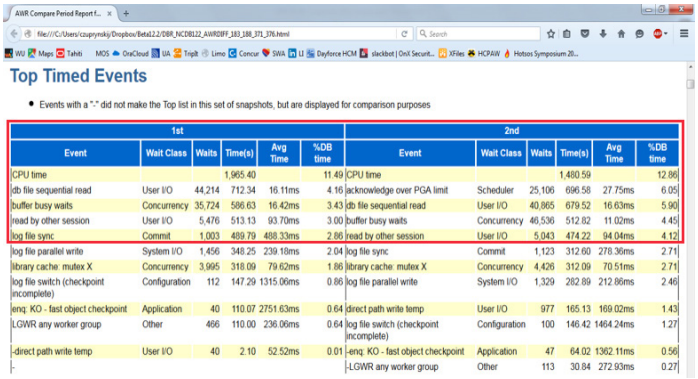
Report Summary

Snapshot Set	DB Name	DB Id	Instance	Inst num	Release	Cluster	Host	Std Block Size
First (1st)	NCDB122	1289746138	NCDB122	1	12.2.0.0.2	NO	ora12201.irvine.lab.onx.com	8192
Second (2nd)	NCDB122	1289746138	NCDB122	1	12.2.0.0.2	NO	ora12201.irvine.lab.onx.com	8192

Snapshot Set	Begin Snap Id	Begin Snap Time	End Snap Id	End Snap Time	Avg Active Users	Elapsed Time (min)	DB time (min)
1st	183	28-Feb-16 12:37:08 (Sun)	188	28-Feb-16 12:46:35 (Sun)	30.2	9.4	285.2
2nd	371	06-Mar-16 20:25:06 (Sun)	376	06-Mar-16 20:32:34 (Sun)	25.7	7.5	191.9
%Diff					-14.9	-21.0	-32.7

Figure 4.1: AWR Results: Summary

Meanwhile, Figure 4.2 compares the Top Timed Events between captured and replayed consolidated workloads. This report snippet looks promising as well: CPU time decreased slightly, but a new wait event — *Acknowledge Over PGA Limit* — has appeared. However, this does make sense because this event is directly related to the population and use of the In-Memory Column Store in the newest release of the database.



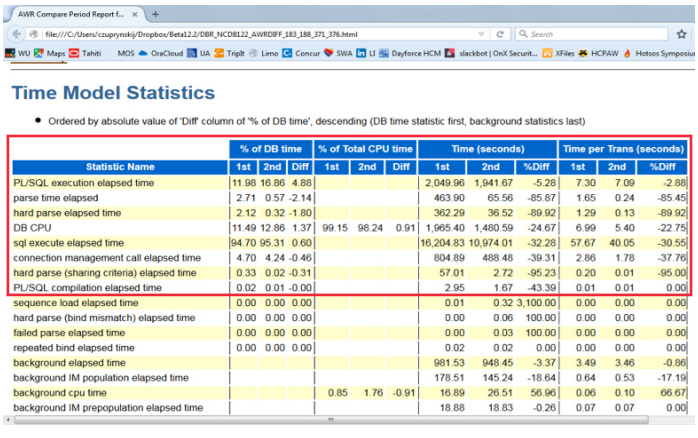
Top Timed Events

- Events with a "-" did not make the Top list in this set of snapshots, but are displayed for comparison purposes

1st						2nd					
Event	Wait Class	Waits	Time(s)	Avg Time	%DB time	Event	Wait Class	Waits	Time(s)	Avg Time	%DB time
CPU time			1,965.40		11.49	CPU time			1,480.59		12.86
db file sequential read	User I/O	44,214	712.34	16.11ms	4.16	acknowledge over PGA limit	Scheduler	25,106	696.58	27.75ms	6.05
buffer busy waits	Concurrency	35,724	586.63	16.42ms	3.43	db file sequential read	User I/O	40,865	679.52	16.63ms	5.90
read by other session	User I/O	5,476	513.13	93.70ms	3.00	buffer busy waits	Concurrency	46,536	512.82	11.02ms	4.45
log file sync	Commit	1,003	489.79	488.33ms	2.86	read by other session	User I/O	5,043	474.22	94.04ms	4.12
log file parallel write	System I/O	1,456	348.25	239.18ms	2.04	log file sync	Commit	1,123	312.60	278.36ms	2.71
library cache: mutex X	Concurrency	3,995	318.09	79.62ms	1.86	library cache: mutex X	Concurrency	4,426	312.09	70.51ms	2.71
log file switch (checkpoint incomplete)	Configuration	112	147.29	1315.06ms	0.86	log file parallel write	System I/O	1,329	282.89	212.86ms	2.46
enq: KO - fast object checkpoint	Application	40	110.07	2751.63ms	0.64	direct path write temp	User I/O	977	165.13	169.02ms	1.43
LGWR any worker group	Other	466	110.00	236.06ms	0.64	log file switch (checkpoint incomplete)	Configuration	100	146.42	1464.24ms	1.27
direct path write temp	User I/O	40	2.10	52.52ms	0.01	enq: KO - fast object checkpoint	Application	47	64.02	1362.11ms	0.56
						LGWR any worker group	Other	113	30.84	272.93ms	0.27

Figure 4.2: AWR Results: Top Timed Events

As Figure 4.3 illustrates, there's some good news in the *Time Model Statistics* section of the report as well. *SQL Execution time* has dropped by a factor of nearly 31 percent, and SQL parse times have improved as well.



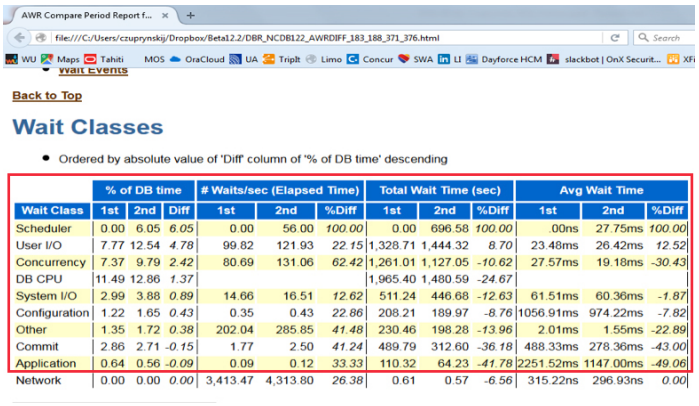
Time Model Statistics

- Ordered by absolute value of 'Diff' column of '% of DB time', descending (DB time statistic first, background statistics last)

	% of DB time			% of Total CPU time			Time (seconds)			Time per Trans (seconds)		
Statistic Name	1st	2nd	Diff	1st	2nd	Diff	1st	2nd	%Diff	1st	2nd	%Diff
PL/SQL execution elapsed time	11.98	16.86	4.88				2,049.96	1,941.67	-5.28	7.30	7.09	-2.88
parse time elapsed	2.71	0.57	-2.14				463.90	65.56	-85.87	1.65	0.24	-85.45
hard parse elapsed time	2.12	0.32	-1.80				362.29	36.52	-89.92	1.29	0.13	-89.92
DB CPU	11.49	12.86	1.37	99.15	98.24	0.91	1,965.40	1,480.59	-24.67	6.99	5.40	-22.75
sql execute elapsed time	94.70	95.31	0.60				16,204.83	10,974.01	-32.28	57.67	40.05	-30.55
connection management call elapsed time	4.70	4.24	-0.46				804.89	488.48	-39.31	2.86	1.78	-37.76
hard parse (sharing criteria) elapsed time	0.33	0.02	-0.31				57.01	2.72	-95.23	0.20	0.01	-95.00
PL/SQL compilation elapsed time	0.02	0.01	-0.00				2.95	1.67	-43.39	0.01	0.01	0.00
sequence load elapsed time	0.00	0.00	0.00				0.01	0.32	3,100.00	0.00	0.00	0.00
hard parse (bind mismatch) elapsed time	0.00	0.00	0.00				0.00	0.06	100.00	0.00	0.00	0.00
failed parse elapsed time	0.00	0.00	0.00				0.00	0.03	100.00	0.00	0.00	0.00
repeated bind elapsed time	0.00	0.00	0.00				0.02	0.02	0.00	0.00	0.00	0.00
background elapsed time							981.53	948.45	-3.37	3.49	3.46	-0.86
background IM population elapsed time							178.51	145.24	-18.64	0.64	0.53	-17.19
background cpu time				0.85	1.76	-0.91	16.89	26.51	56.96	0.06	0.10	66.67
background IM prepopulation elapsed time							18.88	18.83	-0.26	0.07	0.07	0.00

Figure 4.3: AWR Results: Time Model Statistics

Figure 4.4 shows some interesting information, too. This snippet of the report summarizes database activity within Wait Classes, and this reveals some apparently significant "pushback" against the database workloads during replay. This was most likely due to a Database Resource Manager (DBRM) plan directive that was active during the replay and definitely warrants additional investigation, especially since that "pushback" wasn't apparent during the previously captured workloads. Again, this doesn't necessarily mean we've encountered performance regression, but it could be the result of an errant DBRM plan directive that was enabled during replay, or it could even be related to how 12*c*R2 now manages parallelism during population of the In-Memory Column Store.



Wait Events

Back to Top

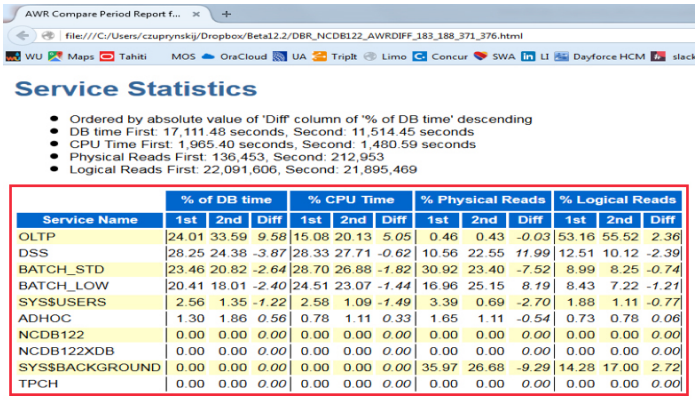
Wait Classes

- Ordered by absolute value of 'Diff' column of '% of DB time' descending

	% of DB time			# Waits/sec (Elapsed Time)			Total Wait Time (sec)			Avg Wait Time		
Wait Class	1st	2nd	Diff	1st	2nd	%Diff	1st	2nd	%Diff	1st	2nd	%Diff
Scheduler	0.00	6.05	6.05	0.00	56.00	100.00	0.00	696.58	100.00	.00ns	27.75ms	100.00
User I/O	7.77	12.54	4.78	99.82	121.93	22.15	1,328.71	1,444.32	8.70	23.48ms	26.42ms	12.52
Concurrency	7.37	9.79	2.42	80.69	131.06	62.42	1,261.01	1,127.05	-10.62	27.57ms	19.18ms	-30.43
DB CPU	11.49	12.86	1.37				1,965.40	1,480.59	-24.67			
System I/O	2.99	3.88	0.89	14.66	16.51	12.62	511.24	446.68	-12.63	61.51ms	60.36ms	-1.87
Configuration	1.22	1.65	0.43	0.35	0.43	22.86	208.21	189.97	-8.76	1056.91ms	974.22ms	-7.82
Other	1.35	1.72	0.38	202.04	285.85	41.48	230.46	198.28	-13.96	2.01ms	1.55ms	-22.89
Commit	2.86	2.71	-0.15	1.77	2.50	41.24	489.79	312.60	-36.18	488.33ms	278.36ms	-43.00
Application	0.64	0.56	-0.09	0.09	0.12	33.33	110.32	64.23	-41.78	2251.52ms	1147.00ms	-49.06
Network	0.00	0.00	0.00	3,413.47	4,313.80	26.38	0.61	0.57	-6.56	315.22ns	296.93ns	0.00

Figure 4.4: AWR Results: Wait Classes Detail

Finally, Figure 4.5 shows some interesting results, too, but this time at the Database Service level: Only the workload sessions played back via the OLTP database service show any significant regression in DB and CPU time, while most of the other application sessions show slight improvement in those categories.



Service Statistics

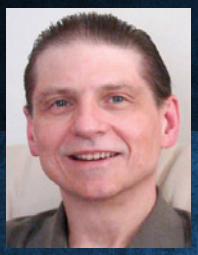
- Ordered by absolute value of 'Diff' column of '% of DB time' descending
- DB time First: 17,111.48 seconds, Second: 11,514.45 seconds
- CPU Time First: 1,965.40 seconds, Second: 1,480.59 seconds
- Physical Reads First: 136,453, Second: 212,953
- Logical Reads First: 22,091,606, Second: 21,895,469

	% of DB time			% CPU Time			% Physical Reads			% Logical Reads		
Service Name	1st	2nd	Diff	1st	2nd	Diff	1st	2nd	Diff	1st	2nd	Diff
OLTP	24.01	33.59	9.58	15.08	20.13	5.05	0.46	0.43	-0.03	53.16	55.52	2.36
DSS	28.25	24.38	-3.87	28.33	27.71	-0.62	10.56	22.55	11.99	12.51	10.12	-2.39
BATCH_STD	23.46	20.82	-2.64	28.70	26.88	-1.82	30.92	23.40	-7.52	8.99	8.25	-0.74
BATCH_LOW	20.41	18.01	-2.40	24.51	23.07	-1.44	16.96	25.15	8.19	8.43	7.22	-1.21
SYS$USERS	2.56	1.35	-1.22	2.58	1.09	-1.49	3.39	0.69	-2.70	1.88	1.11	-0.77
ADHOC	1.30	1.86	0.56	0.78	1.11	0.33	1.65	1.11	-0.54	0.73	0.78	0.06
NCDB122	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
NCDB122XDB	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SYS$BACKGROUND	0.00	0.00	0.00	0.00	0.00	0.00	35.97	26.68	-9.29	14.28	17.00	2.72
TPCH	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 4.5: AWR Results: Service Statistics

Conclusion

In this final part of this article series, we demonstrated how to leverage the latest features of Real Application Testing in Oracle Database 12*c* to replay previously-captured workloads concurrently, even when they were recorded during completely different timeframes. We also discussed how to ramp up those captured workloads so that we could perform a test to destruction against the target database in its new 12*c* environment. Finally, we illustrated how easy it is to determine if the new destination system is able to handle the stresses of those ramped-up workloads using existing tools like AWR Compare Period Reports to quickly determine potential vectors for performance improvement or regression.



Jim Czuprynski has 35+ years of professional experience in his career in information technology. During that time, he served diverse roles at several Fortune 1000 companies — mainframe programmer, applications developer, business analyst and project manager — before becoming an Oracle DBA in 2001. Jim was awarded the status of Oracle ACE Director in March 2014 and is a sought-after public speaker on Oracle Database technology features. He has presented topics at Oracle OpenWorld, IOUG's COLLABORATE, Hotsos Symposium, Oracle Technology Network ACE Tours and Oracle User Group conferences around the world.

In August 2015, Jim joined OnX Enterprise Solutions as their Strategic Solutions Architect for Oracle Database technology, focusing on Oracle Database 12*c* and Oracle's Enterprise Systems, including Oracle Database Appliance, Exadata Database Machine, SPARC Supercluster and ZFS Storage Appliance.

Jim has authored over 100 articles focused on facets of Oracle Database administration to his credit since 2003 at databasejournal.com and ioug.org. Jim's blog, *Generally … It Depends* (http://jimczuprynski.wordpress.com), contains his regular observations on all things Oracle.

His most recent book, *Oracle Database Problem Solving and Troubleshooting Handbook*, was published in May 2016 through Pearson.

Contact

Database Auditing: What to Audit and Why

By Frank Pound ■ Michelle Malcher, Editor

In the first article in this series (How to Comply with Audit and Make Your Life Easier), I outlined how to set up a set of management control processes using an existing Information Technology (IT) policy, if one exists.

In this article, I lay out a set of steps that help clearly articulate what should be audited and why (Steps 1 and 2) using national and international security standards as the basis of the rationale.

The basis of many current IT security policies is balancing and setting standards for the protection of Confidentiality, Integrity, and Availability (CIA) of systems. This article deals specifically with database systems, but the steps described can be used for many other types of systems.

National Institute of Standards and Technology (NIST)

A U.S. Information Technology lab (ITL) is dedicated to promoting innovation and industrial competitiveness, which produces a standard to help organizations increase security. They are mandated to develop information security standards and guidelines according to Federal Information Security Management Act (FISMA), Public Law (P.L.) 107-347.

NIST publishes two documents to aid organizations in the task of setting IT security policies:

1. Special Publication 800-53 Security and Privacy Controls for Federal Information Systems and Organizations (revision 4).
2. Special Publication 800-53A Assessing Security and Privacy Controls in Federal Information Systems and Organizations: Building Effective Assessment Plans (revision 4).

Special Publication 800-53A is a companion guideline to Special Publication 800-53. Each publication provides guidance for implementing specific steps in the Risk Management Framework (RMF) (800-37: Guide for Applying the Risk Management Framework to Federal Information Systems). The outlined framework describes six steps as illustrated in Figure 1.

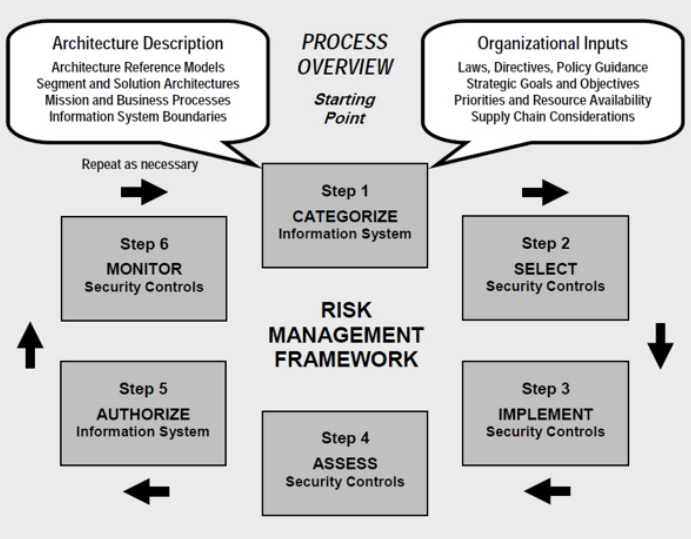


Figure 1: Six Steps of RMF

The RMF apply to each special publication as follows:

- Special Publication 800-53 = RMF Step 2: SELECT the security and privacy control (i.e., determining what controls are needed to manage risks to organizational operations and assets, individuals, other organizations, and the Nation).
- Special Publication 800-53A = RMF Step 4: Assess the control.
- Special Publication 800-53A = RMF Step 6: Monitor the control; and provides guidance on the security assessment and privacy assessment processes.

This guidance includes how to build effective assessment plans and how to analyze and manage assessment results.

FIPS 200 and SP 800-53

The Federal Information Processing Standards (FIPS) 200 is the second of two U.S. Federal standards, which defines a security catalog for categorization of controls. Categorizing the data assets of your organization is the first step in understanding what controls are required to secure your data.

The goal of this set of documents is to help organizations perform their due diligence related to management controls.

The guidance and subsequent management controls are not meant to be static, but rather to be assessed and reevaluated at regular intervals to ensure they are constantly relevant, accurate and valid.

Security Controls and Identifiers	
ID	FAMILY
AC	Access Control
AT	Awareness and Training
AU	Audit and Accountability
CA	Security Assessment and Authorization
CM	Configuration Management
CP	Contingency Planning
IA	Identification and Authentication
IR	Incident Response
MA	Maintenance
MP	Media Protection

Security Controls and Identifiers	
ID	FAMILY
PE	Physical and Environmental Protection
PL	Planning
PS	Personnel Security
RA	Risk Assessment
SA	System and Services Acquisition
SC	System and Communications Protection
SI	System and Information Integrity
PM	Program Management

The security controls identified by NIST are divided into 18 groups/families.

When selecting controls, the organization is responsible to *"determine the most cost-effective, appropriate set of security controls, which if implemented and determined to be effective, would mitigate risk while complying with security requirements defined."*

NIST provides baseline controls to ease the processes of selecting controls and groupings

I strongly recommend reading the NIST documentation to increase your knowledge and skills in selecting, and implementing appropriate controls for your organizations systems.

Before starting a critical pre-requisite is to understand the criticality and sensitivity (statement of sensitivity: SOW) of the data contained in each system.

Baselines are based on the following assumptions:

- Information systems are located in physical facilities
- User data/information in organizational information systems is relatively persistent
- Information systems are multi-user (either serially or concurrently) in operation
- Some user data/information in organizational information systems is not shareable with other users who have authorized access to the same systems

PROPERTY	RATING																
	LOW	MEDIUM	MEDIUM	MEDIUM	MEDIUM	MEDIUM	MEDIUM	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH
CONFIDENTIALITY	L	M	M	M	L	L	L	L	H	H	L	M	L	H	M	H	M
INTEGRITY	L	L	M	M	M	M	L	L	L	M	M	L	H	H	M	M	H
AVAILABILITY	L	L	L	M	M	L	M	H	L	L	H	H	H	H	H	H	H

System Security Control Rating: Property ratings are L = Low, M = Medium, H = High.

- Information systems exist in networked environments
- Information systems are general purpose in nature
- Organizations have the necessary structure, resources and infrastructure to implement the controls.

If the assumptions do not apply to the system for which the control is created, then adjust accordingly. This may mean the implementation of compensating control(s) to compensate for the difference.

As with most control frameworks and architectural decisions, a record of decision is highly recommended and may already be mandated by your organization.

Some controls may be specifically related to one another in an understood flow/sequence. If so, then each relationship should be recorded in a matrix where the ordering of the controls is identified. NIST suggests the use of P0, P1, P2, P3, … Pn. Where P0 indicates no relationship, P1 indicates the first or top priority control followed in ascending order by P2, P3, etc.

Identify Controls Relevant to the Information Systems

After reviewing and becoming comfortable with how security controls are ranked and rated, the next step is to review the set of recommended controls and identify those, which are most relevant to your information (e.g., database) systems.

For brevity, we will review those controls recommended by the NIST standard for Access Control.

In the first round, I recommend focusing on the P1 ranked controls, which are as follows:

CNTL NO.	CONTROL NAME	PRIORITY	INITIAL CONTROL BASELINES		
AC-1	Access Control Policy and Procedures	P1	AC-1	AC-1	AC-1
AC-2	Account Management	P1	AC-2	AC-2 (1) (2) (3) (4)	AC-2 (1) (2) (3) (4) (5) (11) (12) (13)
AC-3	Access Enforcement	P1	AC-3	AC-3	AC-3
AC-4	Information Flow Enforcement	P1	Not Selected	AC-4	AC-4
AC-5	Separation of Duties	P1	Not Selected	AC-5	AC-5
AC-6	Least Privilege	P1	Not Selected	AC-6 (1) (2) (5) (9) (10)	AC-6 (1) (2) (3) (5) (9) (10)
AC-8	System Use Notification	P1	AC-8	AC-8	AC-8
AC-17	Remote Access	P1	AC-17	AC-17 (1) (2) (3) (4)	AC-17 (1) (2) (3) (4)
AC-18	Wireless Access	P1	AC-18	AC-18 (1)	AC-18 (1) (4) (5)
AC-19	Access Control for Mobile Devices	P1	AC-19	AC-19 (5)	AC-19 (5)
AC-20	Use of External Information Systems	P1	AC-20	AC-20 (1) (2)	AC-20 (1) (2)

Typically, all controls require documentation, which is recorded in the first control for each family. In this case it is AC-1: Access Control Policy and Procedures. This is not strictly a control, but is a pre-requisite for the creation and auditing of the controls.

Traditionally, controls AC-2, AC-3, AC-4, AC-5, AC-6, and AC-8 fall under the prevue of database administration. Controls number AC-17, AC-18, AC-19 and AC-20 are traditionally the responsibility of the network and security teams to monitor and control on behalf of many organizations. The controls should be confirmed in writing by the responsible team and kept as proof the database administration team has validated the controls exist and understand where the procedural documentation exists for reference purposes.

Associate All Dependent Sub-Controls

Once the P1 controls have been identified, it is time to identify the P2, P3 and P0 controls to ensure all the controls are included in your security plan.

CNTL NO.	CONTROL NAME	PRIORITY	INITIAL CONTROL BASELINES		
AC-7	Unsuccessful Logon Attempts	P2	AC-7	AC-7	AC-7
AC-12	Session Termination	P2	Not Selected	AC-12	AC-12
AC-21	Information Sharing	P2	Not Selected	AC-21	AC-21

Of the three P2 controls, I envision the controls as being monitored in the following manner:

- AC-7: One report grouped by user and count for the monitoring period.
- AC-7: One report grouped by user by day to understand risk of the unsuccessful logon attempts.
- AC-12: One report listing the configuration of each granted profile listing CONNECT_TIME, and IDLE_TIME.
- AC-12: One report listing the “LOGOFF BY CLEANUP” sessions, which could have been terminated by reaching/exceeding the CONNECT_TIME or IDLE_TIME profiles limits.
- AC-21: One report listing logons recorded where the account used does not match the OS User Account. However, this will

only work if the network/OS account name is the same as the DB account.

CNTL NO.	CONTROL NAME	PRIORITY	INITIAL CONTROL BASELINES		
AC-10	Concurrent Session Control	P3	Not Selected	Not Selected	AC-10
AC-11	Session Lock	P3	Not Selected	AC-11 (1)	AC-11 (1)
AC-14	Permitted Actions without Identification or Authentication	P3	AC-14	AC-14	AC-14
AC-22	Publicly Accessible Content	P3	AC-22	AC-22	AC-22

Of the four P3 controls, I envision the controls as being monitored in the following manner:

- AC-10: One report listing the configuration of each granted profile listing session limit for SESSIONS_PER_USER and CPU_PER_SESSION.
- AC-11: One report listing session locking occurrences.
- AC-14: This would typically be identified by reviewing Oracle Security Alerts, which identify vulnerabilities on a periodic basis. This should be a collaborative effort between the DBA and Security monitoring team. This requires manual documentation of review.
- AC-22: One report listing any non-standard grants to the PUBLIC group.

CNTL NO.	CONTROL NAME	PRIORITY	INITIAL CONTROL BASELINES		
AC-9	Previous Logon (Access) Notification	P0	Not Selected	Not Selected	Not Selected
AC-16	Security Attributes	P0	Not Selected	Not Selected	Not Selected
AC-23	Data Mining Protection	P0	Not Selected	Not Selected	Not Selected
AC-24	Access Control Decisions	P0	Not Selected	Not Selected	Not Selected
AC-25	Reference Monitor	P0	Not Selected	Not Selected	Not Selected

Of the five P0 controls, I believe that each of these require documentation and education of the database administration team, developers and the security teams, agreeing on how these will be mitigated.

Using the examples provided for the Access Control family, the same process needs to be performed for all of the remaining families listed in the Security and Control Identifiers table.

Frank Pound is currently a data modeler/analyst/architect at the central Bank of Canada with over 15 years of experience providing Oracle database/technology support within software support and services, pharmaceuticals, automotive and public sectors. Since 2006, Frank's interests/focus has turned toward the creation and implementation of management control processes to ease the reporting responsibility for database teams, reduce risk, and increase security and resilience. Recently, he has added Oracle licensing as an area of interest. In 2015 and 2016 Frank presented papers on audit, database license process creation, and implementation and reporting at the COLLABORATE conferences.

Contact

SELECT Journal Call for Content

Like what you see in *SELECT* but wish there was more information on a specific topic? Let us know!

Email **select@ioug.org** with your ideas or feedback.

Who Are the Editors?

By Jonathan Gennick

Professional-level books are created by teams, and authors benefit by knowing who the players are and when those players become involved, and why. Everyone's an editor in publishing, and, sometimes, publishers don't manage the handoffs very well. Everyone's an editor, but with a different role to play.

Acquisition Editors

I'm an *acquisition editor.* Sometimes you'll hear the term *acquiring editor* or *commissioning editor.* We are the ones responsible for creating book deals and getting new projects off the ground. We look for people to write, and work in partnership with aspiring authors to plan and begin new books.

Acquisition editors travel to events and conferences, meet people, talk to people and try to stay abreast of trends and where their industry is heading. We do all these things so that we can (hopefully!) publish books that get read and that make a difference. Few things fire my blood like publishing a good author on a trending topic, knowing the content will help readers improve their skills and more easily provide for themselves and their families. We love books, and we love books that get read.

Developmental Editors

Developmental editors are the *substantive editors* who work alongside an author as a book is being written. Developmental editors read each chapter and provide feedback on such things as the overall structure of a chapter and the use of headings, on the sequencing of content, on whether topics are being addressed at the correct level of depth. Developmental editors look at pacing, the use of elements such as figures, lists and tables. Always the focus is on improving the clarity and quality of the content to make it accessible.

Acquisition and development are sometimes combined into one person. Playing both roles is a difficult thing, because the roles each require somewhat opposite skillsets. Acquisitions editing is outward-facing and is about short intervals and lots of task-switching, whereas developmental editors must focus for long periods on a single chapter, with minimal task-switching. Many, if not most, publishing houses split the roles.

Technical Editors

Technical editors are sometimes termed as technical reviewers. They are brought in to help ensure the technical content of a book. It's not even remotely possible for acquisition and developmental editors to know the deep details of each of the wide range of technology topics they are involved in. Technical editors provide the per-book expertise that is deep and specific to an author's topic.

Technical editors help an author by pointing out possible omissions in coverage. (Knowing what is missing that should be included requires domain knowledge). Technical editors also help by suggesting improvements that lead to increased clarity. These can be additional code examples, figures, depth of content, removal of distracting asides and off-topic content. Technical editors do watch for mistakes, but it is always an author's responsibility to ensure correctness.

Coordinating Editors

Someone needs to run the show. At Apress, the job of managing the day-to-day aspects of a book falls to a *coordinating editor*. These are the editors whose job it is to maintain a book's schedule, to manage the flow of chapters among all the other editors and players involved, and to keep track of mundane but critical items such as figure art, cover art and front-matter.

Every published Apress author has had the experience of a coordinating editor calling them about a late chapter or a missed deadline. But what you may not know is that the editors get the same treatment. Coordinating editors tell us what to do as well, and we are also given deadlines. And while deadlines may sometimes be annoying, I am ultimately grateful to our coordinating editors for helping ensure that I get everything done that I need to get done in order to hit my publishing targets.

Copyeditors

Copyeditors read a finished draft of a book and make edits involving grammar, punctuation and spelling.

One of the most important roles of a copyeditor is to bring a text into alignment with a publisher's preferred style. Doing so can include such tasks as ensuring that heading words are initially capitalized, that abbreviations are not used until after they've been spelled out. House style can also include rearranging of sentences, such as, for example, to put modifiers in their proper place.

Production Editors

Production editors enter the maelstrom when a book is "done" and is ready to be published. Production editors at Apress arrange for pages to be laid out, for the various electronic formats — such as for Amazon's Kindle Reader — to be generated. It is a production editor who assigns the copyeditor and arranges for an index to be created. Production editors also perform quality-assurance checks on

(Continued on page 34)

DBA 201

Testing your DB Upgrades & Balancing Risk

By Gary Gordhamer; Simon Pane, Editor

Database upgrades are often a pain. They are impactful, scary, hard to discuss and inevitable.

Just as you finish one upgrade, patch or application change, you will begin discussing the next one. This article will not be a definitive plan for how to test, or exact guidance on what you need to test, but rather guidance on how to think about database testing with respect to DB upgrades.

Testing is all about mitigating risk. Our goal as technical professionals is to reduce the risk of change affecting our business negatively. If the change implications are positive then the business does better or is safer, but if the change effects are negative, that can be a career-limiting move. Truthfully, the results are often based not just on the technical testing, but also how the testing is portrayed and how the conversation between IT and the business happens. When we say that an upgrade is well tested, we are basically saying the risk of doing the upgrade is low.

To help with the framing of risk, let's discuss what goes into a change. First is the technical pieces from Oracle; the second is the application that uses the database; and finally, there is how the business uses the application.

Patches

What can you expect when you patch a database? We need to start with understanding what upgrades or patches are. I'm going to bucket database upgrades into four categories:

1. **One-off patch:** A fix that is intended to resolve one issue, or maybe a small number of issues. Realistically, the patch represents a single area of functionality being touched.
2. **Bundle patch:** This is a combination of one-off patches. This includes most quarterly "Patch Set Updates" (PSUs), merge patches and many other patches. This is where multiple (many) one-off fixes are combined to make it easier to apply them, due to dependencies, or to help make sure high priority issues get fixed.
3. **Minor version upgrades, a "Dot patch or release":** Upgrades generally represented by a third or fourth position numerical change in the product version, such as 11.2.0.3 to 11.2.0.4 or 12.1.0.1 to 12.1.0.2 upgrades. In general, these contain many fixes, and might contain some minor functionality differences in the product (including query optimization and performance). Occasionally, new functional features are released in these versions, but not often. Most third-party applications certify to databases at this level.
4. **Major version upgrades:** These are the upgrades that affect the main product version number or sometimes the second digit, such as 11.1.0.7 to 11.2.0.3 or 12.1.0.2 to 12.2.0.1, or 11.2.0.4 to 12.1.0.2. These represent the most change to the product, and almost always will include functionality changes and new features.

From this list, it is important to note that any patch can change any part of the database. Just because something is a one-off patch, does not mean that it does not change the query optimizer. , Likewise, just because something is a bundle patch doesn't mean that it definitely will touch the query optimizer. It is critical that you become familiar with how to read patch notes, and get a feel for what will or will not be affected. Your mental model of change needs to balance the size of the change and the way the change is packaged by Oracle.

Applications

The other major variable in our equation are databases, which are used by applications. Applications can either be incredibly complex or simple. All good DBAs will begin to understand every application they support (even if they support hundreds or thousands of them). This does not mean knowing every technical nook and cranny of the application (though it could). Generally, I think of applications in a few categories, much like the DB upgrades:

- **Light applications** use just tables and indexes in the database. There are many of these, and they mainly see the DB as a guaranteed data store. They may rely heavily or lightly on the multi-user concurrency of the DB engine.
- **Medium applications** use some of the features of the DB that are specific. They may use Java in the database, XML data types, PL/SQL packages, etc. They are still mainly just table and index consumers, but have some code in the database for performance, unique application functionality, or to handle large volumes of users or data.
- **Heavy applications** use just about every feature of the database and more. These are the most complex systems such as Oracle E-Business suite.

In these three groups, I'm including both commercial off-the-shelf applications (COTS) and in-house written custom applications. One minor difference here is where you go for support. For COTS applications, you reach out to an external vendor for certification and support. For in-house built applications, you may have your own team (or you may not, based on the age of the application). Again, these are things you will have to add to your own mental model.

The second factor to the application is how the business uses the application. This is somewhat irrespective of the weight of the application. For example, one company might have a simple light mobile application that every sales person uses to place every order; not complex, but very critical. On the other hand, a very heavy ERP system might only be used for financial reporting, which would be critical only a few days a month or a few weeks a year. I'm not going to provide much guidance here since this is all unique to your company. You might be a company of 100 people, and a 10-person sales team is very important. Or you may be a company of 100,000 users and apps with thousands of users, which may or may not be as important.

Overall, this process is entirely up to you as the DBA to decide in conjunction with your business partners. Your company may even have a formal group such as a change advisory board (CAB) that helps evaluate the risk. How can we use this information? For starters, let's apply a simple mathematical formula to some of these items and see how it feels:

P = DB Patch or Upgrade (1 = one-off, 2 = bundle, 3 = minor, 4 = major)

W = Application Weight (1 = light, 2 = medium, 3 = heavy)

C = Business Criticality (1 = low, 5 = super critical)

R = Risk

P*W*C = R

Let's walk through a simple example:

We need to apply a one-off patch to the Oracle Cluster Ready Services (CRS) software to prevent RAC node evictions. The application is a mobile sales application used in the U.S. that is in-house built, and has about 500 users. We will call the application "medium" and the business criticality as a 3 (right in the middle).

1 * 2 * 3 = 6

Ultimately, what this means is that the formula has a range of values for results from 1 to 60, an even split would say 1-20 is low risk, 21-40 is medium risk, and 41-60 is high risk. Though it's really up to you to decide. Again, this is not an absolute, but a way to help describe or think about risk.

The important part is not just to do the math, but rather how you would be able to describe it to management and users. Now let's try a non-mathematical way to describe this. I'll preface each line with the risk level based on that logical point:

- Low: Assume the risk is low and try to prove it is not (you can start at either end).
- Low: This is a one off patch, it only changes a small item or single feature.
- Low: This patch is for a known bug; a node eviction would be very impactful (the risk of not applying may be higher than the risk of applying).
- Low: This patch affects the CRS, and the application code never talks directly to the CRS.
- Low: The application is used during U.S. daytime mostly, so getting an outage after U.S. hours should be easy, and there should be ample time to remediate any problems.

Now you can see that through this method we logically came up with the same answer and no math.

Let's do one more quick example. We want to apply the July 2016 PSU to a stand-alone database for an application that is based on XML DB and Oracle Application Express (APEX), used by everyone in a global organization to enter their time cards. From a quick math formula, we get 16 (2*2*4), which would be low (though this is probably not true). Walk through the items again:

- Low: A bundle of security patches, meant to fix known issues.
- Low: If there are small functional changes to the DB, they are to remove undocumented or unintentional functionality (our application should not be based on these).
- Medium: This patch affects PL/SQL and database engine components that our application is directly related to.
- Medium: The application is used globally and there are many users that would be upset if they could not track their time.
- Medium: Since the application is global, it will be hard to find a good timeslot that allows for proper testing, and back out time without interruption to users.

The straight math does not provide as clear a picture. Here we have also added the concept of cost around downtime. As you mature in your mental model on upgrades and patches, you will need to add in multiple aspects about how the business uses the application. For example, if the application is used to manufacture product, then downtime could mean lost manufacturing and directly cost the business either revenue, or lost opportunity. Hopefully, you have a solid understanding of how to describe the risk of database patching.

Testing to mitigate risk

Now that you understand the risk, it is now time to mitigate it — by testing.

Applications are expected to take specific inputs (data), do actions and then probably do some form of output (reports, actions, processes). Also note that there are many roles in testing; from a DBA standpoint, you can probably only be responsible for showing that the database is healthy, where functional users or developers are probably need to test that the application still functions as intended. Also, I'm focusing on the functional aspects of the application and testing, and will not get into performance testing. Let's introduce some generalized ideas:

- **Full functional testing,** walking through all features of the application, verifying every data input, every process, and every output.
- **Some functional testing,** testing the critical functions. For example, on a ticket system this might be create a ticket, close a ticket, route a ticket. For an order system, this might be to make an order, approve the order. We are not testing every possible feature.
- **Minimal testing,** just opening the application, maybe query a record or two.
- **No testing,** startup up the application and make sure login works.

In this list, I focused on the application side of the testing, or showing that the database did not break the application. I also generalized on the term "functional testing;" within your organization, there may be many terms used to describe testing the applications ability to work and many teams involved in this process.

You should always do some basic database health checks during every patch or upgrade. In fact, you may also want to proactively perform some database health checks weekly or monthly, since many things can happen outside of DB patching.

We have discussed risk in a mathematical, logical manner. Now let's show it more visually and integrate testing into the discussion.

DB Change

Major Upgrade				
Minor Version				
Bundle Patch				
One off Patch				
	Light	Medium	Heavy	**Application**

With this chart, we now have three variables: DB change, application type, and testing methodology (for which we have added in the color coding). We could say:

- Green – No testing or minimal testing.
- Yellow – Minimal testing or some functional testing.
- Red – Some functional testing or full functional testing.

Again, you will have to come up with your own language for this relationship that matches your needs. In fact, you could have a different set of rules based on the application, data classification or many other factors.

Rewards

The last part of any conversation on risk is the relation to reward. Again, I'm going to simplify rewards for DB patching into a few categories:

- **Fixing bugs, or removing unintended behavior.** This is the sweet spot for almost all upgrades. All one-off patches, bundle patches, security patches and most minor version updates are included this area. Trying to keep a system running smoothly, closing out security holes and maintaining a system to a supported level.
- **Introducing new features,** such as In-Memory processing, multitenant option, PL/SQL performance enhancements, new data types and more, can be very compelling depending on your goals.
- **Introducing new bugs.** Although this is not an intended reason to upgrade, I want to at least acknowledge that this can happen. In most cases, this is the risk we are concerned about and, generally, where most push back will come from.
- **Maintaining vendor support.** Either at the application level or the database level, often patching is necessary to ensure that the system remains supported by Oracle and other application vendors.

Any patch could include all four of these items or any one. Your goal should be to frame the conversation around any given upgrade to focus on the critical components. For example, if you are applying a PSU patch to close security holes, your reward is closing the security holes. The PSU could possibly contain a new feature or a new bug, but those are not your rewards in this specific request.

Why do we care about the reward? The bigger the reward, maybe the bigger the risk we are willing to accept. Say we have a critical financial application for our company, and a patch comes out for a known security bug that would allow someone to gain access to the database without a password. The risk for this patch might be medium, and would suggest some functional testing. Due to the risk level / reward (remove the possible attack vector), we do only minimal testing and rush the patch into production. In this case, the high reward value offsets the medium risk level.

Perhaps you are beginning to notice a pattern: There is no specific formula that tells you what you need to test. As a database professional, you have to come up with your own internal mental model that represents many factors, from the technical and business aspects. Testing is not the removal of issues or the removal of risk; it is just a safeguard or way to reduce the risk. As a professional DBA, you will be looked to as the expert in many areas, so being able to speak well and communicate clearly and in a confident manner is critical. This all seems very non-technical in nature. Let me leave you with some technical items to satisfy your itch. These are all Oracle-provided scripts and technical notes on things you can check in your database to show it is healthy. Again, this does not guarantee that the application that uses the database is healthy:

- FAQ: Database Upgrade and Migration (Doc ID 1352987.1)
- ORAchk - Health Checks for the Oracle Stack (Doc ID 1268927.2)
- Script to Collect DB Upgrade/Migrate Diagnostic Information (dbupgdiag.sql) (Doc ID 556610.1)
- Master Note: Troubleshooting Oracle Data Dictionary (Doc ID 1506140.1)
- hcheck.sql Script to Check for Known Problems in Oracle8*i*, Oracle9*i*, Oracle10*g*, Oracle 11*g* and Oracle 12*c* (Doc ID 136697.1)
- Best Practices for running catalog, catproc and utlrp script (Doc ID 863312.1)
- Debug and Validate Invalid Objects (Doc ID 300056.1)
- Where Can I Find the Parallel Version of Utlrp.sql? (Doc ID 230136.1)
- Overview of Refreshing Materialized Views (Doc ID 549874.1)
- How to Monitor the Progress of a Materialized View Refresh (MVIEW) (Doc ID 258021.1)
- Script to Check the Status or State of the JVM within the Database (Doc ID 456949.1) S

Gary Gordhamer is a Principal Technologist at GE Power, having worked with Oracle e-Business suite, middleware, database and related technologies for the past 24 years. His professional background covers many industries including healthcare, manufacturing, utilities and government. He is an active member of IOUG, currently serving as the *SELECT Journal* executive editor and COLLABORATE 17 IOUG conference chair. Gary is an Oracle ACE and frequent presenter at OpenWorld and COLLABORATE.

Contact

Platform: Technologies that Enable Change

(Continued from page 9)

specific information through special interest groups, and by facilitating knowledge transfer through networking among the IOUG membership.

Most technical professionals learn best by doing, which is why IOUG dedicates the first day of the COLLABORATE conference to pre-conference workshops where attendees can get hands-on experience with Oracle products. In 2016, four out of six pre-conference workshops involved Oracle Database 12*c* and 2017 promises not to disappoint. Register early and take advantage of these highly attended workshops.

Like any other release, Oracle Database 12*c* has advantages and disadvantages — look for upcoming *SELECT* articles with more on new features. The landscape of IT is changing and with it, the technologies that enable that change continue to evolve.

Seth Miller has been working with Oracle technologies since 2005 and specializes in database administration and solutions architecture and integration. He has a passion for Oracle and spends much of his time contributing to the Oracle community, including currently serving as IOUG's director of communications. He currently holds the titles of Oracle ACE, Oracle Certified Professional and Certified Oracle University Instructor. He actively writes about various Oracle and technology related topics and is a published author. Seth presents on a regular basis at local and national events and conferences, including Oracle OpenWorld and COLLABORATE.

IOUG Press Corner: Who Are the Editors?

(Continued from page 29)

book metadata to help ensure that all needed descriptions and cover images are in place before a book goes live in the feeds that are sent to the retail sector.

Amazing Access

Aspiring authors in the technology world are amazingly lucky in their easy access to acquisition editors. Do you present at conferences? Present to a packed room, and you may find one or more acquisition editors seeking you out about book deals, possibly before you've dismounted the podium!

Take advantage of the access that you have. IOUG is interested in building a strong line of IOUG Press titles. We at Apress love to publish good books. What better way to become published than under the auspices of the Independent Oracle User Group as an IOUG Press author? Contact Jonathan with book ideas at JonathanGennick@Apress.com

Jonathan Gennick is the Apress editor responsible for the IOUG Press, a cooperative publishing arrangement between Apress and IOUG with the aim of providing the very best in content for IOUG members and other Oracle professionals. Learn more at www.ioug.org/iougpress. Contact Jonathan with book ideas at JonathanGennick@Apress.com.