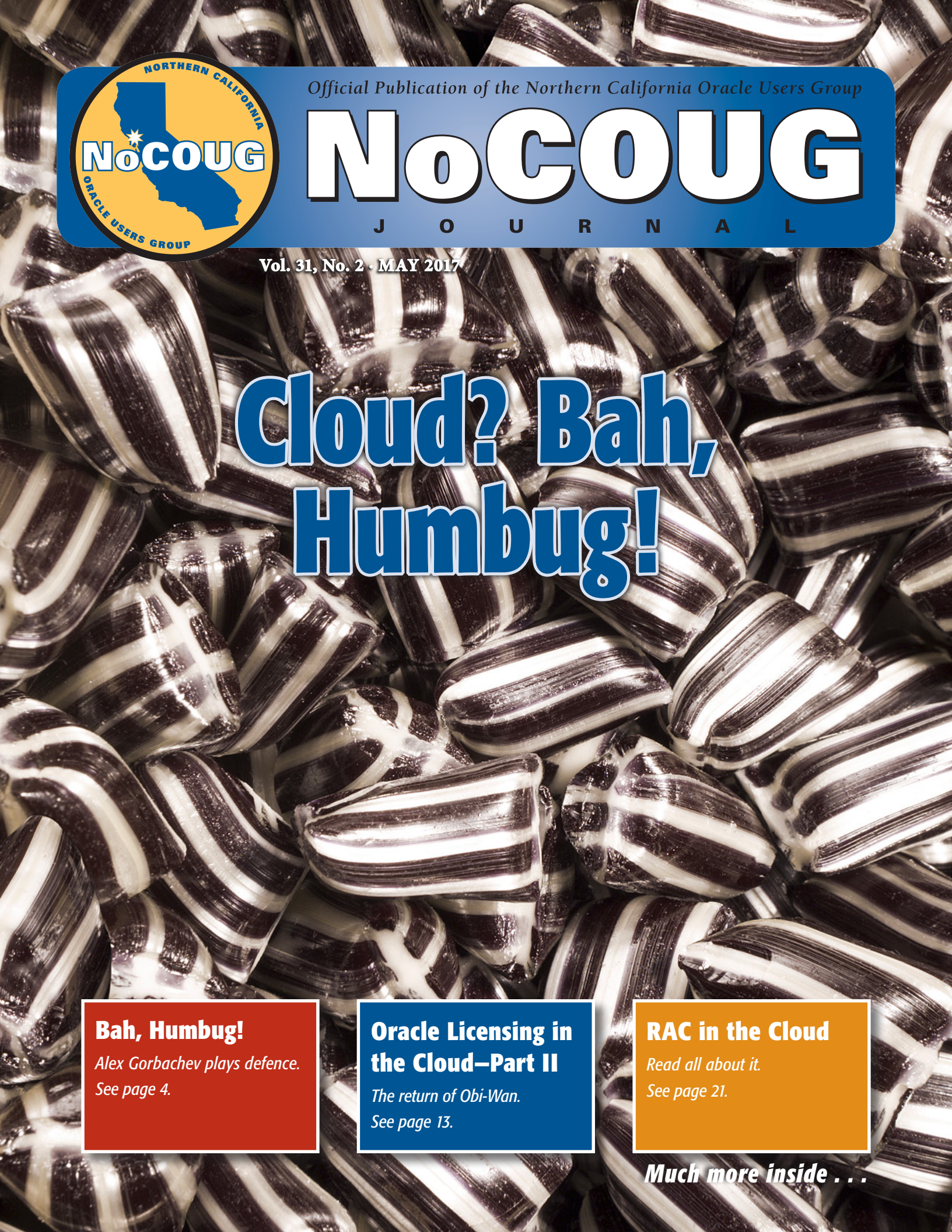


Official Publication of the Northern California Oracle Users Group

NoCOUG JOURNAL

Vol. 31, No. 2 · MAY 2017

Cloud? Bah, Humbug!

Bah, Humbug!
Alex Gorbachev plays defence.
See page 4.

Oracle Licensing in the Cloud—Part II
The return of Obi-Wan.
See page 13.

RAC in the Cloud
Read all about it.
See page 21.

Much more inside . . .

The New Phoenix by Axxana

For Oracle Databases and Applications

Zero Data Loss at Unlimited Distances

Synchronous Protection at Maximum Performance

- Low Cost Replication Lines
- Shortest Recovery Time
- Full Consistency Across Multiple Databases

AXXANA

BUILT TO LAST

www.axxana.com

ORACLE Gold Partner

ORACLE BIG DATA APPLIANCE READY

ORACLE DATABASE READY

ORACLE EXADATA READY

ORACLE LINUX READY

Professionals at Work

First there are the IT professionals who write for the *Journal*. A very special mention goes to Brian Hitchcock, who has written dozens of book reviews over a 12-year period. The professional pictures on the front cover are supplied by Photos.com.

Next, the *Journal* is professionally copyedited and proofread by veteran copy-editor Karen Mead of Creative Solutions. Karen polishes phrasing and calls out misused words (such as "reminiscences" instead of "reminisces"). She dots every i, crosses every t, checks every quote, and verifies every URL.

Then, the *Journal* is expertly designed by graphics duo Kenneth Lockerbie and Richard Repas of San Francisco-based Giraffex.

And, finally, David Gonzalez at Layton Printing Services deftly brings the *Journal* to life on an offset printer.

This is the 122nd issue of the *NoCOUG Journal*. Enjoy! ▲

—*NoCOUG Journal* Editor

Table of Contents

ADVERTISERS

Publication Notices and Submission Format

The *NoCOUG Journal* is published four times a year by the Northern California Oracle Users Group (NoCOUG) approximately two weeks prior to the quarterly educational conferences.

Please send your questions, feedback, and submissions to the *NoCOUG Journal* editor at **journal@nocoug.org**.

The submission deadline for each issue is eight weeks prior to the quarterly conference. Article submissions should be made in Microsoft Word format via email.

2017 NoCOUG Board

ADVERTISING RATES

The *NoCOUG Journal* is published quarterly.

Size	Per Issue	Per Year
Quarter Page	$125	$400
Half Page	$250	$800
Full Page	$500	$1,600
Inside Cover	$750	$2,400

Personnel recruitment ads are not accepted.

journal@nocoug.org

Cloud? Bah, Humbug!

by Alex Gorbachev

Alex Gorbachev

Bah, humbug! It's 1970s-style time-sharing all over again. The IT fashion industry has come full circle. What's the difference between Oracle Cloud and GoDaddy anyway?

Well, I was personally still walking under the table in the 1970s—so, perhaps, I shouldn't quite comment on the agility of a time-sharing approach—but I strongly suspect it's a few orders of magnitude different. I don't want to bore you by saying something smart like "everything new is well-forgotten old" or talking about how resource-sharing evolved through various things like host virtualization to the modern clouds with the latest fashion of serverless compute (everyone wants to put "serverless" on their product today like everyone wanted big data and cloud a couple of years ago). But I'll tell you why people want cloud these days . . .

Modern cloud is all about agility, scale, and commodity. Cloud delivery models are optimized to be a great conductor of value from product creators to consumers in this age of such a crazy-high pace of innovation. When you are creating a digital product for broad adoption, you have to focus on getting this product in the hands of customers as quickly and as easily as possible, without any barriers. If you don't do it quickly enough, your competitors will—and if they don't have a comparable product today, they will tomorrow . . . and it will likely be even better than yours.

Did I mention the high pace of innovation? The pace of innovation in all aspects of human life is increasing, and that's even truer for digital—but then everything is going digital these days, anyway . . . or going extinct. Digital adoption today is growing somewhat exponentially along with the pace of innovation. Put one and one together and it makes five. When your innovation is adoptable broadly (and it should be or it's not worth it), you will quickly end up facing fierce competition; this is what comes to all large-scale markets today with the abundance of capital to invest and the high pace of innovation to absorb such investments. This also means that whatever unique intellectual property advantage you have does not last long, so you have to be constantly innovating to stay competitive. There is no "maintenance" mode anymore; constant change is the new normal.

So . . . unless you are in the business of building niche solutions (and there are very few reasons to do so), you need to have agility, scale, and commodity—you need cloud.

Bah, humbug! What's the difference between Oracle Cloud and AWS (or Azure), except that the fine print has more words than the Old and New Testaments combined?

I'm not as familiar with the SaaS offerings of Oracle, so I'll only focus on IaaS and PaaS offerings, where there are a few main differences (and fine print isn't one of them, I think). It would go like this:

- Today Oracle is several years behind AWS, Azure, and GCP in terms of the capabilities and maturity of the platform.
- Oracle Cloud is a pretty closed platform, and it seems that Oracle focuses on direct large-enterprise adoption. This is either by choice or through failing to build an acceptable user experience that allows customers to try services and evaluate them properly before the purchase. It may just be by choice, because Oracle believes in sales cycles way above software engineering and operations teams when they sell their cloud services. Time will show if they are right.
- Oracle has the advantage of being in charge of the licensing scheme for the software that underpins many enterprise applications deployed today and makes licensing Oracle software more difficult on non-Oracle clouds.

Cloud is also like retail and e-commerce: it's a low-margin business operating at a huge scale. Transitioning a business model of high-margin proprietary software (that's Oracle) into a low-margin business is really hard, and Oracle is between a rock and a hard place. They can't afford to cannibalize their existing revenue and margins, being a publicly traded company living from quarter to quarter while trying to transition to a low-margin commodity cloud business. Typical Innovator's Dilemma.

Bah, humbug! It's a synonym for "loss of visibility," "loss of control," and "blind trust."

Ha-ha-ha-ha-ha. My experience working with complex on-premise and cloud infrastructure tells me that I always have better visibility of what's going on in the cloud—it's a matter of using platform API, which is always faster (if you know what you are doing) than chasing your storage administrators trying to figure out the disk layout of your datafiles in a complicated, multi-layered enterprise storage subsystem.

Same picture with the "control." When you need to add storage in the cloud—it's an API call (or a few clicks in the web UI, if you have to). Compare that with a formal request to your storage team when a human needs to handle it (which will be slow and probably with a healthy dose of error)—and often dependent on provisioning more physical capacity, which is always a struggle to justify.

"Trust" is an interesting thing in the IT industry—and it's amazing how long the industry has been going along by just hiring trusted users and giving them superpowers, hoping they will not do harm (Snowden, anyone?). At Pythian, we have identified this trust problem a long time ago, as we were a remote vendor before almost anyone else was doing it and had to solve the problem of trust. Luckily, there are many patterns applicable from physical life that can be implemented more efficiently in the digital world (like security cameras in banks and casinos, and bank vaults that open only with two keys) to unique digital mechanisms (such as encryption). Of course, it requires careful design, planning, and implementation, but it's much easier to justify such an investment at a larger scale—the economy of scale works an order of magnitude better in the digital world.

Bah, humbug! It's for small businesses only.

Cloud is for businesses that have demands very similar to many other customers: those that really need to compute (the same as many thousands of customers) should adopt IaaS; those that really need databases and application stacks (the same as many thousands of customers) should adopt PaaS; those that really need email and CRM (the same as many thousands of customers) should adopt SaaS. Small businesses have no choice but to quickly realize that many of their needs don't have to be unique and to carefully plan their investments. Big, old businesses have an illusion of unique needs, as it has been for ages, while the reality is that they would probably be just fine with mainstream functionality. But we all know that change is hard and the status quo is comfortable and easy.

Of course, there are always areas where it makes sense to build something special, and this is how you differentiate yourself from the competition. It's a matter of making the right choice as to what those areas are, and large businesses are just making too many bets where they want to be unique.

Bah, humbug! Read my lips: Global outages caused by human error. Is that the new normal?

Yes. That is a challenge, because managing such a large-scale commodity infrastructure requires ubiquitous automation that can hugely amplify the impact of the faults whether it's by human or by code (which is written by humans anyway . . . or at least that's how it is until AI eats the world). There have been large-scale failures of electrical grids, stock markets, and so on. You solve it by engineering systems well and embedding layers of safety measures. Of course, every now and then they all fail (Swiss cheese anyone?), but think about this: who is more likely to make a deadly mistake, a sleep-deprived network administrator upgrading some critical device in a company that has to avoid extensive test infrastructure to cut costs or a top-notch team of network administrators supported by millions spent on reliability of operations? Scale solves many things and allows companies to make investments that are not feasible until such scale is reached.

Bah, humbug! Now Big Brother can spy on everyone. Read my lips: "National Security Letter." The EU won't tolerate it. So why should we?

Well, not that I'm enough of a legal expert to give advice here—and yes, believers in information freedom would find these laws to be unacceptable—but if a U.S.-based public cloud vendor with data centers in the U.S. have to respect NSL, then so do you (providing you are governed by U.S. law). But hey—there are cloud data centers in Canada too. One more good reason to move up north!

"The change is coming and as with any significant change, you have three options: 1) ignore the change hoping it will go away; 2) actively resist the change because you don't agree; or 3) embrace the change—adapt and maximize your opportunities during the transition. Make your choice."

Bah, humbug! Every self-respecting hacker has the cloud in their sights: "Hard on the outside, soft on the inside." Just wait until the fox gets into the chicken coop with the root password. Probably "Welcome1" or something idiotic like that.

Idiotic passwords in the cloud are idiotic passwords on-premise. If you want to be stupid when you're configuring security aspects of your applications and infrastructure, then nothing will help. The good news is that cloud vendors have already thought about good rules of thumb and configuration patterns that help in building secure systems, and they have embedded those good principles in code and in the default infrastructure configurations. And that also means that all the customers are now going to benefit without spending their own millions of dollars to do the same. Did I mention that economy of scale works really well in the digital world?

Bah, humbug! You can't replace me. The world will always need DBAs. ALWAYS.

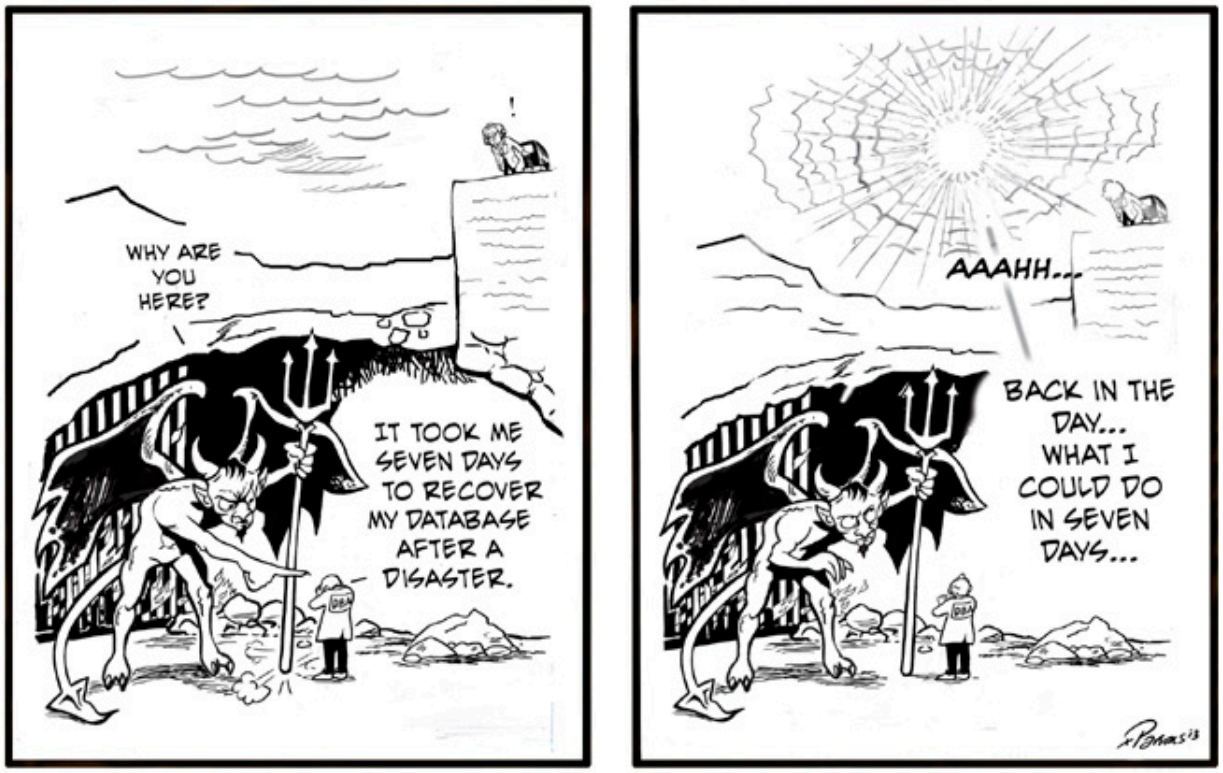
The change is coming and as with any significant change, you have three options: 1) ignore the change hoping it will go away; 2) actively resist the change because you don't agree; or 3) embrace the change—adapt and maximize your opportunities during the transition. Make your choice.

I would like to refer the reader to a paper written by my colleague Warner Chaves with the help of Vladimir Stoyak, "The Evolving Role of DBAs in the Cloud" (**http://resources.pythian.com/evolving-role-of-dbas-in-the-cloud-whitepaper**), where Warner discusses how to fit into this new landscape of managing databases in the public cloud. In addition, I recommend listening to the first episode of *Datascape Podcast*: "Discussing the Future of the DBA Career" (**www.pythian.com/blog/datascape-podcast-episode-1-discussing-the-future-of-the-dba-career/**). ▲

Alex Gorbachev, chief digital officer of Pythian, is a highly sought-after speaker and a well-recognized and respected figure in both the Oracle and Big Data communities. Alex holds the prestigious Oracle ACE Director designation from Oracle Corporation and was recently named Big Data Champion by Cloudera. His deep technological expertise and vision have helped put Pythian at the forefront of emerging data markets. Today, the CDO office is an incubator of new services and technologies. Most recently, Alex built Pythian's Big Data Engineering services team and established a Data Science practice.

Dr. DR

by Rich Parsons



Dr. DR is brought to you by Axxana.

TriCore Solutions: DBA Pro Service

DBA PRO BENEFITS

- *Cost-effective and flexible extension of your IT team*
- *Proactive database maintenance and quick resolution of problems by Oracle experts*
- *Increased database uptime*
- *Improved database performance*
- *Constant database monitoring with Database Rx*
- *Onsite and offsite flexibility*
- *Reliable support from a stable team of DBAs familiar with your databases*

CUSTOMIZABLE SERVICE PLANS FOR ORACLE SYSTEMS

Keeping your Oracle database systems highly available takes knowledge, skill, and experience. It also takes knowing that each environment is different. From large companies that need additional DBA support and specialized expertise to small companies that don't require a full-time onsite DBA, flexibility is the key. That's why TriCore Solutions offers a flexible service called DBA Pro. With DBA Pro, we work with you to configure a program that best suits your needs and helps you deal with any Oracle issues that arise. You receive cost-effective basic services for development systems and more comprehensive plans for production and mission-critical Oracle systems.

DBA Pro's mix and match service components

Access to experienced senior Oracle expertise when you need it

We work as an extension of your team to set up and manage your Oracle databases to maintain reliability, scalability, and peak performance. When you become a DBA Pro client, you are assigned a primary and secondary TriCore Solutions DBA. They'll become intimately familiar with your systems. When you need us, just call our toll-free number or send email for assistance from an experienced DBA during regular business hours. If you need a fuller range of coverage with guaranteed response times, you may choose our 24 x 7 option.

24 x 7 availability with guaranteed response time

For managing mission-critical systems, no service is more valuable than being able to call on a team of experts to solve a database problem quickly and efficiently. You may call in an emergency request for help at any time, knowing your call will be answered by a TriCore Solutions DBA within a guaranteed response time.

Daily review and recommendations for database care

A TriCore Solutions DBA will perform a daily review of activity and alerts on your Oracle database. This aids in a proactive approach to managing your database systems. After each review, you receive personalized recommendations, comments, and action items via email. This information is stored in the Database Rx Performance Portal for future reference.

Monthly review and report

Looking at trends and focusing on performance, availability and stability are critical over time. Each month, a TriCore Solutions DBA will review activity and alerts on your Oracle database and prepare a comprehensive report for you.

Proactive maintenance

When you want TriCore Solutions to handle ongoing proactive maintenance, we can automatically access your database remotely and address issues directly — if the maintenance procedure is one you have pre-authorized us to perform. You can rest assured knowing your Oracle systems are in good hands.

Onsite and offsite flexibility

You may choose to have TriCore Solutions consultants work onsite so they can work closely with your own DBA staff, or you may bring us onsite only for specific projects. Or you may choose to save money on travel time and infrastructure setup by having work done remotely. With DBA Pro we provide the most appropriate service program for you.

TriCore solutions

Oracle SOA Suite 12c Handbook: Design and Deploy an Enterprise-wide Service Oriented Architecture

Book Notes by Brian Hitchcock

Details

Author: Lucas Jellema

ISBN-13: 978-0-07-182460-6

Date of Publication: March 21, 2014

Publisher: Oracle Press

Summary

I have been supporting Oracle Fusion Middleware (FMW)—which includes the SOA Suite—for several years, and I've worked on many SOA environments. I wanted to read this book to learn about all the components of the SOA Suite, most of which I've seen but a few that I haven't worked with.

This book assumes that the reader is a developer and provides many screenshots from JDeveloper. It is the many screenshots that make this book about twice as long as most Oracle Press books.

Introduction

We are told that Service-Oriented Architecture (SOA) is a major trend in IT. The major benefits of SOA are described, including the ever-popular adding real business value. We are also told that adopting SOA is a major challenge that will take considerable resources. To implement SOA you must have a SOA platform (queue Sales!), and you will be shocked to learn that Oracle SOA Suite 12c is the platform that will be explained in this book. The target audience is developers and, to some extent, administrators. I support SOA full time so I'm an administrator, not a developer, but I wanted to know a lot more about what I have been supporting.

While reading this book, I attended a NoCOUG conference where a fellow NoCOUG member informed me that SOA is a niche product. I don't have comprehensive metrics to debate whether SOA is a major trend or not, but I find many Oracle customers are using SOA.

The book is divided into five parts, starting with Part I, Setting the Stage, where we are introduced to the fictitious Saibot Airport that will be used as the project backdrop to discuss Oracle SOA Suite. Part II is Elementary Services, which is the first coverage of Service Bus and SOA composite applications. Part III, Composite Services, describes more complicated services, Part IV covers Asynchronous Services and Events, and Part V covers Processes, which include "human actors." This last section also covers processes used to develop and deliver services from test to production.

The SOA Suite covers a lot of material, and even though this book is about twice as long as most Oracle books I have read, there isn't enough room for all the relevant material. There is a website for this book where you can find more detailed instructions and full-color screenshots to help you with the various sections of the book.

Chapter 1: Saibot Airport Reaching for the Future

Here we learn all about Saibot Airport and its various business processes that need to be improved using SOA—and, of course, why the Oracle SOA Suite 12*c* was the best choice for the SOA platform. Among the many reasons to move to SOA are the need for a B2B service, a web portal, and a mobile application. The airport also wants to go paperless. Strictly as a side note, not directly tied to this book or SOA, but the phrase "go paperless" is easy to say and sounds good, but what does it really mean? Assuming the airport has been in business for a number of years, consider how many paper documents it may already have. Will the move to SOA include the resources needed to convert all the existing paper to digital? Or, will the airport decide to have two sets of information, the old paper information and the new digital information? How will these co-exist? This is outside the scope of this book, but I hope no one reading this will think SOA will help convert all existing non-digital resources.

Sections cover the business vision, strategy, and objectives, and the perceived need to "embrace change" to become more "agile." It seems that the airport wants to be more flexible. IT objectives are discussed, and the word "agile" comes up again along with the need to use open-source software—but don't get concerned, we are quickly told that due to bad experiences, the airport realizes that they should only use software that has a large commercial backing. That was close! Might have needed to get the Oracle sales team to come out and provide re-programming to some of the human actors. No further specifics are given regarding the pros and cons of open-source SOA versus proprietary, let alone any choices other than Oracle SOA Suite 12c.

There are also sections covering the Architecture to Enable the Future (wasn't that a Disneyland ride?), domains of data, Service-Oriented Architecture, event-driven architecture, airport processes and organization, and more. Clearly there is a lot

of material presented before we even begin to see the Oracle SOA Suite.

Chapter 2: Overview of Oracle Fusion Middleware and SOA Suite 12*c*

We start with a generic marketing pitch for the Oracle Technology Stack and all its virtues, including the Cloud and Oracle Fusion Middleware (FMW). The section on the history of Oracle Middleware is well worth reading. This explains why some Middleware components seem to do much the same thing: they came to Oracle through multiple acquisitions and the process of fully integrating them is not yet finished. In a more detailed section about Oracle Fusion Middleware we learn that Oracle itself is the biggest customer for this product since so many other Oracle products are built on top of it. I'm not sure I want to eat the FMW pudding that is described, but I will leave that up to you. WebLogic server or, more formally, the WebLogic Suite, is explained as the foundation of FMW. We then have an overview of SOA Suite 12c, which includes this definition: it is an orchestration and integration platform for receiving, processing, and answering messages. At first I didn't get much from this definition, but after reading the rest of this book, I understand. SOA really is about all the ways your business needs to do stuff with messages. It sounds too simple at first, but it can get a lot done.

There are sections covering the SOA Composite Application (SCA) Engine; Service Bus; adapters to connect with databases, etc.; Event Delivery Network (EDN); and several other SOA Suite components. This chapter ends with instructions on how to set up your own SOA Suite 12c installation. In the chapter summary we have another description of what SOA Suite does for us: it provides a framework for negotiating events between applications, offering a decoupled way of making different applications interact.

Chapter 3: Exposing SOAP and REST Services Using Service Bus

The airport has decided that all their services will be made available through Service Bus. This means all the apps will use standard protocols and message formats. If a service is not available, requests can be rerouted to another provider or results can be served from a cache. Service Bus is described as a stateless, synchronous request/response engine that plays the intermediary role between services and users. I had never heard of the VETRO pattern (Validate, Enrich, Transform, Route and Operate) but this is what Service Bus implements. Users of a service access the service at an endpoint where their request message is received by a proxy that will perform any needed conversions before handing over to the Pipeline, the main processing unit in Service Bus where routing is handled.

A good explanation of how Service Bus works is given by comparing it to the airport Information Desk. Next, we see all the steps needed to actually create a Service Bus service by creating a Temperature Conversion Service, which users could call on to convert Celsius to Fahrenheit. At this point, the text has lots of screenshots showing all the required steps in JDeveloper. I am not developing SOA applications so I didn't study all the screenshots, but for those that are, this level of detail would be very useful. The Temperature Conversion service is tested and extended, and there is more discussion of how to validate the input data to ensure proper results.

Chapter 4: Accessing Database and File System through Outbound Technology Adapters

Adapters allow a service to interact with databases, filesystems, and other enterprise resources. The Technology Adapters are described as well as the various database needs of the airport. A new service is created that will access airport data stored in a database. All the needed steps in JDeveloper are shown with screenshots. Testing this new service is covered. There are many details, such as how to handle requests that will access multiple tables, which is done with stored procedures.

The adapter for accessing a filesystem is covered next and is used in a service that will store all flight event information for the flights at the airport. All the steps to implement this are shown, as is testing and how to enhance the service.

Chapter 5: Introducing SOA Composite Applications

SOA Suite provides another way to create services by using SOA composite applications that use the Service Component Architecture (SCA) standard. Services that use SCA can use various service component engines such as Business Process Execution Language (BPEL) and can be easily integrated with external services. Overall, Service Bus is good for simpler services that require fast execution, and SCA is used for more complicated services that may have many external interactions and may take much longer to run.

The SCA standard is described. It is easier to deploy a bundle of components as a single SCA application rather than deploying all the components separately. It is interesting to learn that for all the virtues of SOA, the SCA standard does not support portability of the applications. This means a SCA that runs in Oracle SOA Suite will not run in Apache or IBM SCA containers. I wasn't expecting that, since so far everything has been about open systems and standards. There is a section covering the background on SCA with a further explanation of service components and composites.

Oracle SOA Suite has two run-time service engines, the previously covered Service Bus and the SCA engine that processes SOA composite applications. A detailed comparison of Service Bus and SCA is presented, and it becomes clear that they are very similar—but again, SCA is used for more complicated long-running services. The same Temperature Conversion service is redone but as an SCA. All the steps in JDeveloper are shown.

Chapter 6: One-Way Services and More Protocols

To this point, the services discussed have been synchronous, two-way services where a request was made and a reply was returned. This also means that the user of the service has to wait until the reply comes back. For many applications, a service does not need to reply and the user of such a service doesn't need to wait. SOA Suite can work with various messaging transports (JMS, MQ, UMS, and AQ) so that a service can hand over a message to an external source without waiting for a response. For the airport, every time an aircraft lands, a message is sent to a queue where it will be processed by a financial application to collect landing fees. This is used as an example of implementing a service using the JMS adapter. A more detailed example is a service that provides flight updates via email and chat using the UMS adapter.

While the details of how to set this up are presented, an interesting bit of history comes to light. Service Bus and SCA engine

are similar but not exactly the same because they were developed separately; Service Bus is from BEA Systems, which was acquired by Oracle. This also means that Service Bus and SCA engine will, over time, converge.

Also interesting is the concept of users not waiting for a response, which makes the app asynchronous, but at the same time, sending the user a reply that their message was received and will be processed later. I am not a lawyer, but I think this provides the user with a way of proving their request was submitted and not just sent off into the void.

Chapter 7: Composite Services with Service Bus

I was confused by this at first. I had seen Service Bus for simple, fast services and SCA, SOA composites for more complicated applications. So I thought I had already seen composite services, but I was wrong. Composite services are services made up of multiple services to build an application that can do more. We can include conditional steps and loops, and even execute parts of the message processing in parallel. This chapter looks at composite services using Service Bus, which means the resulting services are synchronous or one way, do not run for a long time, and are stateless.

This time the airport example is a service that accepts a flight schedule submitted by an airline. This requires planning a flight that requires multiple stages of processing. The detailed example shows setting up the basic application and adding validation and constraint checking as well as handling the errors that may result. In this discussion we are told that the decision as to when to use Service Bus or SCA is important—but, as we have seen earlier, this issue is not clearly defined and it is more art than science.

Chapter 8: Composite Services with SOA Composites Using the BPEL Component

In this chapter we see how to build complex applications using Business Process Execution Language (BPEL). Using BPEL makes it easier to perform flow logic, loops, and if-then conditional logic. The same flight schedule application is implemented again but this time using BPEL. An introduction to BPEL is included, as are many of the steps needed to complete the implementation. A BPEL program is called a "process" and typically consists of calls to services—specific BPEL activities like data manipulation, flow logic, and event handlers. BPEL is recommended for services that need to invoke many other services and need to be long-running. Specific differences between BPEL and Service Bus are described. Versioning is one specific difference that caught my attention, because I've seen it in the wild. You can deploy multiple versions of a SOA composite, all of which are running in parallel, while Service Bus does not support versioning.

Chapter 9: Coherence Integration to Reduce Load and Speed Up Response

So far, all the services we have seen have been stateless. They don't have memory; once they have executed, that's it. The first service we looked at was used for converting temperature between Celsius and Fahrenheit. Once that service was executed, the conversion was performed, and there was no memory of that execution. Now we learn about Coherence, which is yet another component of the Oracle Fusion Middleware (FMW) product. Coherence provides an in-memory grid that can store service results for reuse and can also be used as a cache to provide results to future executions of the same service. There is an adapter for Coherence so it can be integrated into SOA composites.

We see the steps needed to add Coherence to a composite, and how much faster it can be to retrieve a previous result versus re-running the same service. Also important is that results stored in Coherence are available to different SOA composites and not just to different instances of the same composite. I wonder how this is dealt with to prevent security issues. There is a discussion of how to set up a cache refresh strategy to detect stale results and make updates. Note that SOA composite results can be large if, for example, the service generates a document. Results can take up a lot of memory and Coherence can be configured to have the cache local to the same machine where SOA Suite is running or on remote servers.

Chapter 10: Embedding Custom Logic Using Java and Spring Component

For all the capabilities services have, there are times when you need something more specific. For these situations we have the Spring component, which supports custom Java classes and third-party Java libraries. An example shows how to create a service that uses a simple Java class to implement a simple greeting, a service version of Hello World. Another example is covered as the airport wants to use Twitter to send notifications of flight status updates, and this uses the Twitter4J library. Using a number of Spring components, we are shown how to set up a service that will convert a flight update message to a verbal announcement, convert that into an MP3 stream using the Google Text To Speech API, and make the MP3 file available. The website has the full details of this process.

Chapter 11: Business Rules to Capture Business Logic to Automate Decisions

We have seen services that had logic programmed into them. BPEL provides lots of support for what is called "implementation logic" that is very technical. There is also a need for business logic that is usually provided by business analysts that are not programmers. Their descriptions of the business logic will be much less technical. To support such logic, SOA Suite has the Decision Service component, which is also called the Business Rule service. This is used to implement a service that computes the airport charge for a flight, a complicated business process. As the business logic gets more complex, it can be hard to express all decisions as simple if-then rules. We learn about using a Decision Table that supports very complicated rules. In such a table, each column can be read as a single if-then rule, where each column represents a condition. One example would be a column for each of various times of day. Each time of day has a set of rules for a business process.

At this point, the author chooses to use—as an example of a complicated process at the airport—the CEO's process of choosing coffee at Starbucks. What is sad about this is that I have no problem believing that the CEO's most complex task each day is choosing coffee. And no, that doesn't make me sound cynical, not even a little bit!

Chapter 12: Asynchronous Services and Events

We have already seen services that don't need to supply a response. Here we see services that don't immediately provide a

complete response but just do what needs to be done and send a reply when the result is available. This means the request for service doesn't have to wait for the response, which would cause the request to lock up and waste time waiting. How to set up asynchronous services is covered. The example service will generate a request for proposal to suppliers of aircraft services. It may take a long time for a supplier to return such a request, so we don't want to wait for a response.

Chapter 13: Inbound Adapters—Polling Database, Consuming JMS, and Receiving Emails

The services that have been discussed so far have all started with a request message from an external user or consumer. Someone requests a temperature conversion, for example, and the service returns the requested result. These are called outbound operations, going out to a database perhaps. There are many situations where we need to have a service triggered due to an inbound message. When a new file is submitted from a vendor or a database record is updated, we need to have that external event start a new service request.

SOA Suite has inbound adapters to handle events such as these. This chapter covers inbound database, JMS, UMS, email, and chat adapters. For the database adapter, a detailed example is covered, and another example is shown for receiving emails.

Chapter 14: Using the Enterprise Scheduler Service

The Oracle FMW product offers the Enterprise Scheduler Service (ESS), which can be used to make asynchronous scheduled requests to services, including SOA composites. We are told that ESS is not suitable for high-frequency tasks. Job sets and schedules are explained, and several examples are worked.

Chapter 15: The Event Delivery Network

There have been many examples of how we can use SOA Suite to build applications that are decoupled. We have services calling to databases, for example, and responses are returned between systems and services that may be far away from each other. If we need to have a service called at a precise time, what do we use? If we have a service that requests bids from vendors and no vendor replies, what do we want to see happen? We could use Event Delivery Network (EDN) to set up a scheduled event that—after waiting for some time after the request for bids—would call other services if none were received.

We also have a discussion of Event-Driven Architecture (EDA), which could also be called Extremely Decoupled Architecture and is presented as both a complement to and a possible successor to SOA. In EDA, events are messages used to trigger services instead of a direct call to a service.

Events are published and anyone that subscribes to the event will be notified. If no one is subscribed, that's okay too. The example given for the airport is for handling flight status updates. When there is an update, all the subscribers are notified. More detail is shown for transactions, security, and scalability within EDN.

Chapter 16: Fast Data Processing with Oracle Event Processor

"Fast data" refers to streams of data that are continuously moving through a system, such as data from the IoT (Internet of Things). As we have more and more sensors that can connect to the Internet, we will have more cases where we must process fast data. Processing such data must be done as close to real time as possible. The SOA Suite has the Oracle Event Processor (OEP) to handle this. OEP can also be used to look for patterns in a data stream—for example, data coming from Twitter.

The specific example presented is that of managing a car park. Each time a car enters or leaves the lot, an event is created and OEP is used to process all the data to determine when the lot is full. Another example presented covers detecting credit card fraud in the airport.

Chapter 17: The Human Task Service

Now we address a part of any real-world business process that we really don't want to cover . . . humans! The author refers to this issue as the "ghost in the machine"—as if we humans are all just so much flotsam getting in the way of the robots. This makes me think of the Ghost in the Shell manga where my favorite non-humans were the Tachikoma . . . but I digress.

Humans, we are told, are needed to handle things that can't be fully automated. Things that require insight, negotiation skills and, for example, an understanding of abstract paintings! Seriously, we do need to have an interface to these annoyingly slow beings, and SOA Suite provides that with the Human Task Service, which accepts service request messages just like other services. This sets up the asynchronous interaction with the humans. This is covered in detail as we add human approval to the service that the airport uses to request proposals from vendors.

Chapter 18: Business Process Oriented BPEL

We have seen BPEL before, but now we see how it is used to build a service from a description of a business process. The specific example is a new staff member at the airport requesting a security clearance. This chapter wants us to see what it takes to go from high-level BPEL process to the detailed flows, logic, and human tasks needed for the final implementation.

Chapter 19: Business Process Management and Adaptive Case Management

Here we learn about modeling more complex business cases using Business Process Modeling Notation (BPMN) and Adaptive Case Management (ACM). BPMN was developed to provide a clear, precise notation for business processes. ACM is used to handle business processes where we have a cluster of business activities. The cluster is called a "case," and the activities are steps that will or won't happen when the case is executed. Examples of this are a taxpayer interacting with the tax authorities or someone booking a hotel.

Chapter 20: Monitoring for Insight into Business Process Execution

While all of these services are running we need to be able to monitor their progress and status. This chapter describes what is in BPEL and BPMN that will expose relevant data about specific instances to keep track of what is happening. We can use this to measure response time against the SLA rules in place. BPEL offers sensors that report on specific activities and can be monitored in the FMW EM Console. At the airport, this would be used to monitor the progress of security accreditation requests. Specific examples of how to set up the monitoring and generate reports are shown. Another feature of SOA Suite, Business

Activity Monitoring (BAM), is also reviewed and used to create a dashboard of critical performance data for the airport.

Chapter 21: Governance

SOA supports creating new services by reusing services that have already been created and deployed. While this is a good thing, it also brings up the need for some way to control all this flexibility. How does a business make sure that all the existing and new services are doing what they should be doing? How does the business decide what new services to create? This chapter introduces the topic of governance, but early on we are told that the subject is too big to cover in this book and really needs a separate book of its own. Specific topics include the issues around reuse of existing components, dependencies, and coexistence of multiple versions of components.

The discussion of canonical data models and message definitions is interesting. Establishing this data model for a business is an important part of the process of moving to SOA.

Chapter 22: Building and Deploying

This chapter discusses the process needed to take a bunch of services from development to a production system. This includes refactoring, debugging, and automated testing as well as auditing. JDeveloper includes an integrated debugger that is discussed. Refactoring, the process of improving the structure and maintainability of source code, can make use of several JDeveloper features. A test suite for SOA composites can be created within JDeveloper as well. The use of Maven, an open-source project management system, is reviewed.

Chapter 23: Run-Time Administration

Once you have all this SOA stuff up and running, you have to worry about some specific administration tasks. One of these is purging the remainders of past events. All of the services are running and generating lots of output. There are many pieces of this output that linger and quickly fill up the database. I've seen this issue many times in my support role. It is not as easy as you might expect to purge all the old records. Even knowing which records are old enough to safely be removed can be complicated. We have sections discussing monitoring, log files, alerts, using Enterprise Manager FMW Control, how best to react to operational issues without destroying things, and how to proactively prevent problems. We are told that in order to maintain stable performance, we may need to throttle the volume of requests that are handled by the SOA Suite. Service Bus provides such a throttling feature. How your customers might feel about this feature is not discussed.

Chapter 24: SOA Suite and Security

This is the last chapter and, up to now, there has been no mention of security. How can that be? We are told again that this topic deserves an entire separate book. The discussion covers how to control the access of administrators (those pesky humans again!) to the contents of service messages, role-based authorization, using partitions to group SOA composites and control user access, hiding sensitive data in log messages, and using encryption. A general overview of security terminology is given, transport versus message security is discussed, and Oracle Platform Security Services (OPSS) is presented. The chapter ends with a review of the auditing features provided by SOA Suite.

Post-Review Comments

After reading this book, I have some questions and concerns that were not addressed. This is not a criticism; I can't expect one book to address all the possible issues of any complex software product.

Now that I've learned that SOA is based on setting up services that interact with other services both local and remote, I'm curious about the performance impact of all this messaging. In theory, it sounds good, but how many users can request how many services before it all grinds to a halt? As you design your SOA application, when do you decide you can't add any more services—or users? How do you overlay all these SOA composites on top of your hardware layout?

My other concern is about security. While there are discussions of passwords and encryption, what about the services you interact with? For any service you don't completely control, how do you know who you are dealing with? For many business processes, you must exchange sensitive information and maybe money. How do you know what the service provider is doing with that data? One of the advantages of SOA is the flexibility to use an alternate service if the preferred service is not available. When you have to switch to the alternate service, can you be sure you will get the same results?

Conclusion

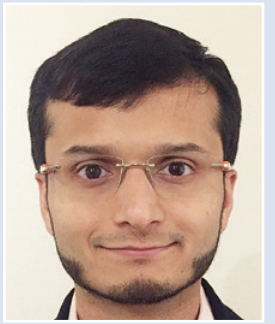
Reading this book provided what I was expecting: a review of all the components of the Oracle SOA Suite.

Since I am not a developer, I can't comment on how useful all the JDeveloper details are, but I do think the level of detail presented would be very useful for someone starting out developing SOA applications. If you have any exposure to SOA Suite in your world, I would encourage you to read this book. I read this book through Safari books online. ▲

Brian Hitchcock works for Oracle Corporation where he has been supporting Fusion Middleware since 2013. Before that, he supported Fusion Applications and the Federal OnDemand group. He was with Sun Microsystems for 15 years (before it was acquired by Oracle Corporation) where he supported Oracle databases and Oracle Applications. His contact information and all his book reviews and presentations are available at **www.brianhitchcock.net/oracle-dbafmw/.** *The statements and opinions expressed here are the author's and do not necessarily represent those of Oracle Corporation.*

Oracle Licensing in the Cloud—Part II

by Mohammad Inamullah



Mohammad Inamullah

Editor's Note: *This article contains information on Oracle licensing that is provided as-is and without guarantee of applicability or accuracy. Given the complex nature of Oracle licensing and the ease with which license compliance risk factors can change significantly due to individual circumstances, readers are advised to obtain legal and/or expert licensing advice before performing any actions based on the information provided.*

Introduction

In my February 2017 *NoCOUG Journal* article on Oracle licensing in the cloud, I went over the basics of correctly sizing instances in cloud environments for Oracle licensing purposes. A few important topics were left unattended in that article, including a detailed discussion of Oracle's updated cloud licensing policy.

In this article, we will close the loop with discussions of the following three items:

1. Oracle's policy and how to deal with it methodically. This section of the article was recently published on my LinkedIn blog. It is reproduced here with updates and additional details.
2. The use of dedicated resources in cloud environments to assist with Oracle license management.
3. Ways in which cloud vendors could help Oracle customers with Oracle license management.

First, the Policy

On January 23, 2017, Oracle stirred up a storm in its user community by announcing a significant change to its licensing policy in third-party cloud environments. With this unilateral pronouncement, Oracle effectively doubled the price of moving traditional on-premise Oracle licenses to Azure and AWS. This policy updated a prior version of the policy that was similar, except that the old policy did not negate the Core Factor table. We will take a closer look at the policy, its implications, and how to deal with it in a methodical and contractually defensible way.

It Is a Policy and Not a Contract

Before discussing the beginning of the document, I will start at the very end: the small-font footer. It reads, in part, as follows:

> "This document is for educational purposes only and provides guidelines regarding Oracle's policies in effect as of January 23rd, 2017. It may not be incorporated into any contract and does not constitute a contract or a commitment to any specific terms."

Even if the above disclaimer was not included in the document, it would still be contractually meaningless. Your signed contract is the only authority on your licensing commitment with Oracle. In my experience, I have not come across any Oracle agreements (OLSAs, OMAs, ULAs, etc.) that include any language like what is stated in the policy.

So, is this policy document entirely inconsequential? It depends (more on that later). But having a firm understanding of the document—and its implications—is important.

Approved Vendors

The policy seems to apply to "approved vendors"—Amazon Web Services and Microsoft—only. Their cloud offerings—AWS's EC2 and RDS, and Microsoft's Azure—are referred to as "Authorized Cloud Environments." It's not clear if there are any obvious benefits to being an approved vendor (other than being targeted by this policy), or if there are any disadvantages to not being an approved vendor. Contractually, there is no reason to believe that moving your Oracle on-premise licenses to other cloud vendors like Google Cloud Platform would constitute a violation or result in license compliance issues.

The Core Factor . . . and Its Elimination

The single most controversial part of the policy is the elimination of the processor core factor that is used in calculating Oracle Processor license requirements. The policy states:

> "For the purposes of licensing Oracle programs in an Authorized Cloud Environment, customers are required to count as follows:
>
> Amazon EC2 and RDS—count two vCPUs as equivalent to one Oracle Processor license if hyper-threading is enabled, and one vCPU as equivalent to one Oracle Processor license if hyper-threading is not enabled.
>
> Microsoft Azure—count one Azure CPU Core as equivalent to one Oracle Processor license.
>
> When counting Oracle Processor license requirements in Authorized Cloud Environments, the Oracle Processor Core Factor Table is not applicable."

Given that the Core Factor for most multi-core chips is 0.5, it serves to reduce the Processor license requirement to half the number of cores. By not applying the Core Factor, per Oracle, customers need twice as many Processor licenses in the cloud to cover the same number of cores as on-premise. For further details on cores, virtual cores, and hyper-threading in cloud environments, refer to this article (**nocoug.org/Journal/NoCOUG_Journal_201702.pdf**). What makes this aspect of the policy especially unusual, to say the least, is that it appears to negate a contractually defined component. Applying the Core Factor Table is part of the contract (in the Processor definition).

Separately, for socket-based programs like Database Standard Edition, SE1, and SE2, Oracle has revised its position from the prior policy and reduced the number of virtual cores that equate to an occupied socket. Arbitrarily, four vCPUs in AWS and two virtual cores in Azure correspond to an occupied socket. Note that in the older cloud licensing policy, it was four virtual cores corresponding to an occupied socket. In AWS, that was eight vCPUs (due to hyper-threading) and four virtual cores in Azure. Consequently, this also reduces the maximum allowable number of AWS vCPUs and Azure virtual cores where SE, SE1, and SE2 programs may run.

On a related note, Oracle has added guidance for licensing the Standard Edition products in Oracle Cloud in the Processor Core Factor table itself (**www.oracle.com/us/corporate/contracts/processor-core-factor-table-070634.pdf**). This is an interesting move, since the Core Factor table is contractually referenced and isn't just a policy. Per the document:

> "When licensing Oracle programs with Standard Edition One, Standard Edition 2 or Standard Edition in the product name . . . where a processor license is counted equivalent to an occupied socket, every one (1) Processor license covers the use of the program on four (4) OCPUs."

An Oracle Compute Unit (OCPU) is an x86 core with hyper-threading enabled. As such, four OCPUs equate to eight hyper-threads. Note that the limit in AWS is four vCPUs (or hyper-threads), and the limit in Azure is two virtual cores (with no hyper-threading). The new position for Standard Edition products makes it twice as expensive to license these products in AWS and Azure than in Oracle Cloud.

Exclusion from Unlimited License Agreement (ULA) Certifications

The other significant edict relates to ULA certifications. Per the policy, while customers may deploy their ULA programs in AWS and Azure, they may not include these numbers in their ULA certifications. The document states:

> "Licenses acquired under unlimited license agreements (ULAs) may be used in Authorized Cloud Environments, but customers may not include those licenses in the certification at the end of the ULA term."

As a result, a large company with substantial Oracle deployments in AWS and/or Azure stands to leave tremendous value on the table due to not being able to include those deployments in the certification. Instead, those deployments will effectively become compliance liabilities requiring immediate purchase or uninstallation. This part of the policy is clearly very challenging and intimidating for the large enterprise customers that have come to rely on Oracle's ULAs as a means of licensing their large and/or high-growth environments. It's important to note that Oracle's previous policy on cloud licensing also had this exclusion for ULA certifications.

Policy for Oracle VM in AWS EC2 Environments

A section in the old policy read, in part, as follows:

> "Amazon has implemented Oracle VM EC2 instances in accordance with the practices defined in the Oracle policy document. . . . From an Oracle product licensing point of view, this means that each virtual processor is equivalent to a physical core, and the standard Oracle Processor metric definition applies."

The new policy has completely removed this section. I don't know the reason for this.

Is This Policy Inconsequential?

Whether this policy is inconsequential depends on you, the customer. Understanding your contracts and having a well-thought-out game plan is important. Here are some of the key points to keep in mind as you develop your Oracle-in-the-cloud strategy:

- This document is a policy and should have no role in your discussions with Oracle.
- Ensure that you understand all your relevant licensing metrics as they are defined in your contract, including the processor metric.
- Consider limiting and strictly controlling information that your company shares with Oracle sales and auditors, especially details around the cloud and virtualization.
- For whichever cloud vendor you consider, make sure you have a firm grasp on issues such as differentiating between vCPUs, virtual cores, and hyper-threading.
- Remember that counting cores and processors correctly is only one aspect of maintaining compliance in the cloud.
- When working with Oracle Sales and Oracle License Management Services, request that they provide any information, guidance, or their licensing position in writing.
- Use contractual language to hold your position when discussing Oracle licensing in the cloud with Oracle. This fundamental rule should apply to all areas of software license management.
- There are a couple of things to keep in mind regarding ULA certifications:

  First, there is no contractual reason not to count cloud deployments in your ULA certification. In the ULA contracts I've seen, I've never come across any exceptions to or exclusions from including deployments in your certification. As long as the software is installed and running, it should be included in the certification.

  Second, for the ULA certification, you are not required to share any environment details, server names, virtualization configurations, or cloud deployment details with Oracle. The certification process does not entitle Oracle to anything other than a signed piece of paper with usage

quantities (**www.redwoodcompliance.com/oracle-ulas-lost-value-compliance-exposures/**).

Dedicated Resources

The forgoing discussion may leave some readers with a sense of uncertainty around Oracle license compliance in the cloud. One alternative that can certainly enhance confidence around compliance involves using dedicated resources. The nomenclature may vary slightly from vendor to vendor, but the idea is the same: customers can pay for instances where the underlying hardware is dedicated. From an Oracle licensing perspective, this makes for an even more defensible position in being able to positively identify the processors where Oracle software is running. This can be especially helpful in socket-based licensing of the "Standard" Edition products, and also provide a stronger argument of correct processor and core-level licensing for Oracle's processor metric–based licensing. Let's take a quick look at the current offering landscape among the leading cloud vendors.

AWS Dedicated Hosts

AWS offers dedicated hosts whereby customers get fully dedicated physical hosts for their instances. According to AWS:

> *An Amazon EC2 Dedicated Host is a physical server with EC2 instance capacity fully dedicated to your use. Dedicated Hosts can help you address compliance requirements and reduce costs by allowing you to use your existing server-bound software licenses.*

For further reading, continue here: **aws.amazon.com/ec2/dedicated-hosts/**

While there is a premium attached to this, this option does enhance licensing certainty by enabling Oracle customers to dependably identify and account for hardware running Oracle software. Additionally, while I have not validated the effectiveness for Oracle licensing purposes, AWS Config service (**aws.amazon.com/config**) could be used for enhanced tracking and monitoring of instance activity.

It should be noted that, at the time of this writing, Amazon's RDS service does not seem to offer the dedicated host option. Launching Oracle Database (Enterprise Edition or the socket-licensed SE/SE1/SE2 flavors) in RDS happens on traditional instances with no insight into tenancy.

Azure

Microsoft's Azure service does not offer a dedicated hosting option, per se. I say "per se" because Azure does document that some of its larger instance classes are in fact dedicated hardware resources—for example, its Dv2 series instances (**azure.microsoft.com/en-us/updates/announcing-new-dv2-series-virtual-machine-size**). In its announcement of the new instance class, Azure states:

> *Each Standard_D15_v2 instance is isolated to hardware dedicated to a single customer, to provide a high degree of isolation from other customers. This addition and the Standard_G5 are the two available sizes that are on hardware dedicated to a single customer.*

It's worth noting that the instance class mentioned above is fairly large: 20 cores and 140 GB of RAM. However, this does not appear to be offered as an intended or guaranteed dedicated hosting option. Customers looking for dedicated hosts in Azure may need to consult with Azure account management for details about which instance classes include dedicated hardware before relying on these for Oracle license management.

Google Cloud Platform (GCP)

At the time of this writing, GCP's Compute Engine offering does not appear to offer any dedicated hosting options. On a somewhat related note, with regard to its Compute Engine offerings, GCP masks many technical details, including the processor model. While this does not simplify things from an Oracle licensing perspective, it does make it a lot harder for Oracle auditors to potentially pursue compliance findings against a customer if there isn't much technical detail forthcoming from the cloud vendor.

Simplifying Oracle Licensing in the Cloud—How Vendors Can Help

A customer running Oracle on a known number of cores on premise should not need to pay more in Oracle licensing for running the same Oracle software on the same number and class of cores on the cloud. Short of a new licensing framework, policy guidance—even if it's non-contractual but is nonetheless reasonable and does not penalize customers for moving to alternative cloud vendors—would be a positive move too.

This leads to a corollary. Can leading cloud vendors—AWS, Azure, GCP—do anything to help their customers with Oracle licensing management? While this may not be a top priority for them, and they will typically want to promote their own database options as a first choice, a number of ideas come to mind:

1. Offer dedicated hosting options. AWS already does this. Perhaps Azure and GCP should follow suit on this. AWS could consider offering RDS options on dedicated hosts for Oracle Database Enterprise Edition and Standard Edition products.
2. Allow some sort of reporting or tracking of workloads that can report on the details of the hardware touched by the instances. For Oracle products with socket-based licensing, it would be beneficial for customers to get some sort of reporting indicating that the instances were, in fact, bound to identifiable and countable sockets.
3. Cloud vendors could share more details publicly on how instances are tied to their CPU resources and how often the hardware allocation gets refreshed, etc. Such information may not be a surefire solution to Oracle licensing in the cloud, but it will enable software asset management personnel to make informed decisions through better understanding of the underlying hardware.

Concluding Remarks

The march to the cloud continues. Customers with investments in Oracle software should have a clear and crisp understanding of their Oracle contracts, with access to expert licensing assistance if needed. You should consider all cloud options and let the decision making be driven by best fit and how things fall in line with your strategic plan. ▲

Mohammad Inamullah is the principal at Redwood Compliance in Palo Alto, California. He can be contacted at **mohammad@redwoodcompliance.com**.

Controlling and Monitoring Oracle Recovery Manager—Part I

Norbert Debes

by Norbert Debes

Introduction

All enterprises that entrust their data to the Oracle DBMS software require a sound backup and recovery strategy for their databases. Most companies use home-grown program code to control Oracle Recovery Manager (RMAN). Having served as a consultant for numerous companies in the telecommunications, automotive, and pharmaceutical industries, I am very much aware of frequent deficiencies in program code for controlling RMAN. In this article I present some aspects of a commercial software package for controlling RMAN that is called "*RmanJ*." RmanJ aims to remedy all known deficiencies, such as clear-text passwords, failure to check the RMAN return code, lack of command line switches, missing notification by email, lack of flexibility, no monitoring, and no reporting of RMAN data volume and throughput. RmanJ takes controlling RMAN even one step further. It offers much needed parallel backup and restore capabilities for large Oracle Standard Edition (SE) databases. A customer has reported that a full backup with Oracle SE that used to take around 6 hours and 45 minutes, due to the single channel limitation of SE, now takes just under one hour with RmanJ and a parallel degree of eight separate RMAN invocations that share the backup workload and are controlled by RmanJ. Last but not least, RmanJ may also be used to implement a standby database environment for Oracle SE that does not include Data Guard as a standby database solution.

This article is the first of two parts. In this instalment I will address controlling and monitoring RMAN. Part 2 will focus on parallelizing RMAN operations in an Oracle SE environment. Both articles use RmanJ as a blueprint for implementing the features discussed. For those who would like to gain a more thorough and complete understanding of RmanJ features, I recommend reading the *RmanJ User Guide* that is available at **www.oradbpro.com/rmanj.html**.

Time Pressure

At some point in their careers, numerous Oracle database administrators (DBAs) have had to perform database restore or backup operations under time pressure. Some of the worst-case scenarios are downtime during business hours due to disk failure, data file corruption, or incorrect data processing that can only be undone by a tablespace point-in-time recovery. Management's favorite question in such a scenario usually is, "How much longer will it take?" An experienced DBA needs to be able not only to answer this question but to answer it with precision. Another example is a maintenance window that is winding down while a DBA is waiting for a backup to complete, since he or she needs a consistent snapshot of the database should something go wrong. Forecasting the remaining time of the RMAN backup operation becomes crucial to avoid unplanned downtime beyond the approved downtime window. Using correct forecasting, the maintenance can be cancelled ahead of time should the backup take too long. In doing so, no extra downtime is incurred.extra downtime

Diagnosability

Taking RmanJ, a control utility for Oracle RMAN, as an example, I will go into some details of controlling and monitoring RMAN as well as reporting on ongoing and completed RMAN operations. This article will show that the Oracle RDBMS has some quite sophisticated diagnosability features for RMAN operations. Oracle's own manual, *Oracle® Database Backup and Recovery User's Guide 12c Release 1 (12.1) E50658-07 November 2015,* contains some information on these features. However, the matter is much more complex and comprehensive than suggested by the short section on the view `V$SESSION_LONGOPS` in the aforementioned manual.

Presumably, many DBAs have been involved in coding what I will call a control utility for Oracle RMAN—program code that invokes RMAN with the required environment variable settings, the correct command line arguments, and, of course, the RMAN commands that perform backup or restore operations.

RMAN is not capable of sending email notifications. RMAN uses locking mechanisms within the RDBMS, but it has no locking to prevent execution of the same commands or different commands that interfere with an ongoing operation. Consider the following example: A full database backup is running. Archived logs are backed up and removed near the end of the full database backup script. If a separate invocation of RMAN backs up and removes archived logs that the full database backup has seen when it started, the full database backup will be marked as failed, since it will try to access archived logs that have already been backed up and removed by another invocation of RMAN.

RMAN Catalog

RMAN cannot protect passwords. Monitoring RMAN requires a database session and program code that runs independently of RMAN. The Java programming language is ideally suited to control and monitor RMAN since it is capable of creat-

ing subprocesses and communicating with them. Java offers parallel programming, and encryption algorithms are also integrated into Java development kits (JDK). Considering that basically all versions of the Oracle RDBMS have security flaws that may be leveraged to escalate privileges and that an RMAN catalog may reside in a production database, it is obvious that safeguarding passwords is important. In the paragraphs that follow I will briefly explain what a *Secure External Password Store* is. Next, I will present how RmanJ protects database passwords. Please keep in mind that all shell programming languages (Bourne, Korn, Bash, etc.), as well as Perl, are not capable of protecting passwords.

A connection to an RMAN recovery catalog database always requires a password. Connections to target and auxiliary instances may also require passwords, albeit for highly privileged user accounts. Prior to Oracle12*c*, SYSDBA privileges were mandatory for using RMAN. Since the introduction of Oracle12*c*, the SYSBACKUP privilege that does not provide access to tables in user schemas is sufficient. Practically all companies have a security policy in place that prohibits storing clear-text passwords in unencrypted files such as RMAN or shell scripts. How can a DBA comply with such a policy? The RDBMS has a feature called "*Secure External Password Store*." Essentially, it is a wallet that contains an encrypted password at a certain indexed position within the file. The wallet must have automatic login enabled such that anyone with access to the file and knowledge of the indexed entry within the wallet that holds the login credentials gains access to the underlying database account. It becomes a matter of protecting the wallet and the control script, since the latter must contain the name of the indexed entry within the wallet. All copies of the wallet and control script on all filesystems and backup media need to be protected.

RmanJ and Password Encryption

RmanJ goes one step further in that it offers password encryption without a hardcoded encryption key. In order to gain access to a password, an attacker would need access to RmanJ software, to an RmanJ configuration file that contains an encrypted password, and, finally, to the environment where the password encryption was performed. Furthermore, it would probably be necessary to decompile certain Java classes. Attempts to use an encrypted password outside of the environment where it was encrypted fail with the Java exception `javax.crypto.BadPaddingException: Given final block not properly padded`.

RmanJ is available on all platforms that support Java and the Oracle DBMS. On Unix-like platforms a shell script wrapper called rmanj is used to invoke a Java Virtual Machine (VM). On Windows a command script called rmanj.bat is used. Password encryption with RmanJ is simple. The command line switch -e invokes the encryption component of RmanJ:

```
$ rmanj -e
RmanJ startup on 06-Mar-2017 16:58:01 - Version 2.0 (RCS revision 1.49) built by
ORADBPRO GmbH on 2017/02/27 12:32:17 is executed by JVM with PID 19282 on host
devsrv.oradbpro.com running operating system Linux
Password:
Password (verification):
cleartext password length: 10 bytes; encrypted password: 'Uv3IGVlVdWLbbIiQL8R2iA=='
```

Up to three different encrypted passwords (target instance, auxiliary instance, catalog instance) may be required in order to run an RMAN script. Since RmanJ is implemented in Java, it uses the Java property file format for storing configuration files. The names of the password properties are self-explanatory:

```
$ grep encr rmanj.properties
target_encrypted_passwd=tV3MZ5mGIjlyiev17O1V1InQ+Dfffu/N
catalog_encrypted_passwd=9rUNDr3pVFvjIZYx3dyvO8Ov8M+oxaA+
#auxiliary_encrypted_passwd=tV3MZ5mGIjlyiev17O1V1InQ+Dfffu/N
```

Whenever one of the above properties is set, RmanJ will instruct RMAN to connect to the respective instance. In the example above, properties are set for target and catalog passwords. The property for an auxiliary instance is commented out. Thus RmanJ would instruct RMAN to connect to both a target and a catalog instance. RmanJ supports more than three dozen properties. Please refer to the *RmanJ User Guide* for details.

A Minimalistic RmanJ Example

Of course, RmanJ also needs to know user names and connect strings as well as commands to execute. These are set using additional properties with equally self-explanatory names.

```
$ grep -v ^# simple-backup.properties
ifile=/home/ndebes/src/rmanj.common.properties
log_directory=/home/ndebes/src/log
log_file=%HOSTNAME%-%ORACLE_SID%-Inc%INC_LEVEL%-%START_DATE_TIME%.log
internal_logon=SYSBACKUP
target_user="c##ora$rman_bkp"
target_encrypted_passwd=tV3MZ5mGIjlyiev17O1V1InQ+Dfffu/N
target_connect_string=twelve_bkp_ipc
backup_cmd=BACKUP SPFILE TAG "%START_DATE_TIME%";
backup_epilogue=LIST BACKUP TAG "%START_DATE_TIME%";
catalog_encrypted_passwd=9rUNDr3pVFvjIZYx3dyvO8Ov8M+oxaA+
catalog_connect_string=pdbone
```

The property `backup_cmd` contains the main command that RMAN will execute. There are also prologue and epilogue commands for backup, copy, and restore operations. In the example above, the property `backup_epilogue` specifies one or more RMAN commands that will be executed after the actual backup operation. From the value of the property `target_user` it is evident that RmanJ runs against a container database (CDB). Just like RMAN, RmanJ can be used with non-CDBs, CDBs, and pluggable databases (PDBs).

Note the string `%START_DATE_TIME%`. RmanJ supports several placeholders that are expanded before commands are handed over to RMAN. `%START_DATE_TIME%` is replaced by the date end time at which RmanJ was invoked. As such it is suitable for assigning a unique tag to RMAN backup pieces. The value SYSBACKUP of property `internal_logon` above shows that RmanJ can use the user SYSBACKUP for RMAN operations. SYSBACKUP is a new user in 12*c* that allows RMAN to run with fewer privileges than in prior releases. I recommend using SYSBACKUP instead of SYS in 12*c*, especially when connecting with a password as opposed to operating system authentication without a password. We are now ready to invoke RmanJ for the first backup. The switch -o selects the type of operation (copy, backup, or restore).

```
$ rmanj -o backup -f simple-backup.properties
```

The RmanJ command above generates 161 lines of output on standard out. The entire output is also written to a log file:

```
$ wc -l /home/ndebes/src/log/devsrv.oradbpro.com-TWELVE-Inc0-2017Mar06-17h52m13s.log
161 /home/ndebes/src/log/devsrv.oradbpro.com-TWELVE-Inc0-2017Mar06-17h52m13s.log
```

There is not enough space to show the entire log file here. Again, please refer to **www.oradbpro.com/rmanj.html** for an example of a complete RmanJ log file. On startup, RmanJ provides a wealth of information on the target database. This includes all the tablespaces, their block sizes and tablespace sizes, and how they relate to CDBs or PDBs (or a non-CDB). The log file also shows whether archivelog mode and flashback database are enabled. Due to space limitations the word "Tablespaces" in the last four lines of the output below had to be truncated to "Ta."

```
Container      Tablespace Name     Block Size Size(GB) Files
-------------- ------------------- ---------- -------- -----
CDB$ROOT       SYSAUX                    8192    0.586     1
CDB$ROOT       SYSTEM                    8192    0.391     1
CDB$ROOT       UNDOTBS                   8192    0.576     1
CDB$ROOT       USERS                     8192    0.005     1
PDB$SEED       SYSAUX                    8192    0.161     1
PDB$SEED       SYSTEM                    8192    0.205     1
PDBONE         SYSAUX                    8192    0.171     1
PDBONE         SYSTEM                    8192    0.205     1
PDBONE         USERS                     8192    0.012     1
CDB$ROOT       All Permanent Ta                  1.558     4
PDB$SEED       All Permanent Ta                  0.366     2
PDBONE         All Permanent Ta                  0.388     3
All Containers All Permanent Ta                  2.312     9
```

RmanJ uses operating system file-locking mechanisms to prevent concurrent execution of multiple RmanJ invocations against the same database. The lock file name contains the database ID:

```
Acquired lock (Success: Obtained valid file lock on file /tmp/1641612861.lck)
```

The log file appears as if RMAN had been used interactively. This makes RmanJ log files very convenient to read:

```
Recovery Manager: Release 12.1.0.2.0 - Production on Mon Mar 6 17:52:20 2017

Copyright (c) 1982, 2014, Oracle and/or its affiliates.  All rights reserved.

RMAN>
RMAN-03029: echo set on

RMAN> CONNECT TARGET *

RMAN-06005: connected to target database: TWELVE (DBID=1641612861)

RMAN> SET COMMAND ID TO '20170306175213198957=1F=OBS=8192';
RMAN-03023: executing command: SET COMMAND ID

RMAN> BACKUP SPFILE TAG "2017Mar06-17h52m13s";
```

Most control scripts for RMAN use standard input to feed commands to RMAN (`rman <<EOF <command sequence> EOF`). Others use the command line argument `CMDFILE` to execute a file containing RMAN commands. No matter how commands are passed to RMAN, `SET ECHO ON` should always be used to render log files more readable, as in the RMAN output above.

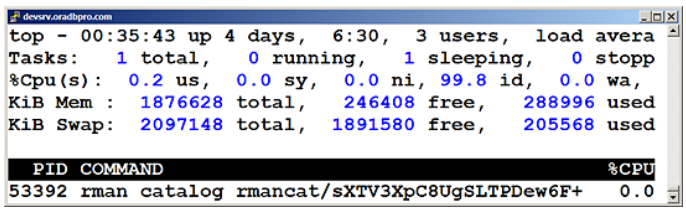


Figure 1: Passwords specified on the RMAN command line are exposed through the process monitoring tool top

Passwords should never be passed as arguments to RMAN since they will be visible as part of the process list. Passwords in RMAN scripts are also problematic since unauthorized persons might gain access to them. I have seen clear-text passwords in world-readable files (-rw-r--r--) several times during my career. Hence the best way to handle passwords is to pass them over a private communication channel such as standard input or a pipe-style connection to an RMAN subprocess.

Normally, only RMAN error stacks are prefixed with RMAN message numbers in the format RMAN-nnnnn where n is an integer. The five-digit message number may be translated to the corresponding message using the command oerr:

```
$ oerr rman 3023
3023, 1, "executing command: %s"
// *Cause: This is an informational message only.
// *Action: No action is required.
```

RMAN supports a command line argument named `MSGNO`. `MSGNO` is ideally suited to evaluate RMAN's responses to commands since each line output by RMAN will have RMAN-nnnnn as a prefix. I recommend invoking RMAN with `MSGNO` as the only command line argument. As a consequence, all the database connections have to be made using CONNECT commands, and password exposure on the command line is eliminated. Furthermore RMAN-nnnnn strings can be compared to a set of expected RMAN responses to commands. The example below shows the difference between running RMAN without message numbers enabled (default) and with `MSGNO` (message numbers turned on). Note how the second excerpt below includes RMAN message 6005 since `MSGNO` was used as a command line argument.

```
$ rman
RMAN> connect target '"c##ora$rman_bkp"@twelve_bkp_ipc as sysbackup'
target database Password:
connected to target database: TWELVE (DBID=1641612861)
$ rman msgno
RMAN> connect target '"c##ora$rman_bkp"@twelve_bkp_ipc as sysbackup'
target database Password:
RMAN-06005: connected to target database: TWELVE (DBID=1641612861)
```

If an RMAN catalog user, password, and connect string are specified, RmanJ does not connect to a recovery catalog until after all backups have completed. Bear in mind that a catalog instance might be unavailable. This would result in a failed backup. Some of the RMAN control scripts that I have seen in the field always connect to the target and catalog instances at the same time. RmanJ eliminates the risk of failed backups due to catalog outage by using a separate RMAN invocation to synchronize catalog and control file after all backup operations have completed.

Each RMAN command is logged in the control file, given that the database is mounted. Access to the log is available via the V$ views `V$RMAN_STATUS` and `V$RMAN_OUTPUT`. RmanJ generates command IDs based on date and time along with additional information such as the task ID assigned by RmanJ. This feature allows RmanJ to monitor the success or failure of RMAN operations as well as the data volume processed by RMAN. Based on the command ID set by RmanJ, it is possible to report that the SPFILE backup was successful and that it processed ten blocks with a block size of 8192 bytes.

```
SQL> SELECT end_time, status, output_bytes
FROM v$rman_status
WHERE command_id='20170306175213198957=1F=OBS=8192';
```

```
END_TIME            STATUS             OUTPUT_BYTES
------------------- ------------------ ------------
06-Mar-2017 17:52:23 COMPLETED                81920
```

The reporting shown above is incorporated into RmanJ and performed automatically after each stage of RmanJ execution (i.e., after a prologue, main command, epilogue, and archived log backup, and once again after all stages have completed). Below is an excerpt from an RmanJ log file that reports the status and volume of a data file and archived log backup for a small test database:

```
Successfully released lock (Success: Released valid lock on file /tmp/1641612861.lck)
RmanJ invoked on 01-Mar-2017 22:31:17 and finished on 01-Mar-2017 22:33:05 (total
elapsed time 00:01:48.493)

Status    Object Type   Input GB Output GB Input/Output GB
--------- ------------ -------- --------- ---------------
COMPLETED ARCHIVELOG    0.006063  0.002223        0.008286
COMPLETED DATAFILE INCR 1.571564  0.336281        1.907845
Any       Any           1.577627  0.338504        1.916131

RmanJ is terminating with exit code 0 (0 RMAN errors encountered)
```

Note that RmanJ parses the output returned by RMAN for error stacks. In other words, RmanJ counts how many errors RMAN encountered. RmanJ only terminates with exit code zero if RMAN did not produce any error stacks. RmanJ has several properties for email notification. One of them is smtp_notification. Additional properties control the sender, recipient, subject, and SMTP server. The default value of smtp_notification is error. Thus RmanJ will send an email message only if RmanJ detected an error. When smtp_notification=always is set, RmanJ will send a message each time it runs, independently of any errors that may have occurred.

RMAN Throughput

Oracle Corp. published the following query for monitoring RMAN progress in their *Backup and Recovery User's Guide*:

```
SELECT SID, SERIAL#, CONTEXT, SOFAR, TOTALWORK,
ROUND(SOFAR/TOTALWORK*100,2) "%_COMPLETE"
FROM V$SESSION_LONGOPS
WHERE OPNAME LIKE 'RMAN%'
AND OPNAME NOT LIKE '%aggregate%'
AND TOTALWORK != 0
AND SOFAR <> TOTALWORK;

SID    SERIAL#    CONTEXT      SOFAR  TOTALWORK %_COMPLETE
--- ---------- ---------- ---------- ---------- ----------
291       3786          1       9470      22400      42.28
```

This query is certainly helpful, but it leaves much room for improvement. First of all it does not consider RAC environments where RMAN backup operations may be spread out over several cluster nodes in order to leverage the computing power and I/O bandwidth of several nodes. Hence the view `GV$SESSION_LONGOPS` should be used instead of `V$SESSION_LONGOPS`.

The query result above does not inform the DBA how much more time the RDBMS instance will need to complete a backup or restore operation. The kind of operation that is underway is also not part of the result. Hence a closer look at `GV$SESSION_LONGOPS` is warranted. The type of RMAN operation is available in the column opname.

```
SELECT DISTINCT l.opname, l.units
FROM gv$session_longops l
WHERE l.opname like 'RMAN:%';
```

```
OPNAME                             UNITS
---------------------------------- ------
RMAN: full datafile restore        Blocks
RMAN: aggregate input              Blocks
RMAN: aggregate output             Blocks
RMAN: incremental datafile backup  Blocks
```

The unit of all RMAN operation is "Blocks," as in the query result above. The RDBMS supports up to five block sizes in a single database. Unfortunately, no information is available on the size of the Blocks handled by RMAN. There are two simple cases, however. The block size of archived logs is always 512 bytes:

```
SQL> SELECT DISTINCT block_size FROM v$archived_log;

BLOCK_SIZE
----------
       512
```

Most databases have merely a single block size:

```
SQL> SELECT block_size, count(*) FROM v$datafile GROUP BY block_size;

BLOCK_SIZE   COUNT(*)
---------- ----------
      8192          9
```

At least two more columns of `GV$SESSION_LONGOPS` are noteworthy. They are `TIME_REMAINING` (in seconds) and `ELAPSED_SECONDS`. Throughput is calculated by the following formula:

> Throughput (MB/s) = (Blocks * Block_Size / 1048576) / Elapsed_Time

Matters are more complicated when a database has multiple block sizes:

```
SQL> SELECT block_size, count(*) FROM v$datafile GROUP BY block_size;
BLOCK_SIZE   COUNT(*)
---------- ----------
      2048         45
      8192        122
```

Unless RMAN is controlled in a special way, it is impossible to correlate an RMAN operation on a set of data files with the block size of those data files. Two solutions exist:

- Backup or restore a database by tablespace. Each tablespace has but a single block size. Thus the formula above can be applied.
- Backup or restore a database by data file.

Both approaches require an RMAN control utility that can either determine all the tablespace names (`V$TABLESPACE.NAME`) or all the data file numbers (`V$DATAFILE.FILE#`) of a database and generate RMAN commands accordingly. RmanJ currently implements only the second approach. For this purpose it provides the placeholder `%FILE_ID%`. If this placeholder appears within the property `backup_cmd` (restore_cmd or copy_cmd may also be used) then RmanJ creates a task for each data file that will also run in parallel if the RmanJ property `parallelism` has a value that is greater than 1.

An undocumented feature needs to be leveraged in order to figure out which RMAN session handles which data file (or tablespace). As far as I know it is not documented that the command ID set within RMAN appears as part of `V$SESSION.CLIENT_INFO`.

```
RMAN> SET COMMAND ID TO 'NoCOUG';

executing command: SET COMMAND ID
SQL> SELECT inst_id, sid, serial#, substr(program, 1, 4) AS program, client_info
FROM gv$session
WHERE client_info LIKE '%NoCOUG%';

 INST_ID  SID  SERIAL# PROGRAM  CLIENT_INFO
-------- ---- -------- -------- ------------
       1  296    45063 rman     id=NoCOUG
```

Due to space constraints I can only use a small number of columns in the SELECT below that joins database sessions with long operations. I invite the reader to amend the queries that follow with additional columns. The query below was executed while RMAN was processing the command `BACKUP DATABASE`:

```
SELECT l.opname, s.client_info, l.time_remaining AS "Remain.(s)", l.elapsed_seconds
AS "Ela(s)"
FROM gv$session s, gv$session_longops l
WHERE s.inst_id=l.inst_id
AND s.sid=l.sid
AND s.serial#=l.serial#
AND l.opname like 'RMAN:%'
AND l.sofar<l.totalwork;

OPNAME                    CLIENT_INFO   Remain.(s)  Ela(s)
------------------------- ------------ ---------- -------
RMAN: aggregate input      id=NoCOUG                   0
RMAN: full datafile backup id=NoCOUG,rm          7      7
                          an channel=0
                          RA_DISK_1
```

As can be seen in the query result above, the command ID is propagated to the session that performs the data file backup. Thus a RMAN control utility that is able to set a command ID and to make sure that each RMAN command handles a single data file is able to derive the block size of an operation by joining `GV$SESSION` and `GV$SESSION_LONGOPS` thereby inferring the block size from the command ID in the result. When all these prerequisites are met, it becomes feasible to calculate RMAN throughput in MB/s for databases with multiple block sizes. The downside of this approach is that `V$SESSION_LONGOPS.TIME_REMAINING` no longer displays the remaining time for an entire database, since the database backup is split up into several RMAN commands. The remaining time may be forecasted based on the data volume that has been processed already (reported by RmanJ), the database size, and the throughput.

RmanJ attempts to calculate the throughput every five minutes unless the command line switch -m is used to choose a different monitoring interval. RmanJ also provides information on the current wait event and wait time for RMAN sessions. A stalled tape access due to a media manager being unable to gain access to an unused tape drive becomes very easy to detect by looking at the wait time and wait event name.

Figure 2 is an excerpt of monitoring information provided by RmanJ. The screenshot shows different sections of the same file using VIM (Vi IMproved). The top portion of the window shows output of RmanJ's periodic monitoring. This type of output is always preceded by a timestamp. It includes the column `V$PROCESS.SPID`, under the column heading "Spid." This column contains the operating system process identifier of a dedicated server process that handles an RMAN command. This information is useful when RMAN is interrupted but not all database sessions created by RMAN are terminated. An operating system kill command can be used to forcefully terminate such sessions.

The data type of column "Days Remaining" is `INTERVAL DAY TO SECOND`. Throughput information is provided under the column heading "MB/s." Notice that throughput information is available merely for the operation "aggregate input." This type of session controls other RMAN sessions that perform the actual workload. In this case, Enterprise Edition was used with three channels allocated by RMAN itself. In other words, parallelism within RmanJ was not used, since Enterprise Edition provided the parallelization.

The bottom portion shows the summary section that RmanJ generates before terminating. It informs the DBA when it started, how much elapsed time it incurred, and when it finished. It also provides an overview of the RMAN operations that were invoked and the data volume processed. Finally, RmanJ reports the exit code it will use and how many error messages it recognized within the RMAN output. Thus the usage of RmanJ in conjunction with RMAN has been shown from invocation to termination. This concludes the first installment on RmanJ and details of its implementation that can be applied to any control script for Oracle RMAN.

Summary

Controlling and monitoring Oracle RMAN is the subject of this installment. The article shows how to avoid exposing passwords, how to make RMAN log files more readable (`SET ECHO ON`), and how to use `MSGNO` in order to make RMAN's output more accessible to parsing. Several V$ views, such as `V$SESSION_LONGOPS`, `V$RMAN_STATUS`, and `V$SESSION`, form the basis of sound monitoring of RMAN operations. The article demonstrates how the remaining time of an RMAN operation may be retrieved from `V$SESSION_LONGOPS`. Calculating the throughput of an active RMAN operation in MB/s is more involved and may require the use of `SET COMMAND ID TO` as well as making sure that each RMAN command deals with data files of a single block size. The data volume processed by RMAN may be derived by selecting from `V$RMAN_STATUS`.

The article briefly mentions that RmanJ is able to parallelize RMAN operations in an Oracle Standard Edition environment. The second installment will address the subject of parallelization in detail. ▲

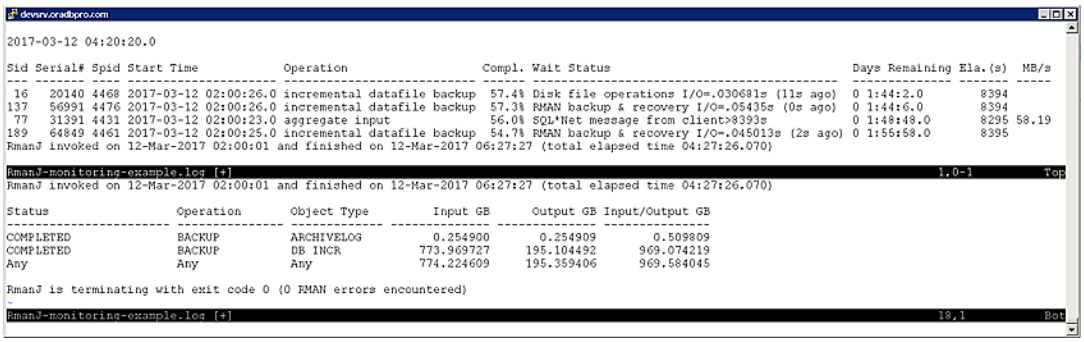


Figure 2: An example of monitoring information generated by RmanJ

Norbert Debes is the principal at ORADBPRO GmbH, Germany

Mission-Critical Databases in the Cloud—Oracle RAC in Microsoft Azure

Artem Danielov

By Artem Danielov

Ensuring high availability of backend relational databases is a critical part of the cloud migration strategy. Many IT organizations have been using Oracle RAC as a trusted high-availability solution for running mission-critical databases and would like to continue using it in the cloud. Additionally, many of those organizations that have not been using Oracle RAC in their on-premise environments see Oracle RAC as the technology enabling the required level of database high availability in the cloud. However, Oracle RAC has the following infrastructure requirements that are not directly available in public clouds:

- Shared high-performance storage accessible from all nodes in the cluster
- Multicast-enabled network between all nodes in the cluster
- Separate networks for different types of traffic: client, cluster interconnect, and storage

FlashGrid Storage Fabric and FlashGrid Cloud Area Network™ (CLAN) technologies address these requirements and enable mission-critical Oracle RAC clusters in public clouds. This paper provides architectural overview of the solution that should help with planning and designing Oracle RAC deployments in a Microsoft Azure public cloud.

Introduction to FlashGrid Software

High-speed shared storage is critical for seamless database infrastructure failure handling with zero downtime and zero data loss. FlashGrid Storage Fabric software enables high-speed shared storage in a variety of infrastructure environments—including bare-metal servers, virtual machines, or extended distance clusters—without the use of proprietary storage arrays. FlashGrid Storage Fabric adds shared access required by Oracle RAC to the standard storage capabilities of the Azure compute cloud.

FlashGrid Cloud Area Network software enables migration of mission-critical applications to the Azure cloud by bridging the gap between the standard network capabilities of the Azure virtual networks and the networking requirements of Oracle RAC.

Why Oracle RAC in the Azure Cloud?

Oracle RAC provides an advanced technology for database high availability. Many organizations use Oracle RAC for running their mission-critical applications, including most financial institutions and telecom operators where high availability and data integrity are of paramount importance.

Oracle RAC is an active-active distributed architecture with shared database storage. The shared storage plays a central role in enabling automatic failover, zero data loss, 100% data consistency, and in preventing application downtime. These HA capabilities minimize outages due to unexpected failures, as well as during planned maintenance.

Oracle RAC technology is available for both large-scale and entry-level deployments. Oracle RAC Standard Edition 2 provides a cost-efficient alternative to open-source databases while ensuring the same level of high availability that the Enterprise Edition customers enjoy.

FlashGrid software brings the superior economics, flexibility, and agility of Azure to a broad range of Oracle RAC customers. It enables existing enterprise Oracle RAC customers to realize the full benefits of migrating their entire IT infrastructure to Azure. It also lowers entry barriers for new customers starting with small scale database deployments.

FlashGrid supports 2- or 3-node Oracle RAC configurations in Azure. Configurations with 4+ nodes are planned in the future.

Configurations with two RAC nodes

Configurations with two RAC nodes have 2-way data mirroring using Normal Redundancy ASM disk groups. An additional VM is required to host quorum disks. Such a cluster can tolerate the loss of any one node without database downtime.

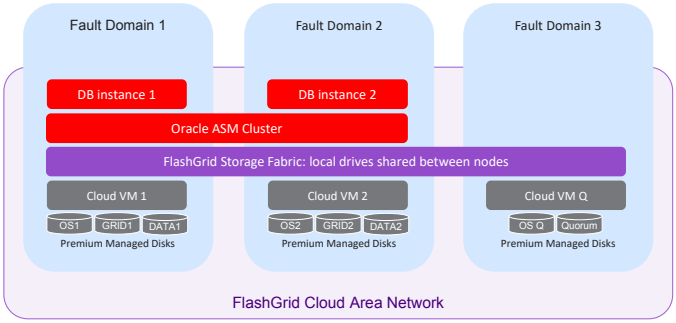


Figure 1. Two RAC nodes

Configurations with Three RAC Nodes

Configurations with three RAC nodes have 3-way data mirroring using high-redundancy ASM disk groups. However, a

normal-redundancy ASM disk group is used for clusterware files (the GRID disk group). Such a cluster can tolerate the loss of any one node without database downtime. However, the loss of a second node will result in downtime. The main reasons for using three (vs. two) RAC nodes are the additional CPU and memory resources, and the additional storage read bandwidth.

It is possible to have a 3-node RAC cluster that can tolerate the loss of two RAC nodes without database downtime, but such configurations are beyond the scope of this article.

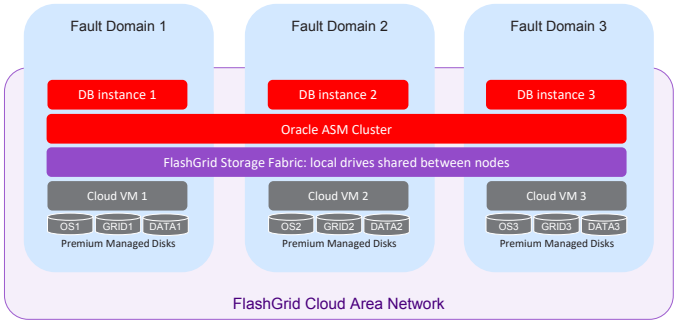


Figure 2. Three RAC nodes

Architecture Highlights

- FlashGrid Cloud Area Network (CLAN) enables high-speed overlay networks with multicast, virtual IP addresses, and bandwidth allocations.
- FlashGrid Storage Fabric turns local drives (e.g., Premium SSDs) into shared drives accessible from all nodes in the cluster.
- FlashGrid Read-Local™ Technology minimizes network overhead by serving reads from local drives.
- The technology allows 2-way or 3-way mirroring of data across separate nodes or Availability Zones.
- Oracle ASM and Clusterware provide data protection and availability.

Network

FlashGrid Cloud Area Network enables high-speed clustered applications to be run in public clouds or multi-datacenter environments with the efficiency and control of a Local Area Network.

The network connecting Azure VMs is effectively a single IP network with a fixed amount of network bandwidth allocated per VM for all types of network traffic. However, the Oracle RAC architecture requires separate networks for client connectivity and for the private cluster interconnect between the cluster nodes. There are two main reasons for that: 1) the cluster interconnect must have low latency and sufficient bandwidth to ensure adequate performance of the inter-node locking and Cache Fusion, and 2) the cluster interconnect is used for transmitting raw data, and for security reasons must be accessible by the database nodes only. Also, Oracle RAC requires networks with multicast capability, which is not available in Azure.

FlashGrid CLAN addresses the limitations described above by creating a set of high-speed virtual LAN networks and ensuring QoS between them.

Network capabilities enabled by FlashGrid CLAN for Oracle RAC in Azure include the following:

- Each type of traffic has its own virtual LAN with a separate virtual NIC, e.g., *fg-pub*, *fg-priv*, and *fg-storage*

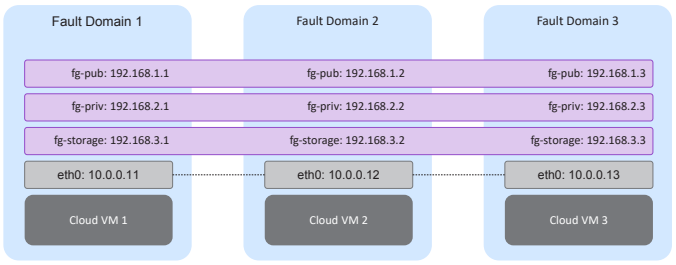


Figure 3. CLAN virtual subnets

- Minimum guaranteed bandwidth allocation for each traffic type while accommodating traffic bursts
- Low latency of the cluster interconnect in the presence of large volumes of traffic of other types
- Transparent virtual IP failover between nodes
- Multicast support

Shared Storage

FlashGrid Storage Fabric turns local drives into shared drives accessible from all nodes in the cluster. The local drives shared with FlashGrid Storage Fabric can be block devices of any type, including Premium SSD disks or LVM volumes. The sharing is done at the block level with concurrent access from all nodes.

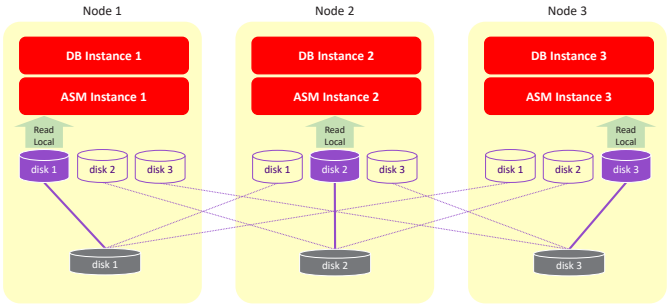


Figure 4. FlashGrid Storage Fabric with FlashGrid Read-Local Technology

Each database node has a full copy of user data stored on Azure Premium SSD disks attached to that database node. The FlashGrid Read-Local Technology allows serving all read I/O from the locally attached disks and increases both read and write I/O performance. Read requests avoid the extra network hop, thus reducing the latency and amount of the network traffic. As a result, more network bandwidth is available for the write I/O traffic.

The FlashGrid software maintains persistent disk names and sets the required disk permissions. There is no need to configure ASMlib or UDEV rules.

ASM Disk Group Structure and Data Mirroring

FlashGrid software leverages proven Oracle ASM capabilities for disk group management, data mirroring, and high availability. In Normal Redundancy mode each block of data has two mirrored copies. In High Redundancy mode each block of data has three mirrored copies. Each ASM disk group is divided into failure groups—one failure group per node. Each disk is configured to be a part of a failure group that corresponds to the node where the disk is located. ASM stores mirrored copies of each block in different failure groups.

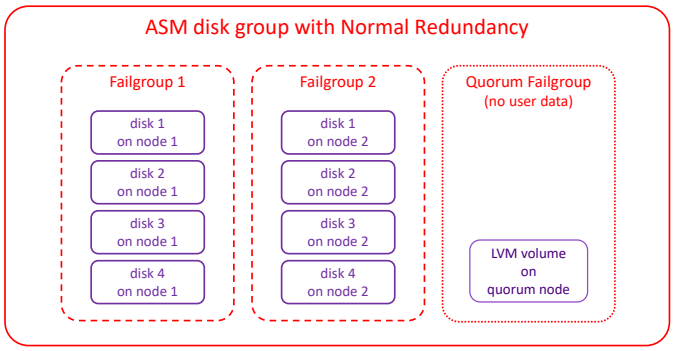


Figure 5. Example of a normal redundancy disk group in a 2-node RAC cluster

A typical Oracle RAC setup in Azure will have three Oracle ASM disk groups: GRID, DATA, and FRA.

In a 2-node RAC cluster all disk groups must have normal redundancy. The GRID disk group containing voting files is required to have a quorum disk for storing a third copy of the voting files. Other disk groups also benefit from having quorum disks, as they store a third copy of ASM metadata and improve failure handling.

In a 3-node cluster all disk groups except the GRID disk group must have high redundancy in order to enable full Read-Local capability. In a 3-node RAC cluster the GRID disk group would typically have normal redundancy. Note that in such 3-node RAC clusters, loss of no more than one node is tolerated without causing downtime.

If a 3-node RAC cluster must tolerate simultaneous failure of two nodes without causing downtime, then the GRID disk group must have high redundancy, and an additional two quorum nodes must be provisioned to accommodate five copies of voting files. Details of such configuration are not covered in this article.

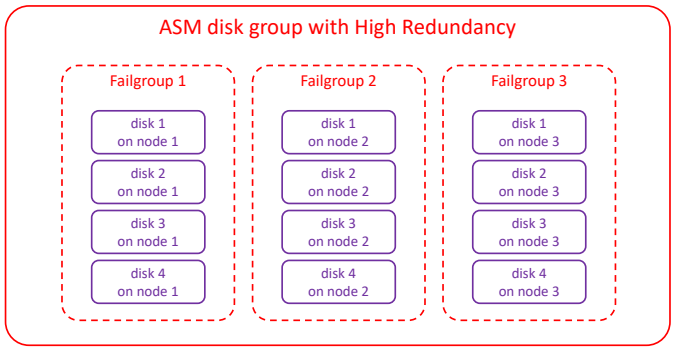


Figure 6. Example of a high redundancy disk group in a 3-node RAC cluster

Fault Domains and Update Domains

FlashGrid Storage Fabric and FlashGrid Cloud Area Network have a fully distributed architecture with no single point of failure. The architecture leverages HA capabilities built in Oracle Clusterware, ASM, and Database.

In order to protect a cluster from two VMs going offline simultaneously, it is important to place all VMs of the cluster in an a*vailability set* and spread the VMs across separate *fault domains* and *update domains*.

Placing VMs in separate fault domains ensures that those VMs have separate power sources and network switches. Thus, failure of a power source or a network switch will be localized to a single fault domain and will not affect VMs in other fault domains.

Placing VMs in separate update domains ensures that those VMs will not be rebooted simultaneously during a planned update of the underlying Azure infrastructure.

Because all instances are virtual, failure of a physical host causes only a short outage for the affected node. The node instance will automatically restart on another physical host. This significantly reduces the risk of double failures.

Note that not all regions provide three fault domains. It is still possible to deploy 2-node clusters in the regions that provide only two fault domains by placing the quorum VM in a different region. Details of such configurations are beyond the scope of this white paper.

Data Availability

A Premium SSD disk in Azure provides persistent storage that survives a failure of the node VM. After the failed VM restarts on a new physical node, all its volumes are attached with no data loss.

Premium SSD disks have built-in redundancy that protects data from failures of the underlying physical media. The mirroring by ASM is done on top of the built-in protection of Premium SSD disks. Together Premium SSD disks plus ASM mirroring provide durable storage with two layers of data protection, which exceeds the typical level of data protection in on-premises deployments.

Supported VM Types and Sizes

Database node VMs must have 2+ CPU cores, 8+ GB of memory, and Premium storage support. The following VM types are recommended for database nodes when using Enterprise Edition:

- DS11_V2, DS12_V2, DS13_V2, DS14_V2, DS15_V2
- GS1, GS2, GS3, GS4, GS5

DS11_V2 (2 cores, 14 GB memory) type or GS1 (2 cores, 14 GB memory) type is recommended for running the Standard Edition 2 database.

DS1_V2 (1 core, 3.5GB memory) type is recommended for use as a quorum node. Note that there is no Oracle Database software installed on the quorum node.

Supported Disk Types

Currently only Managed Premium SSD disks are supported.

Each disk provides up to 5,000 IOPS, depending on its size. The maximum performance of 5,000 IOPS is available for disk sizes of 513 GB or larger. For databases that require high performance but smaller capacity, use of multiple 513 GB disks may be optimal to maximize the total number of IOPS. Note that the maximum number of IOPS per VM is also capped and depends on the VM size: 64,000 for DS15_V2.

Support of local SSDs is planned in future.

Reference Performance Results

The main performance-related concern when moving database workloads to the cloud tends to be around storage and network I/O performance. There is a very small to zero overhead related to the CPU performance between bare-metal and the Azure cloud. Therefore, in this paper we focus on the I/O performance.

Calibrate_IO

The CALIBRATE_IO procedure provides an easy way to measure storage performance, including maximum bandwidth, random IOPS, and latency. The CALIBRATE_IO procedure generates I/O through the database stack on actual database files. The test is read-only and it is safe to run it on any existing database. It is also a good tool for directly comparing the performance of two storage systems because the CALIBRATE_IO results do not depend on any non-storage factors, such as memory size or the number of CPU cores.

Test configuration:

- Two database nodes, DS15_V2
- Sixteen 513 GB Premium SSD disks per node

Test script:

```
SET SERVEROUTPUT ON;
DECLARE
  lat INTEGER;
  iops INTEGER;
  mbps INTEGER;
BEGIN DBMS_RESOURCE_MANAGER.CALIBRATE_IO (32, 10, iops, mbps, lat);
DBMS_OUTPUT.PUT_LINE ('max_iops = ' || iops);
DBMS_OUTPUT.PUT_LINE ('latency = ' || lat);
DBMS_OUTPUT.PUT_LINE ('max_mbps = ' || mbps);
end;
/
```

Our results:

```
Max_IOPS = 121597
Latency = 0
Max_MB/s = 1348
```

SLOB

SLOB is a popular tool for generating I/O-intensive Oracle workloads. SLOB generates database SELECTs and UPDATEs with minimal computational overhead. It complements Calibrate_IO by generating mixed (read+write) I/O loads. AWR reports generated during the SLOB test runs provide various performance metrics. For the purposes of this paper we focus on the I/O performance numbers.

Test configuration:

- Database node VM type: DS15_V2
- Sixteen 513 GB Premium SSD disks per node
- SGA size: 2.6 GB (small size selected to minimize caching effects and maximize physical I/O)
- 8 KB database block size
- Schemas: 45 x 240 MB per node
- UPDATE_PCT= 20

The table below shows our results for tests performed in the same configuration (provided above) with a 2-node RAC cluster and a single-instance database (on a single VM) as a baseline.

Note that the results are very similar for a 2-node RAC and for a single-instance database. While there is an extra network overhead in the 2-node RAC for writes and global cache operations, the additional storage read bandwidth of the second node increases the overall performance and helps keep it on par with the single-instance configuration. At 53 K IOPS the performance is comparable to using a storage array with SSDs.

	2-node (both nodes combined	Single-instance
Read+Write Database Requests (IOPS)	53,839	52,990
Read Database Requests (IOPS)	43,020	43,159
Write Database Requests (IOPS)	10,819	9,831

Database Client Connectivity

There are two options for connecting clients (including application servers) to the RAC database nodes:

- Database clients that have *fg-pub* CLAN subnet configured can connect via SCAN address. In case a database node goes down, the SCAN VIPs move to the surviving database node(s). This allows instantaneous and transparent client connectivity to the surviving database node(s). This option requires FlashGrid CLAN software installation on the database client system.
- Database clients that do not have CLAN configured can connect to the database nodes via their Azure VNET private IP addresses. This option allows higher degree of isolation between the client and database nodes by placing them in different network subnets or security groups and enabling a firewall between them.

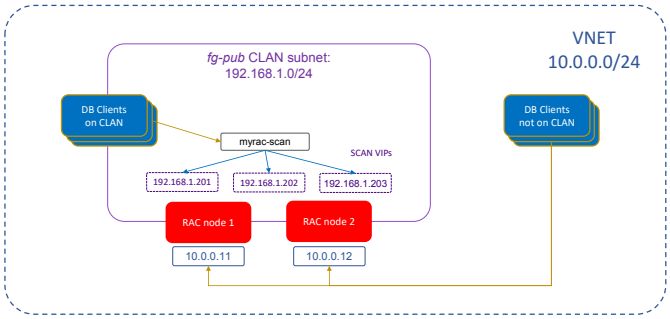


Figure 7. Database client connectivity

Software Compatibility

The following versions of software are currently supported:

- Database: 12.2.0.1, 12.1.0.2 or 11.2.0.4 with latest PSU
- Grid Infrastructure: 12.2.0.1 or 12.1.0.2 with the latest PSU
- OS: Oracle Linux 7.3, RHEL 7.3, or CentOS 7.3
- FlashGrid software: ver. 17.05

Deployment Process

Below is a brief overview of the steps for provisioning of an Oracle RAC cluster in Azure.

1. Provision three VMs using Resource Manager.
2. Attach the required number of Managed Premium SSD disks.
3. Download Oracle installation files to the database nodes.
4. On each node run the OS configuration script provided by FlashGrid.
5. On each node run the automated cluster configuration script provided by FlashGrid.
6. Create a database using DBCA.

Conclusion

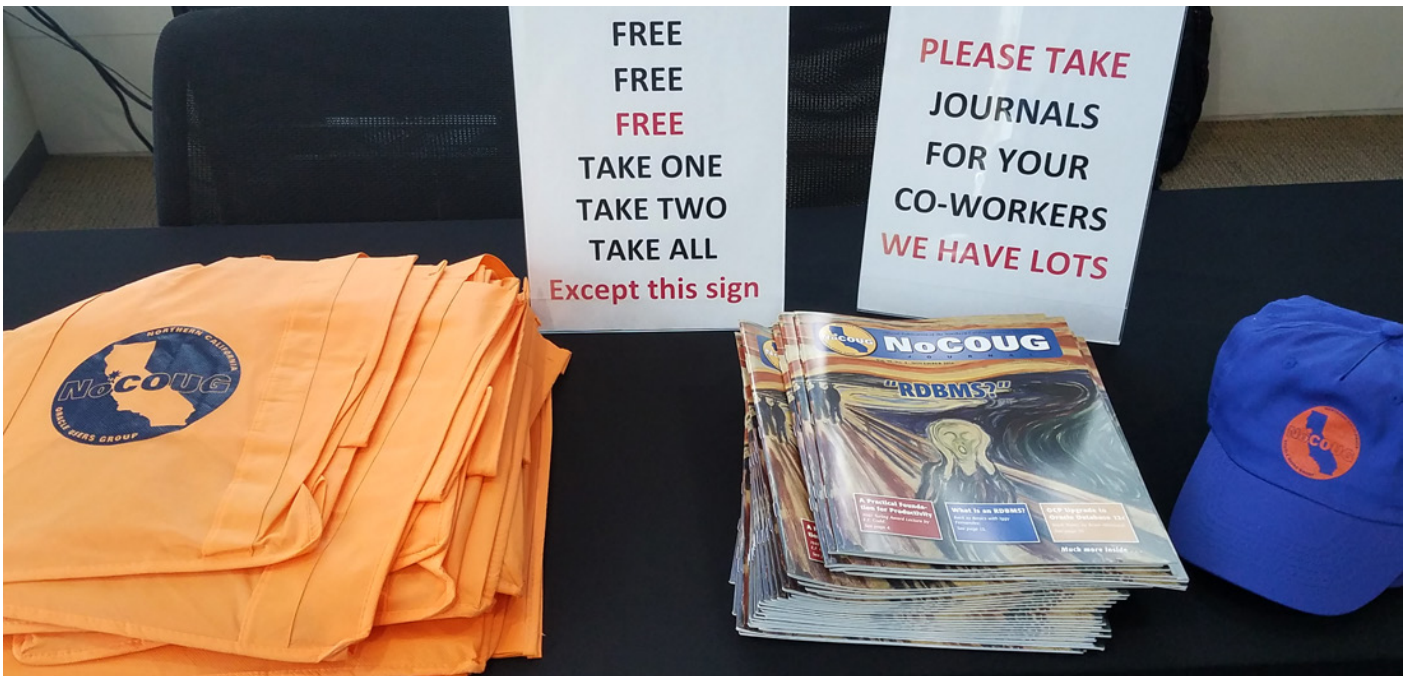
Running Oracle RAC clusters in the Microsoft Azure cloud has historically been challenging due to storage and network constraints. FlashGrid Cloud Area Network and FlashGrid Storage Fabric remove those constraints and enable Oracle RAC clusters with high-availability characteristics exceeding those of the traditional on-premises deployments. ▲

Artem Danielov is CTO at FlashGrid.

30th Anniversary Celebration
Sponsored by Google Cloud Platform

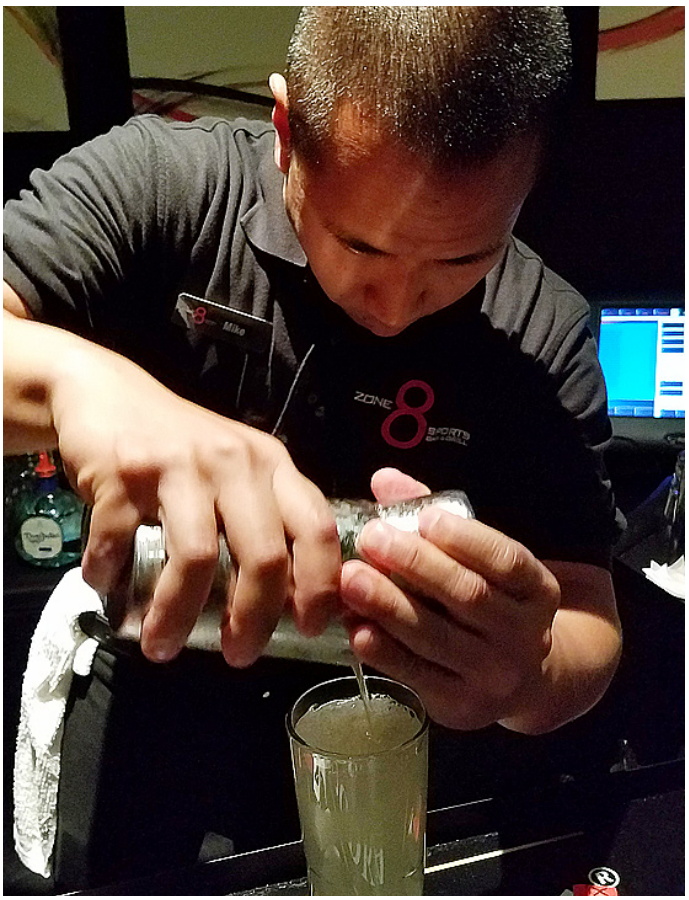
Happiness is a breakfast donut.

Database Specialists is now TriCore Solutions.

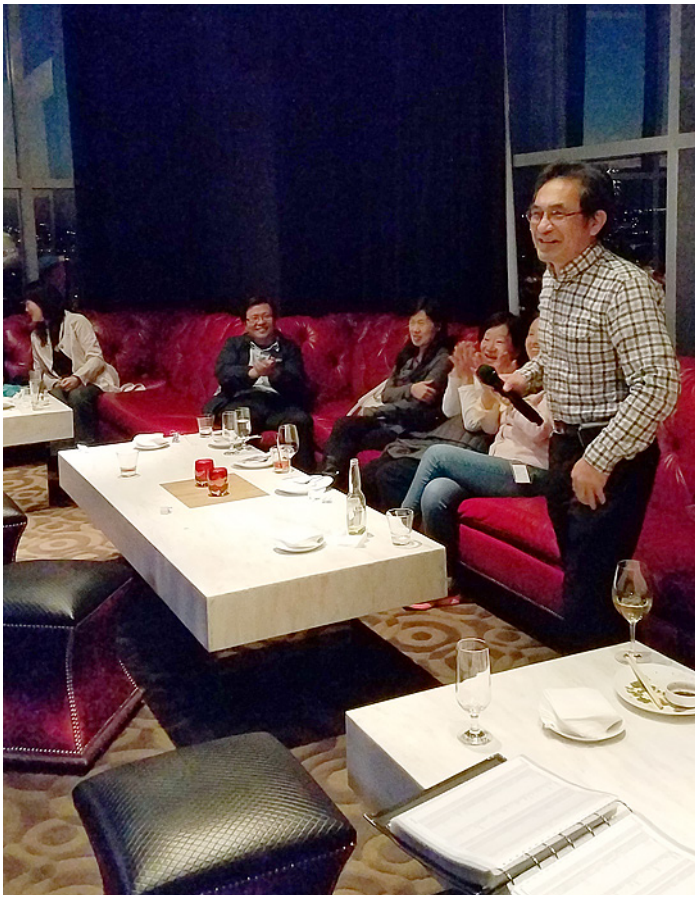


Post-conference celebration at Casino M8trix.

Included with your NoCOUG membership.

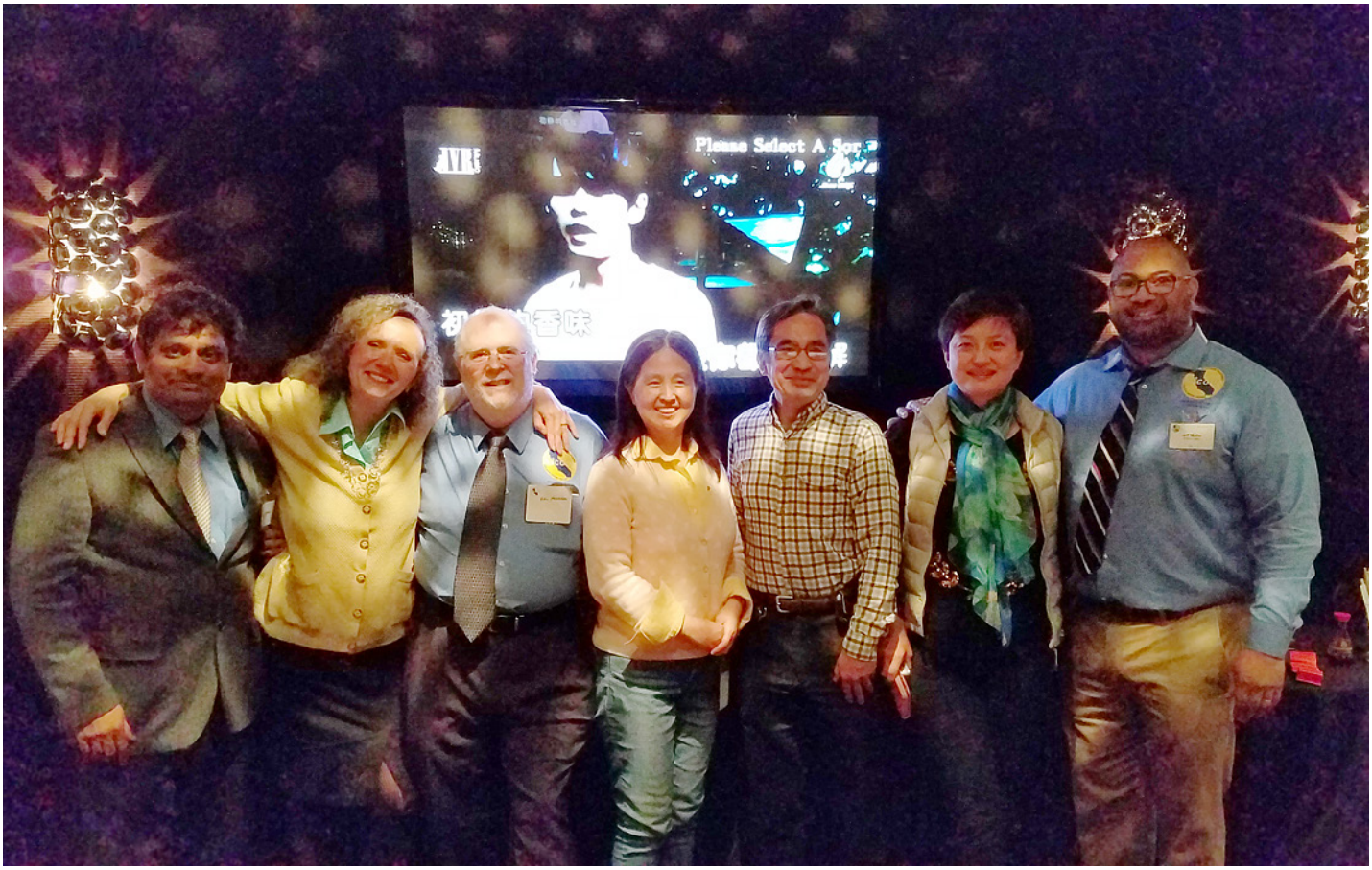


Also included with your NoCOUG membership.



Knock three times on the ceiling if you want me.

First-ever life-time member of NoCOUG.



Golden voices.

NoCOUG
P.O. Box 3282
Danville, CA 94526

RETURN SERVICE REQUESTED