#PTK

PASS THE KNOWLEDGE

INDEPENDENT
UKOUG
UK ORACLE USER GROUP

WWW.UKOUG.ORG

AUTUMN 2019 / ISSUE 71

FORMERLY ORACLE SCENE

TECH EDITION

TURN OVER FOR BUSINESS APPS

ORACLE TABLE SCANS

How to improve performance and release space

OCI + AZURE = ???

What could the Oracle/Microsoft cloud partnership mean for you?

The compute cloud performance SHOWDOWN

ORACLE v AWS v IBM v GOOGLE v MICROSOFT – WHO'LL COME OUT ON TOP IN THE IAAS STAKES?

TECHFEST 19

1-4 DECEMBER 2019 // THE GRAND BRIGHTON // UK

Be part of the most important tech event of the year

Our major annual technology conference, **Techfest19**, will take place over four incredible days in Brighton, at the famous Grand Hotel – so make sure you don't miss out.

This is your chance to rub shoulders with some of the world's leading Oracle experts, take a deep dive into the technology that you care most about, and if you've something to share, **#passtheknowledge.**

To view the agenda and book your place, go to:
ukoug.org/page/techfest19

INDEPENDENT
UKOUG
UK ORACLE USER GROUP

#PTK Tech | Autumn 2019

IN THIS ISSUE

PLUS

FLIP OVER →

FOR BUSINESS APPS!

Flip over to check out the Business Apps side of this issue. You'll find an in-depth examination of the strategic thinking behind the move to cloud (page 14), a debate on the real value of AI (page 30), plus lots more from the Oracle community.

#PTKOnline

View the latest edition online and access the archive of *#PTK* and *Oracle Scene* editions here: **ukoug.org/ptk**

Welcome to the new issue of #PTK Pass The Knowledge

Count them: how many times have you heard, over the past decade, that cloud changes everything? If you're anything like me, it'll be so often that the words have almost become an empty cliché.

Yet, like all clichés, it's rooted in truth. Since cloud first became a serious business technology, it has demonstrated time and again – for better or for worse – that it's truly transformative. It disrupts the way businesses function, the way they interact with customers, the way they innovate; it changes the role of the IT department, pricing models, relationships with vendors; and it transforms the vendors themselves – as demonstrated by the recent strides Oracle has been making in the cloud space, after its slow start.

Yet, even now, after so many years of scrutiny, discussion and success stories, only 42% of UK enterprises used some form of cloud service in 2018 (according to research by the EU). That makes me believe that in many organisations there is still a high level of uncertainty and woolly thinking, coupled with corporate inertia, that surrounds the move to off-premise IT.

For this reason, we've decided to devote a lot of this issue – in both the Tech and Business Apps halves – to cloud, where we hope you'll find the knowledge and insight necessary to help you make the best decisions for your organisation, wherever you are on your 'cloud journey'.

Our cover story, ***The compute cloud performance showdown* (page 12)**, compares five of the world's major cloud providers, so you'll be able to find some degree of certainty on how Oracle stacks up against its rivals. We also take a look at what benefits will stem from **Oracle's cloud interoperability partnership with Microsoft (page 9)**.

Plus, over on the Business Apps side, you can read our deep dive into the strategic thinking that needs to lie behind any move to cloud ***(Finding cloud certainty*, page 14)** and our case study detailing how **data analytics in the cloud helped the NHS identify £1.2bn of savings (page 22)**.

I'd also like to thank all our Tech contributors who've taken the time to share their knowledge with the rest of the Oracle user community via the articles in this issue. If, like them, you have a useful experience to impart, or some helpful insights that will benefit others in the Oracle world, please don't hesitate to let us know. I guarantee our expert editors will make it easy for you, even if you're not a confident writer. Please send your ideas, proposals and comments to **editor@ukoug.org** – and you'll find our helpful contributors' guidelines at: **ukoug.org/ptk**.

Enjoy the issue

Andy Nellis
Editor

About the Editor
Andy Nellis, founder of e-Resolve, and more recently Managing Director of Cognition24, has worked in the world of ERP for over 20 years. His focus is on enabling business transformation through people, process and culture underpinned by the right technology.

#PTK: PASS THE KNOWLEDGE
EDITORIAL TEAM

Editor
Andy Nellis
editor@ukoug.org

Consulting Editor
James Lawrence
james.lawrence@topdogcomms.co.uk

Art Director
Peter Allen
peter@peterallendesign.co.uk

UKOUG contact
Tania Huntington
tania.huntington@ukoug.org

Advertising
Kerry Stuart
kerry.stuart@ukoug.org

Membership
Jordan Osborne
jordan.osborne@ukoug.org

UKOUG governance
A full listing of Board members, along with details of how the user group is governed, can be found at: *ukoug.org/meettheteam*

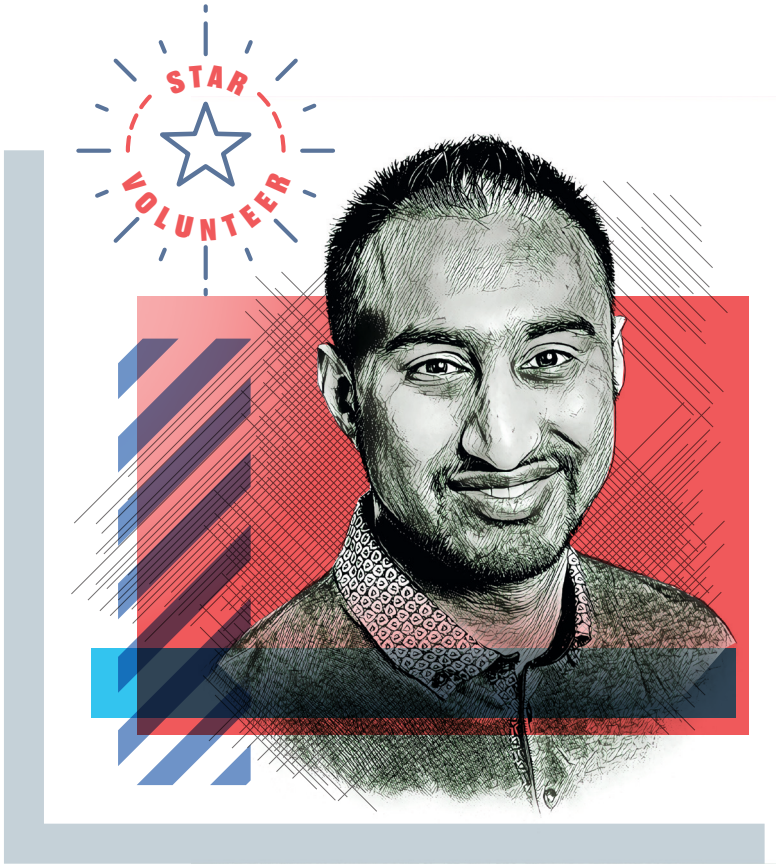
UKOUG office
19-23 High St,
Kingston upon Thames,
London KT1 1LL
Tel: +44 (0)20 8545 9670
info@ukoug.org
www.ukoug.org

Produced by
Top Dog Communications Ltd
Tel: +44 (0)7913 045917
james.lawrence@topdogcomms.co.uk
www.topdogcomms.co.uk

Designed by Peter Allen Design
peter@peterallendesign.co.uk
www.peterallendesign.co.uk

ME & the UKOUG

Zahid Anwar from Version 1 explains how volunteering for the UKOUG has played an important role in his career development



I first decided to present at a UKOUG event because I had valuable insight into how to set up Oracle 12c Cloud Control in a maximum availability architecture. I thought it would be great to share that with customers and the wider Oracle community, so I presented with Telefónica, an existing customer, at UKOUG Scotland in 2015. After that, I got the bug!

Since overcoming my initial nerves, I've really enjoyed presenting at the events and have found it very satisfying. I've done it nine times now and highly recommend it. Sharing my insights and experiences and giving back to the community gives me a real sense of accomplishment.

Presenting led me to start volunteering for various committees. I've been part of the UKOUG Partner of the Year committee for two years now, as well as being on the committee for Tech 18 last year and Techfest this year. But my most significant contribution so far was being the project lead for the Southern Tech committee.

I gained a good insight into how much hard work and effort is required to put on a great conference when I volunteered for the Tech 18 committee. You don't appreciate it until you've been involved in what goes on in the background, from the kick-off meeting, to judging papers, the agenda planning day, the actual conference... it's hard work, but it's rewarding – and you get to work with a great bunch of people too!

> Sharing my insights and giving back to the community gives me a sense of accomplishment

I first went to Oracle OpenWorld in 2015, the same year I did my first UKOUG presentation. The trip was a reward from my employer for receiving my Oracle Certified Master qualification. I was fortunate to meet Tim Hall, who's one of the top Oracle bloggers (oracle-base.com). I explained to him that I had presented at the UKOUG and was keen to also start blogging but wasn't that confident. His advice was to continue to take part in presentations at the UKOUG and to start writing some simple posts. It took nearly two years, but in 2017 I kicked off my blog (zeddba.com) and that's when I started ramping up my presentations too. Ultimately, this led to the great news that I had gained recognition as an Oracle ACE, an achievement I'm really proud of.

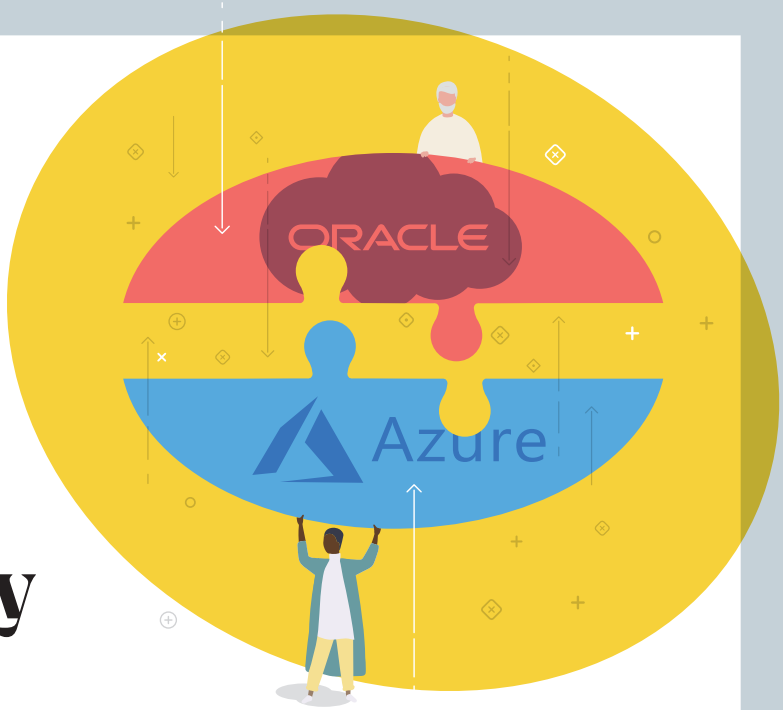
Presenting and blogging have definitely helped me to build my career and gain recognition at work – and Version 1 has always been very supportive. My journey with the UKOUG and sharing my insights has taken me out of my shell and been an important part of my journey from a DBA to a Consultant. Gaining this experience, and achieving my Oracle ACE, adds credibility to who I am as a Consultant and what I do for Version 1's enterprise customers across the UK and Ireland.

If you're not sure about whether to volunteer or not... just go for it! Don't forget, it's just like learning any new skill. Give it a go, don't go straight in at the deep end – and at least you'll know you tried!

Zahid Anwar, an Oracle Certified Master and Oracle ACE, is a Principal Consultant at Version 1, an Oracle Platinum Partner, where he has worked since 2014.

TO LEARN MORE about volunteering for UKOUG and its benefits, go to: ukoug.org/page/volunteersarea

NEWS ETC...



OCI/Azure partnership set to reduce cloud complexity

One of the biggest stories concerning Oracle in recent months has been the announcement of its cloud interoperability partnership with Microsoft. The OCI/Azure link-up includes the following key aspects:

- A direct interconnection between the two clouds, starting in North America and then rolling out to the rest of the world.
- Unified identity access management and a single sign-on for both clouds.
- Supported deployment of custom applications and packaged Oracle applications on Azure with Oracle databases deployed in Oracle Cloud, while the same Oracle applications will also be certified to run on Azure with Oracle databases in Oracle Cloud.
- A "collaborative support model" for customers wanting to take advantage of this new capability.

Many industry experts are greeting the announcement with cautious optimism, particularly as enterprises are moving increasingly towards multi-cloud environments. Key business advantages are most likely to stem from the reduced complexity and simpler migration opportunities the partnership offers.

For an analysis of where the biggest benefits could lie, see our article on page 9 of this issue.
And you can read Oracle's take on the partnership here: *www.oracle.com/uk/cloud/oci-azure.html*

DON'T MISS OUR PARTNER OF THE YEAR AWARDS

The votes are in, the shortlists have been announced and the trophies are being polished in preparation for 2019's Partner of the Year awards ceremony.

The hotly-anticipated event will take place on 3 October at the Kimpton Fitzroy Hotel in London.

To view the shortlists and book tickets, go to: ukoug.org/page/pya

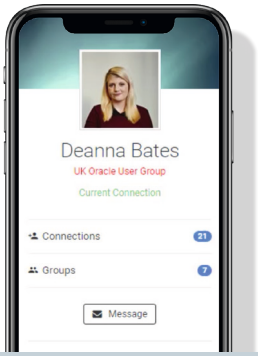
ON THE OTHER SIDE...

When you've finished reading this side of the magazine, try flipping it over to get up to date with some of the latest news, information and insights from the Oracle Business Apps community and beyond. You'll find an in-depth examination of the strategic thinking behind cloud, plus lots more. We've also included handy 10-second summaries of the features in case you're in a hurry.

Take us with you wherever you go

SocialLink is a great way to get even more from your UKOUG membership. By signing up on our website or via the SocialLink mobile app, you can stay connected to other Oracle users, wherever and whenever you like. It allows you to instantly access member benefits at any time, message a host of Oracle experts and share knowledge and thought-leadership with the rest of the community on the go.

To find out more, go to: ukoug.org/page/sociallink



UKOUG EVENTS

Sharing knowledge, thought-leadership and expertise at our many events is one of the most valuable things about being a UKOUG member. Here are some key dates for your diary...

FIND OUT MORE

For all the latest information on UKOUG events, and to book your place, visit our website: ***ukoug.org/programmes2019***

19 SEPTEMBER

Higher Education Forum (London)

3 OCTOBER

Partner of the Year Awards (London)

8 OCTOBER

Analytics Modernisation Summit (London)

5 NOVEMBER

Security & Compliance (London)

6 NOVEMBER

Cloud Applications Experience (London)

12 NOVEMBER

JD Edwards Customer Day (London)

1-4 DECEMBER

Techfest19 (Brighton)

15-16 JUNE, 2020

BAX 2020, the Business Applications Exchange (London)

Coming soon!

The 2020 events calendar

Check out our website to see all next year's events in one place: ***ukoug.org/programmes2020***

UKOUG MEETUPS

We run many informal Meetups throughout the year, on topics ranging from cloud technology to blockchain. To find out more, and get alerts sent to your inbox, go to: ***www.meetup.com/UK-Oracle-User-Group-Meetup***

The best things about being a UKOUG member

We polled our members to find out what they love most about the UK Oracle User Group. Here are the five most common reasons you gave for joining up and staying with us...

To network with other users and Oracle experts

To find solutions to Oracle-related problems

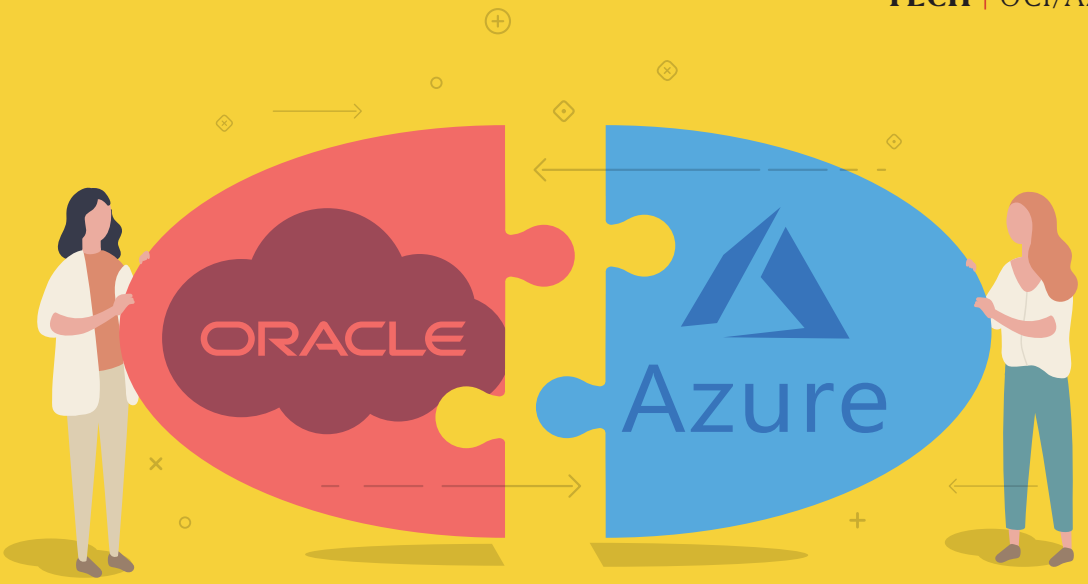
To develop and raise their professional profile

To keep up-to-date with Oracle information – from an independent voice

To connect with Oracle partners

If you're not a UKOUG member and would like to join – and benefit from all of the above and more – or if you have any queries about your membership, please get in touch with Jordan Osborne at: **jordan.osborne@ukoug.org**

INDEPENDENT
UKOUG
UK ORACLE USER GROUP



The technical impact of the ORACLE/MICROSOFT cloud partnership

The announcement of the OCI/Azure interoperability deal made big headlines right across the tech industry. But what could this mean in reality for your organisation?

By Philip Brown

If you're new to the cloud then you may see Oracle's recent announcement of its 'cloud interoperability partnership' with Microsoft's Azure (see News, page 6) as a removal of technical barriers that will enable true multi-cloud architectures. But if you've been working with cloud for some time, you might think… hold on, I can do this already!

The Oracle and Microsoft partnership is a genuinely interesting development and a first within the industry. However, the immediately obvious possibility of connecting two clouds together isn't actually where the true benefit lies, in my opinion. Here, I'll drill into some of the detail and explain why.

10-SECOND SUMMARY

- **The ability to connect clouds is not a new thing – but it has often created complexity and performance issues.**
- **The biggest upside of the OCI/Azure partnership is most likely to be its Unified Cloud Management.**
- **As multi-cloud environments become more commonplace, the partnership is likely to help simplify connections, creating real user benefits.**

CONNECTING CLOUDS

If you wanted to connect two clouds together – be it AWS, Oracle, Azure or GCP – you could already do this. Let's start at the beginning: with an on-premises-to-cloud connection, cloud providers offer a site-to-site VPN-as-a-Service (VPNaaS) connection. This allows you to deploy a VPN endpoint in your cloud virtual network then specify the IP ranges and routing to enable traffic to flow between your on-premises site and the cloud. For all cloud providers, the cloud side of the

VPN is nothing more than specifying a few settings in the UI and then completing the configuration of your on-premises VPN endpoint / firewall normally administered by the network team (see Figure 1).

Networking teams see VPNs as a security minimum if they are to connect an on-premises network to the cloud network – and it's always been very easy to set up. All cloud providers' VPNs have the ability to support different Phase 1 and Phase 2 configurations which means that *most* on-premises endpoints have a reasonable chance of getting connected, unless you're using really old or obscure configurations.

The challenge with VPNaaS is that you can't connect one VPNaaS to another. VPNaaS works primarily due to the fact that the main configuration is done on your on-premises equipment – which means you're very limited as to what you can specify in the VPNaaS configuration and the connection is established once you've added the VPNaaS settings / IPs into your on-premises equipment. Therefore connecting Azure and Oracle together using the VPNaaS is impossible as they're both expecting the bulk of the configuration to be done on 'the other side' as one VPNaaS is not designed to work against another VPNaaS.

The typical way to connect two clouds together is to use one side's VPNaaS and then, on the other side, to create an IaaS Virtual Machine and deploy something like StrongSwan to act as the VPN endpoint (see Figure 2). Deploying a software VPN in IaaS enables you to have more control of the configuration and hence is something akin to having your own VPN appliance.

The challenge with this approach is that, now you have an IaaS instance that you need to manage, you potentially need to make it high-availability, plus there are all the other issues of patching and maintenance which come into play.

Also, this connection is across the internet which means it may suffer from latency problems and not deliver a consistent level of performance. This solution works, but if performance levels are important, you will need to make use of dedicated connections.

> Since most enterprises are running Oracle and Microsoft workloads, this partnership will be of great benefit to them

Figure 1: Connectivity from on-premises to Oracle Cloud

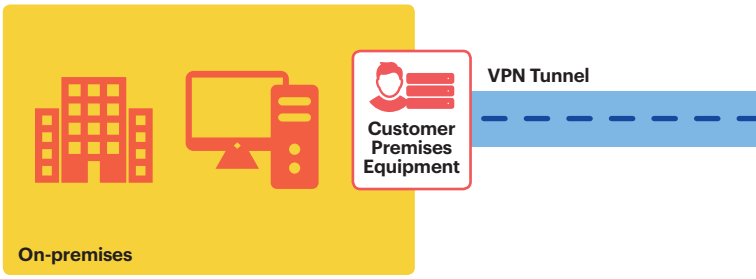


Figure 2: How you would historically connect another cloud to Oracle Cloud

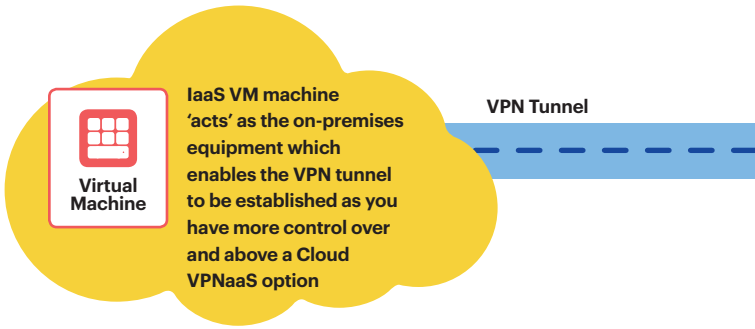
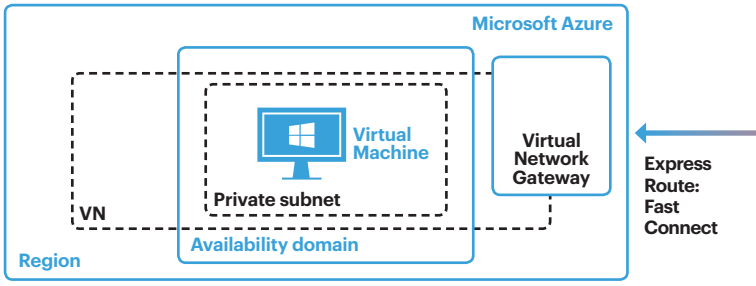
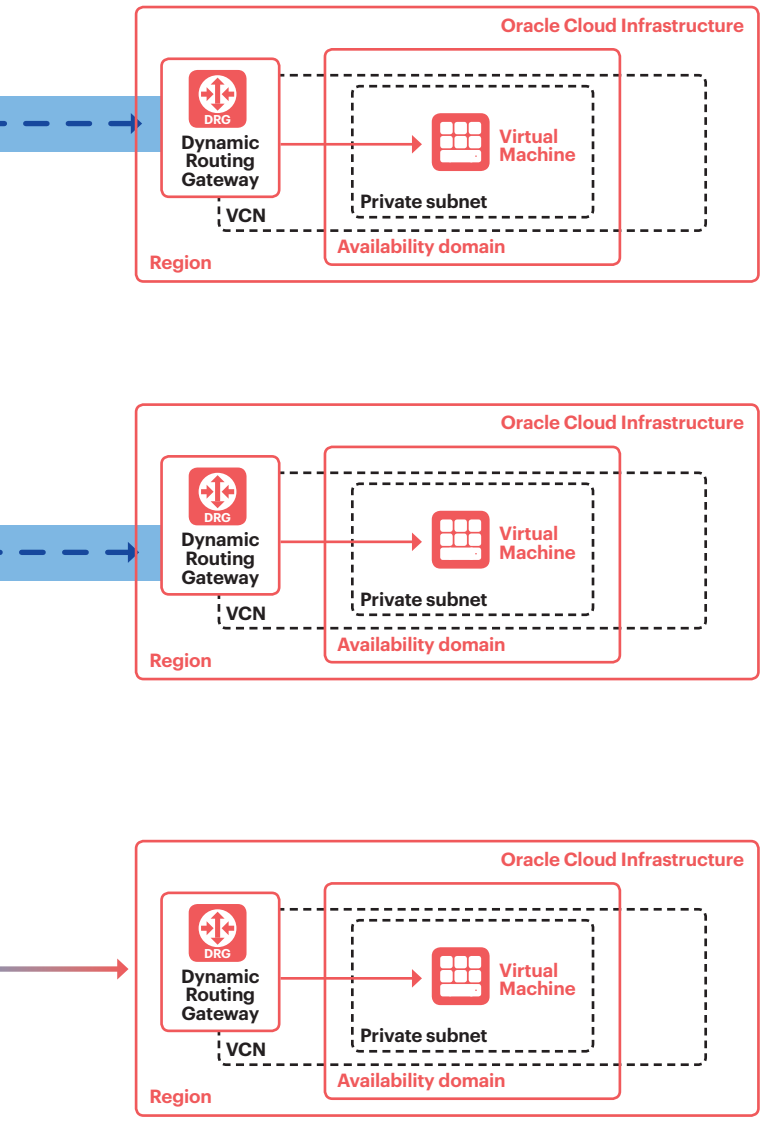


Figure 3: Direct connectivity between clouds using Express Route and Fast Connect



Dedicated cloud connections have been around as long as VPNs, and like VPNs they are relatively easy to set up, but normally require some additional plumbing between your on-premises data centres and the cloud provider's. A recent innovation, however, has been the dedicated inter-cloud connectivity provided by companies like Megaport and Equinix.

At the same time, cloud exchanges have been providing the ability to link two dedicated cloud connections from two different cloud providers – so, for example, you could have your Azure Express Route linked directly to your Oracle Fast Connect. The great thing about this is that the traffic between the two clouds doesn't go via your on-premises network and there isn't any additional plumbing to be configured as the cloud exchanges already have the connections in place. I've seen this configured in real time and it's impressive.



It's also worth pointing out that this inter-cloud connectivity using cloud exchange *isn't* like a VPN solution in the sense that traffic isn't encrypted. It's a dedicated private connection so it's secure, but if you require encryption from a compliance perspective then a dedicated connection alone will not provide it. The same is true for using Express Route or Direct Connect from on-premises to the Cloud – they are dedicated connections and not VPNs.

IS THE PARTNERSHIP JUST HYPE?

So, if you could do this before with VPNs, or if I can do this now with cloud exchanges, then surely this partnership is no more than hype? Well… not really. The enabling of inter-cloud connectivity without a cloud exchange is likely to lead to a lower total cost of ownership for that connection, which is good. But, what's more, it will be a fully-managed and highly-available connection that doesn't require complex networking between multi-cloud deployments (see Figure 3).

The connectivity of clouds is great, of course, and anything that reduces the cost and complexity can only be seen as a benefit. However, it really gets interesting when you can manage both Azure and Oracle from a single cloud UI.

Unified Identity and Access Management (IAM) seems like the least exciting element of this partnership, but in reality it could turn out to be the most important. This is because cloud security – i.e. the security of the cloud console and its users – will always be a challenge in a multi-cloud environment. You will have two UIs which gives you two different users who then require two different security policies which have different policy language to configure. The problem this creates is either one of laziness or complexity.

Laziness simply means that it's easier to administer your multi-cloud environment with two super administrators, which isn't great. This goes against the principle of least privilege, and super admins in a cloud console can ultimately affect the security of the resources running in the cloud (think ingress and egress rules for VM access). Therefore, this is not an approach we would want to endorse.

Complexity means you need to create lots of users and policies to manage cloud security across these two cloud portals. This fixes the issue of least privilege but the resulting complexity can lead to mistakes and misconfiguration.

The partnership talks about the future capabilities of unified IAM and the ability to control both Azure and Oracle resources from a single pane of glass. There are increasingly blurred lines on corporate and cloud infrastructure which has led to the switch from perimeter network security to asset security – so anything that provides simplicity must be seen as a massive advantage.

Oracle have also stated that there will be a joint support model, which again points towards creating additional simplicity in a multi-cloud environment.

Since most enterprise organisations are running both Oracle and Microsoft workloads, this partnership is likely to be of great benefit to them. At the end of the day, it boils down to simplicity – in terms of the networking connectivity but, more importantly, the management of what's only going to become a more commonplace scenario in the coming years: multi-cloud environments.

ABOUT THE AUTHOR

Philip Brown is head of Oracle Cloud Services at DSP. He's an Oracle ACE and has been presenting and writing articles for the technical community for over 10 years.

The compute cloud performance SHOW

Put to the test: the performance of the big five cloud providers – Amazon

10-SECOND SUMMARY

- Attain compared the top five cloud providers, focusing on application server and database server performance running on provisioned virtual machines to give an indication of overall performance.
- Unsurprisingly, the results indicated that more powerful CPUs tended to perform better, regardless of the cloud provider (with a few anomalies).
- AWS came out on top, but with Oracle close behind as a serious contender – particularly from a cost and performance point of view.

DOWN

By Ahmed Aboulnaga

Web Services, Oracle Cloud, IBM Cloud, Google Cloud, and Microsoft Azure

Most cloud providers will claim that the performance of their infrastructure is ahead of the competition, but there are no independently tested or published findings comparing the performance among the leading providers. For this reason, we conducted a series of objective tests to compare Infrastructure as a Service (IaaS) performance – and specifically compute cloud – across the five leading cloud providers, namely: Amazon Web Services (AWS), Oracle Cloud, IBM Cloud, Google Cloud and Microsoft Azure.

Our tests focused on application server and database server performance running on provisioned virtual machines, as well as evaluating the host itself. We deliberately chose a multi-tenant configuration, as opposed to a dedicated one. As we will explain shortly, though the results of these tests should not be considered definitive, it provides a fair and neutral look at the compute performance of each of these cloud providers.

However, caution should be exercised when interpreting the results as there are a number of factors that could explain some of the variances, such as:

- Ongoing and unknown back-end and hardware changes at each provider
- Varying load on back-end infrastructure due to multi-tenancy.

Overall, there was nothing alarming in the results. Generally speaking, more powerful CPUs yielded better performance irrespective of the cloud provider, though there were some anomalies. The results were mostly reproducible, with the exception of I/O.

> Generally, more powerful CPUs yielded better performance, irrespective of the cloud provider

AWS demonstrated a slight processing edge due to a newer and higher-end CPU model. Meanwhile, Microsoft Azure consistently underperformed compared to the other providers in both the application server and database tests.

EVALUATION: BEYOND PERFORMANCE

With the exception of the underwhelming performance of the Microsoft Azure instances, all other providers performed almost identically. Thus, in our opinion, other factors not related to performance are likely to influence your cloud provider decision. We experienced a number of non-performance related factors which can affect the overall experience and are worth sharing.

Unchanged is the fact that, across the board, cloud billing remains unpredictable, confusing at times, and difficult to estimate for pay-as-you-go plans. Nonetheless, cost can be interpreted more easily on the AWS and Oracle Cloud billing dashboards. Oracle Cloud continues to provide the best cost value of all providers (see Figure 1).

Unfortunately, Oracle's cloud account management remains messy. Despite multiple escalations through Oracle Sales, Oracle Technical Solutions Engineer, Oracle Account Manager, Oracle Cloud Trial Coordinator, and a Severity 1 Oracle SR, upgrading from the free to a paid account took eight days, compared to just hours on all other providers. Account management remains an area of frustration for many new users of Oracle Cloud.

Completely unacceptable is the fact that IBM Cloud has only four non-customisable firewall ports to choose from: HTTP 80, HTTPS 443, SSH 22, or "all". The only alternative is to upgrade to a $1,000 to $2,000-per-month firewall that can only be paid for via PayPal. Despite this and other limitations with IBM Cloud (for example, max network speed of 1Gbps compared to 10Gbps for AWS and 8.2Gbps for Oracle Cloud), their support outshines all other providers.

Figure 1 Compute cloud cost per month comparison

CPU/memory only, based on official pricing sheets, not actuals

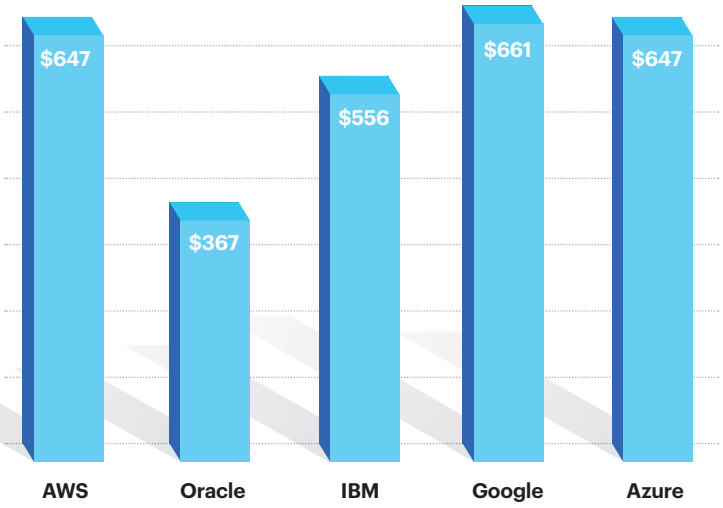


Figure 2 Final specifications of the virtual machines

	AMAZON WEB SERVICES	ORACLE CLOUD	IBM CLOUD	GOOGLE CLOUD	MICROSOFT AZURE
Region	N. Virginia	US-ASHBURN-AD-1	NA East (WDC01)	us-east4 (Northern Virginia)	East US
Profile / type / shape	m5.4 xlarge	VM. Standard 2.8	B1.16x64	(custom)	D16s_v3
vCPU	16	16	16	16	16
Memory	64 GB	**120 GB**	64 GB	64 GB	64 GB
Kernel	3.10.0-957.el7.x86_64	**4.14.35-1818.5.4.el7uek.x86_64**	3.10.0-957.1.3.el7.x86_64	3.10.0-957.1.3.el7.x86_64	3.10.0-957.1.3.el7.x86_64
Operating system	RHEL 7.6	OL 7.6	RHEL 7.6	RHEL 7.6	RHEL 7.6
CPU	8 cores / 16 threads	8 cores / 16 threads	**16 cores** / 16 threads	8 cores / 16 threads	8 cores / 16 threads
Model	**Intel Xeon Platinum 8175M CPU**	**Intel Xeon Platinum 8167M CPU**	**Intel Xeon CPU E5-2683 v3**	**Intel Xeon CPU**	**Intel Xeon CPU E5-2673 v3**
MHz	**2.50 GHz**	**2.00 GHz**	**2.00 GHz**	**2.20 GHz**	**2.40 GHz**

Google Cloud is a strong and solid contender, but is not as mature as AWS when it comes to registration, account upgrades and billing. For users of any Google service, a chief complaint is the lack of decent support. This is no different in their cloud service. During our month-long exercise, Google Cloud deleted our billing ID (a possible bug?), we lost access to all services created, then Google refused to support us because we had no billing ID.

Microsoft Azure as a cloud platform is generally solid, despite its significantly poorer performance compared to the other providers. We experienced only minor annoyances. In one case, a firewall rule took over 10 minutes to take effect. We also experienced a few random console exceptions such as, "An error has occurred attempting to gather the data."

We experienced irrecoverable instance losses from two of the providers. An "OS Reload" operation on the IBM Cloud virtual machine rendered it permanently inaccessible, and a Red Hat OS update on the Google Cloud virtual machine caused it to crash, also permanently corrupting it. The same operation worked fine with all other providers.

Only Oracle Cloud and IBM Cloud provided technical support at no extra cost.

VIRTUAL MACHINE SPECIFICATIONS

For our performance testing, we considered mid-sized virtual machines. We aimed for the general specifications of 16 vCPUs, 64 GB of RAM and 8 cores (see Figure 2).

It's difficult to perform an apples-with-apples comparison, however, as none of the CPU models were identical across any cloud provider. Each has its own instance types, shapes or profiles, and we selected models that closely resembled the target specifications, with variances highlighted in bold in the table on the left.

HOST PERFORMANCE

We used the Linux utility stress-ng to conduct all

host load testing. This is a simple workload generator that will stress test a server in areas of CPU, cache thrashing, drive stress, I/O syncs, VM stress, process creation and termination, and much more. It's ideal for establishing baselines and comparing loads across identically configured systems.

For example, the following command spawns 2,000 workers and stresses the CPU for 15 minutes:

```
stress-ng --cpu 2000 --timeout 15m
--verbose
--metrics-brief
```

Nothing alarming was found in the results, which indicated that more powerful CPUs yielded better performance (see Figure 3).

Memory stress tests yielded identical performance across all providers.

As for the I/O stress tests, as shown in the next graph, the results were relatively inconsistent during the three rounds of testing with fluctuations as high as 85% (on AWS), 35% (on Oracle Cloud), 70% (on IBM Cloud), 90% (on Google Cloud), and 89% (on Microsoft Azure). IBM Cloud reported considerably inferior performance than the rest (see Figure 4).

We also performed a 'large file copy' stress test, with a total of eight workers copying 2GB files for a period of 15 minutes (see Figure 5).

As it shows in the graph, the results are unexpected if taken at face value. Why was the performance of AWS, IBM Cloud, and Google Cloud so bad here? This can't be explained without proper back-end access or more repeated and varied tests.

APPLICATION SERVER PERFORMANCE

To test Oracle WebLogic Server 12.2.1.3 with Java 8u191 running on the provisioned virtual machines, we developed a minimalistic ADF application which included a few RESTful services and queried the standard HR schema in an Oracle Database 18c. No clustering or load balancing was considered.

We used Apache JMeter 5.0, an open source software designed to load test functional behaviour and measure performance, to run the tests (see Figure 6).

Oracle Cloud, IBM Cloud, and Google Cloud completed the total of 100,000 transactions in the same amount of time, with AWS taking slightly longer (see Figure 7).

Generally speaking, AWS, Oracle Cloud, IBM

Figure 3 CPU stress test *(higher is better)*

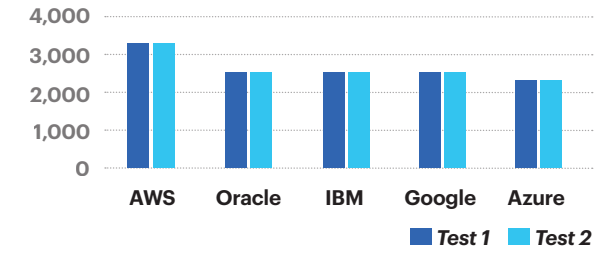


Figure 4 I/O stress test *(higher is better)*

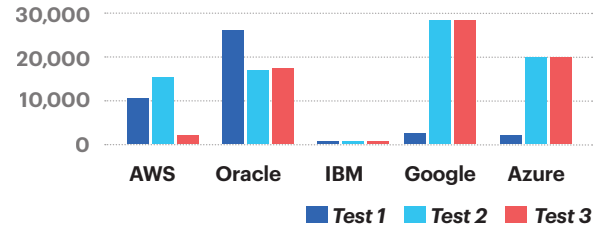


Figure 5 Large file stress test *(higher is better)*

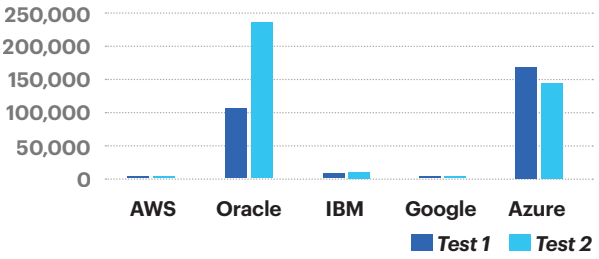


Figure 6 Apache JMeter 5.0 dashboard

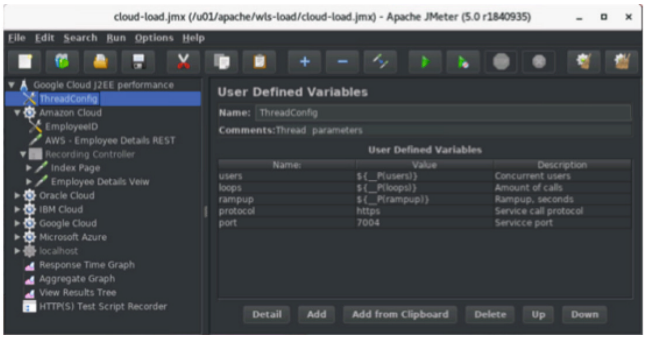
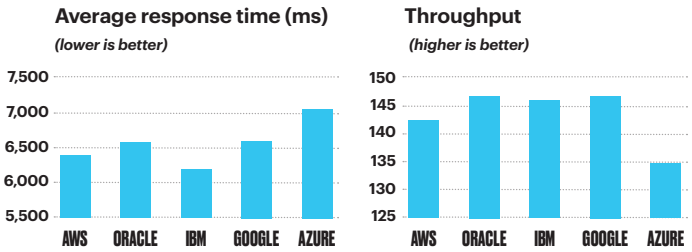


Figure 7 Results of WebLogic server load testing

	Number of transactions	Numbers of errors	Duration (minutes)
AWS	100,000	2	11.42
Oracle	100,000	2	11.23
IBM	100,000	3	11.23
Google	100,000	3	11.23
Azure	100,000	6	12.23

Figure 8



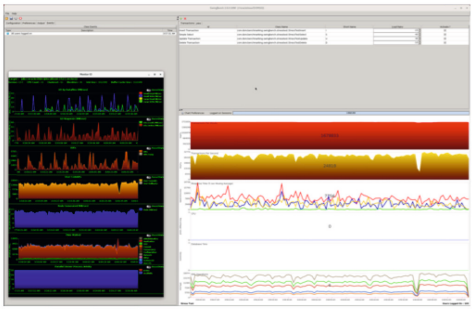
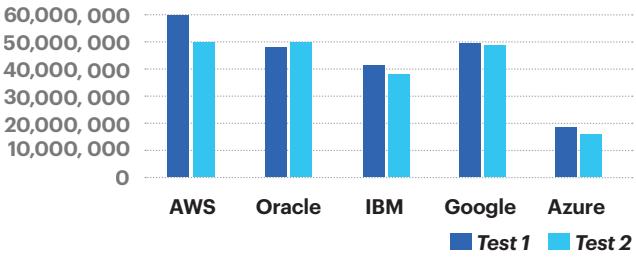


Figure 9 SwingBench dashboard

Figure 10 Total completed transactions *(higher is better)*



ABOUT THE AUTHOR

Ahmed Aboulnaga is a Principal at Attain and focuses on technical management, architecture, and consulting in Oracle, Java, middleware and cloud technologies. He is an Oracle ACE and a frequent speaker on trending technologies.

Cloud and Google Cloud had comparable throughput while Microsoft Azure completed all transactions in the longest length of time, had the highest response times, the lowest throughput, and reported 8% lower transactions per second than the rest (see Figure 8).

We found nothing major on WebLogic managed server CPU usage, requests per minute, data source statistics or heap usage (though AWS had double the heap usage compared to the rest). Worth noting is the fact that AWS's superior CPU did not directly translate to better application performance in this case.

DATABASE PERFORMANCE

To stress test Oracle Database 18c, we used SwingBench 2.6, a free load generator designed for Oracle databases (see Figure 9).

We completed two formal test runs, each running a total of 48 minutes, with a 100-user load, and a healthy mix of SELECT (40%), INSERT (15%), UPDATE (30%) and DELETE (10%) statements.

The database instance was created as a single node (no RAC), used file system datafiles (no ASM) and used the default DBCA configuration (see Figure 10).

There were zero errors or rollbacks in all tests on all providers. AWS, Oracle Cloud, IBM Cloud and Google Cloud all reported comparable throughput. AWS had a little bit of an edge while IBM Cloud a little less. Microsoft Azure had the lowest throughput consistently.

We were unable to interpret the findings for the database WAIT times, as the results were wildly inconsistent.

Oracle Cloud consistently had the highest IOPs (input/output operations per second), while Microsoft Azure graphs had lengthy gaps in IOPs and appeared to do nothing at times. We are unable to explain the severe pauses (or throttling?) in CPU and I/O that the Oracle Database on Microsoft Azure was demonstrating. This clearly had an effect on its low performance throughout.

SUMMARY

Here's a brief round-up of our results:

From a compute cloud standpoint, AWS, Oracle Cloud, IBM Cloud and Google Cloud generally performed comparably. CPU models are published on the websites of each provider and can easily be confirmed using commands such as *lshw*, *dmidecode*, and *cpuid*. Microsoft Azure compute cloud consistently performed worse than the rest.

Based on our overall experience with set-up, provisioning and support throughout the exercise:

- AWS continues to be a leader in terms of stability and maturity of its overall processes and services
- Oracle Cloud is a serious alternative from a cost and performance standpoint
- IBM Cloud is solid, but lags in service maturity and features compared to the others
- Because of its sub-par support, Google Cloud has some work to do to convince us that it's ready for prime-time
- The only real issue with Microsoft Azure was its compute cloud performance.

These tests were conducted in January 2019 and we recognise that repeating them today may or may not yield different outcomes. ☒

• *Many thanks to Michael Mikhailidi for his support.*

Exadata Cloud at Customer: Real-life experiences

'ExaCC', as it's known, seems like a perfect mix of cloud and on-premise – but what are the limitations and drawbacks? Here, we take a look at some essential information drawn from a recent implementation

By Szymon Skorupinski

10-SECOND SUMMARY

- ExaCC includes all the advantages of Exadata, a flexible pay-as-you-go model and the opportunity to enable all Oracle Enterprise Edition options and features, plus all the database enterprise management packs.
- However, as with any cloud solution, organisations deploying this platform need to change the way they think about their environment.
- Specific areas to consider include architecture and design, Data Guard, pluggable databases, supported grid infrastructure and database versions, patching, cloud UI and creating a starter database.

The Oracle Database Exadata Cloud at Customer (ExaCC) offering looks like a perfect mix of the cloud and on-premise worlds. You can have everything a cloud solution offers, but keep your data locally, which can help you to fulfil your security requirements or avoid network latency-related problems.

The list of advantages is very long. First of all, you get the state-of-the-art hardware for Oracle Database – Exadata – with all it can offer, like Smart Scans, Storage Indexes, Hybrid Columnar Compression, Smart Flash Cache, and so on. All this is available with the flexible, subscription-based pricing 'pay-as-you-go' model. If you decide not to use already existing database licences, by choosing Oracle Database Enterprise Edition Extreme Performance you enable all Oracle Enterprise Edition options and features, as well as all the database enterprise management packs. How cool is that!

Another very important advantage is the flexibility to have on-demand capacity increase with the Online Compute Bursting feature. This is certainly something many companies are looking for: to be able to handle temporarily increased workloads, which can be triggered, for example, by marketing campaigns.

On the other hand, there are some limitations and drawbacks. Knowing them before taking a final decision is crucial for avoiding disappointments (in the best scenario) or preventing problems (in the worst case). This article will help anybody considering this solution to take informed decisions about which direction to follow in order to efficiently and reliably manage their Oracle Database workloads.

(Please note that this article is based on my experiences of implementing ExaCC using Exadata Cloud at Customer Release 18.1.4.4. The project is still ongoing and I'm convinced there is still a lot to discover. Things may look completely different when newer releases come, so it's always a good idea to check the latest documentation, which can be found here: *https://docs.oracle.com/en/cloud/cloud-at-customer/exadata-cloud-at- customer/wnecc*).

THE PLATFORM PHILOSOPHY

The first thing to consider on a very general level is the philosophy of the platform. It's important to remember that even Exadata hardware is delivered with the ExaCC solution, there will be no access to most of the 'toys'. It's a cloud platform, where most of the things are supposed to work 'automagically'. Up to, and including, the level of Dom0, it's managed by the Oracle Cloud Operations Team. This means mainly that:

1. **Some features are not available or they're only available with a limited set of options.**
2. **You'll have either no access to underlying components (e.g. Cloud Control Plane, ToR and InfiniBand Network and Switches, Dom0) or very limited ones, for example "to a set of essential Exadata Storage Server monitoring and management functions, which can be performed without direct administrative access to the Exadata Storage Servers" using ExaCLI command.**
3. **In case of problems, the way to go forward is always to create an Oracle Service Request.**

This requires changing the way organisations think about their environments, and how they design and operate them. Although this isn't specific to Exadata Cloud at Customer, but rather to all cloud solutions, it's very important to keep it in mind. An example consequence could be, for instance, related to patching the underlying infrastructure which is managed by Oracle – it has to be agreed between all parties, not only internal ones.

Let's go now through some more specific points, which can have a significant impact on the success of an ExaCC implementation. They are mostly well documented, but not immediately visible when you look at the ExaCC platform for the first time.

ARCHITECTURE/DESIGN

Having well-thought-out architecture with proper naming, not only just before starting the installations, but even before ordering new hardware, has always been crucial in the IT world. In the world of ExaCC, it's even more important due to the fact that many things can't be modified, without recreating them from scratch.

This is just a time issue, when you're not yet in production – but imagine if you need to introduce that kind of change in systems which are already in use…

So, keep in mind the following rules:

CPUs

1. **CPU oversubscription – cannot be disabled (enable only).**
2. **CPUs can be increased and decreased.**

MEMORY

1. **Memory can only be increased!**
2. **Depending on the Application Type setting associated with starter database deployment (there's more on this later), the amount of system memory reserved for Huge Pages allocated on each VM is as follows:**

a. Transactional (OLTP) – 70%
b. Decision Support or Data Warehouse – 50%

STORAGE

1. **Sparse Disk Group supporting Oracle Database snapshots – you can't use this functionality if it hasn't been enabled at VM cluster creation time.**

2 Database storage – the ratio between DATA and RECO disk groups is predefined and cannot be changed. This ratio is also impacted by choosing Sparse Disk Group and Backup on Exadata Storage related options. The full table can be found here: *https://docs.oracle.com/en/cloud/cloud-at-customer/exadata-cloud-at- customer/exacc/service-instances.html#GUID-D212A16D-7CB5-4D2C-835D- 99D3E73A9EE6.*

3 Local storage – this can't be changed after the VM cluster creation.

With the above in mind, it's obvious that proper planning, especially as regards storage and memory, is inevitable. The good news is that exceptions are possible, but for each case Oracle Support has to be involved.

DATA GUARD

Exadata Cloud at Customer allows you to perform key Data Guard operations, such as switchover, failover and reinstating a failed primary database through Cloud UI, more specifically using the Oracle Database Cloud Service console. However, there's a very important limitation: both the primary and the standby database have to exist in Exadata systems configured in the same identity domain. If this is not the case, the only option would be to manage the standby database manually.

You would be forced to do that anyway, in the situation when there is more than one standby database, as ExaCC is able to manage using its tooling for only one standby database per primary. It's kind of an artificial limit, as thanks to plenty of log_archive_dest_n parameters, Oracle Databases can have up to 30 direct standby destinations.

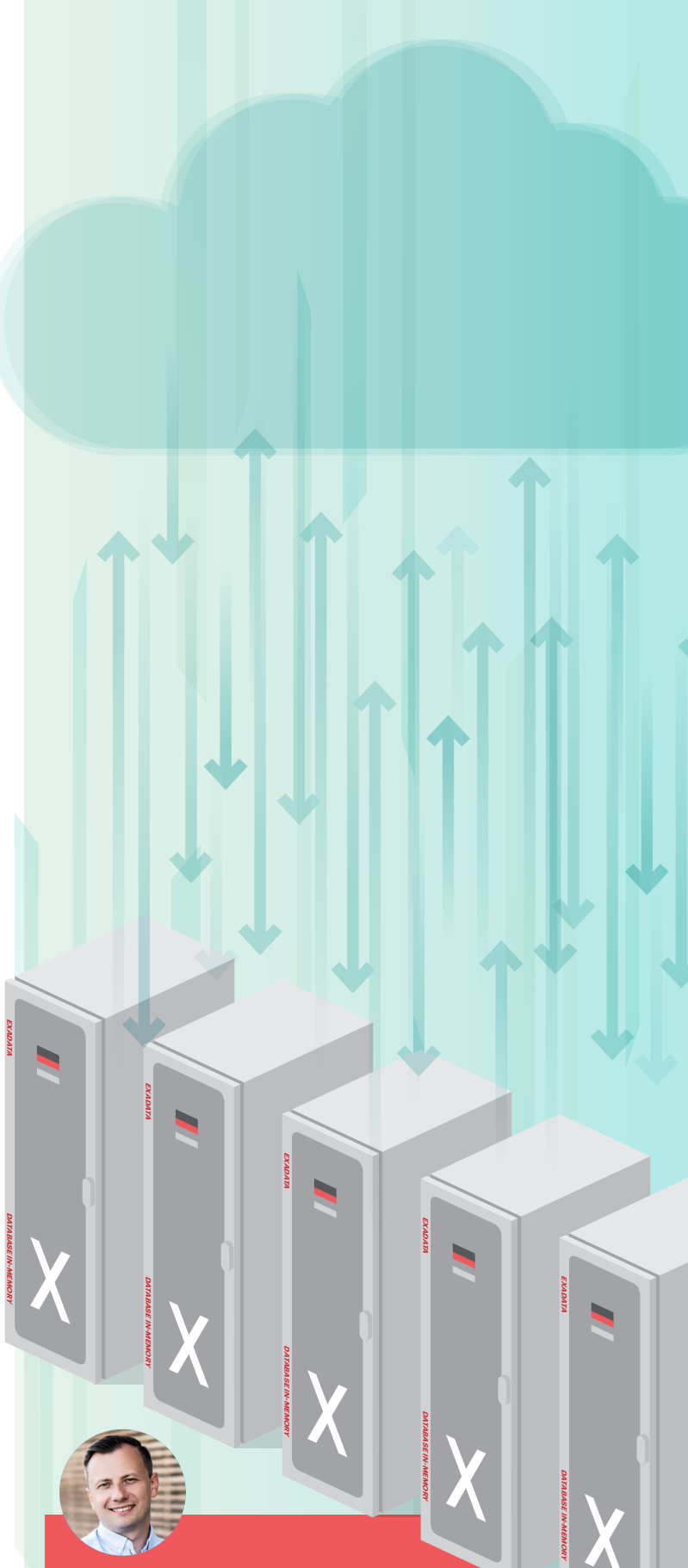
I think this limit comes from Cloud UI design and I hope it will be removed in the next ExaCC release, as it's quite a common situation to have more than one standby database for different purposes, e.g. one for disaster recovery and/or backups, the other for read-only workloads, and so on.

PLUGGABLE DATABASES

This is something that surprised me a lot – on ExaCC there is neither REST API nor Cloud UI support for administering pluggable databases. Cloud UI is 'aware' only of the first pluggable database in the container. By 'aware' I mean that it's listed on the Instance Overview page, but no single action is possible for this entity.

Fortunately, it's still possible to manage pluggable databases using a command line, thanks to the dbaascli command line tool, which allows many PDB lifecycle operations to perform.

However, the usage of the tool is strongly limited due to the fact that it can only be used for a database running at least on Oracle 12.2.0.1 and the ones which are not in Data Guard configuration. You can find more details here: *https://docs.oracle.com/en/cloud/cloud-at-customer/exadata-cloud-at- customer/exacc/administer-pdbs.html.*



ABOUT THE AUTHOR

Szymon Skorupinski is a Senior Database Engineer at Trivadis and an Oracle ACE Associate, with over 15 years' experience working in complex IT environments at every stage of their lifecycle. He is an Oracle OCP for 10g, 11g and 12c, as well as an OCI 2018 Certified Architect Associate.

SUPPORTED GRID INFRASTRUCTURE AND DATABASE VERSIONS

Oracle 19c has already been released earlier this year. It's supposed to be the terminal release for the Oracle 12.2 branch, with a focus on stability and long-term support.

Unfortunately, it's not currently possible to use this version on ExaCC. I'm not aware of any plans and dates when it will be available, but it seems the problem is related to the version of the Linux operating system used on the platform, which is OEL 6.9.

This is another thing to keep in mind, especially for long-term planning.

PATCHING

Apart from the synchronisation challenge between internal teams and Oracle Cloud Ops for patching underlying infrastructure, as already mentioned, there is another one: for patching supposed to be done by internal DBAs (e.g. Grid Infrastructure or Database patching). Using Cloud UI or REST API, it's not possible to control when the node and services running on it will be restarted. For the applications not using, for example, Transparent Application Failover, it's crucial to have strict control of the timing, in order to be in line with agreed downtime. Again, the solution is to do patching manually, for example using the exadbcpatchmulti command line tool. It's also very important to keep the tooling up to date: the process is described in *EXADATA CLOUD: Updating the Cloud Tooling for Exadata Cloud Environment dbaastools_exa (Doc ID 2495335.1)*.

Another point is that, using ExaCC provided tools, you can only apply patches staged by Oracle Cloud Operations. The only patches allowed to be staged are Quarterly Bundle patches along with Cloud patches. However, the reality is that for most Oracle database deployments, at some point, there is a need to apply one-off patches, to solve very specific issues. This can be done using opatch, a tool every DBA is already familiar with – but don't forget to create a Service Request to get the green light from the Oracle Cloud Operations Team.

CLOUD UI

To manage ExaCC deployments better using Cloud UI, it's important to understand that it's been designed in a database-centric way. That means, for example, that the *Instance Overview* page is the one to be used when you want to restart a cluster node from a web interface, as those are listed as *Resources* of the database. Of course doing that will affect availability of the other databases running on this node… Not a very obvious design choice, but this is how it is.

STARTER DATABASE

Another term to learn in order to understand the platform better is 'starter database'. It's the first database you create on a VM cluster. First of all, it determines the Oracle Grid Infrastructure software release version that's configured in your VM cluster. The process is not very well optimised, but I understand it's designed like that to avoid unnecessary problems. So, when you want to create a new VM cluster that's meant to host 18c databases inside your Service Instance, you have to:

1. **Create a new cluster – it's provisioned with Grid Infrastructure 12.2 installed.**
2. **Create a starter database using Oracle 18c. This action triggers Grid Infrastructure removal and installation of GI 18c, before the database is created.**

If you want to host, let's say, 12c databases on GI 18c, the starter database can be removed afterwards. However, it's best to do it only after at least one additional database is created, as during the creation of the first database or removal of the last database in the cluster, some additional actions are performed, making the process last longer.

Please also keep in mind the additional difference between starter and later database deployments. The former is configured with USE_LARGE_PAGES=ONLY, which forces it to use Huge Pages, while the latter is configured with USE_LARGE_PAGES=TRUE. This comes from the assumption that the first database will always have enough Huge Pages to allocate SGA using them, which can't be guaranteed with the next ones.

SUMMARY

In my opinion the biggest issue with the ExaCC platform comes from the limitations introduced due to the interfaces (mostly Cloud UI). Fortunately, a lot of functionalities are still available if the web interface is not used. If you decide not to use it, you should at least try to keep it synchronised. Sometimes it's not possible – for instance when you create a database manually, Cloud UI will not be aware of that.

However, as regards the patching: this seems to be possible. After installing Bundle Patch manually, you need to click on the Patch button in Cloud UI. It will start the patching process and (fortunately) without restarting any machine it will realise that the patch is already there and will update Cloud UI to remove it from the list of available patches. At least it worked that way when we tested it in our environment… Nevertheless, I'm still concerned about future compatibility/support problems, caused by manual actions not reflected in the Oracle Cloud Control Plane.

As an old-school DBA, I suggest not relying too much on WebUI/GUI interfaces. Even though they can sometimes be helpful, it's always a good idea to verify things manually. As an example: patching initiated and reported successfully in Cloud UI, left one node of the cluster not fully patched. So even if you initiate the patching using web interface, log in to the machines and check if everything is fine using command line tools.

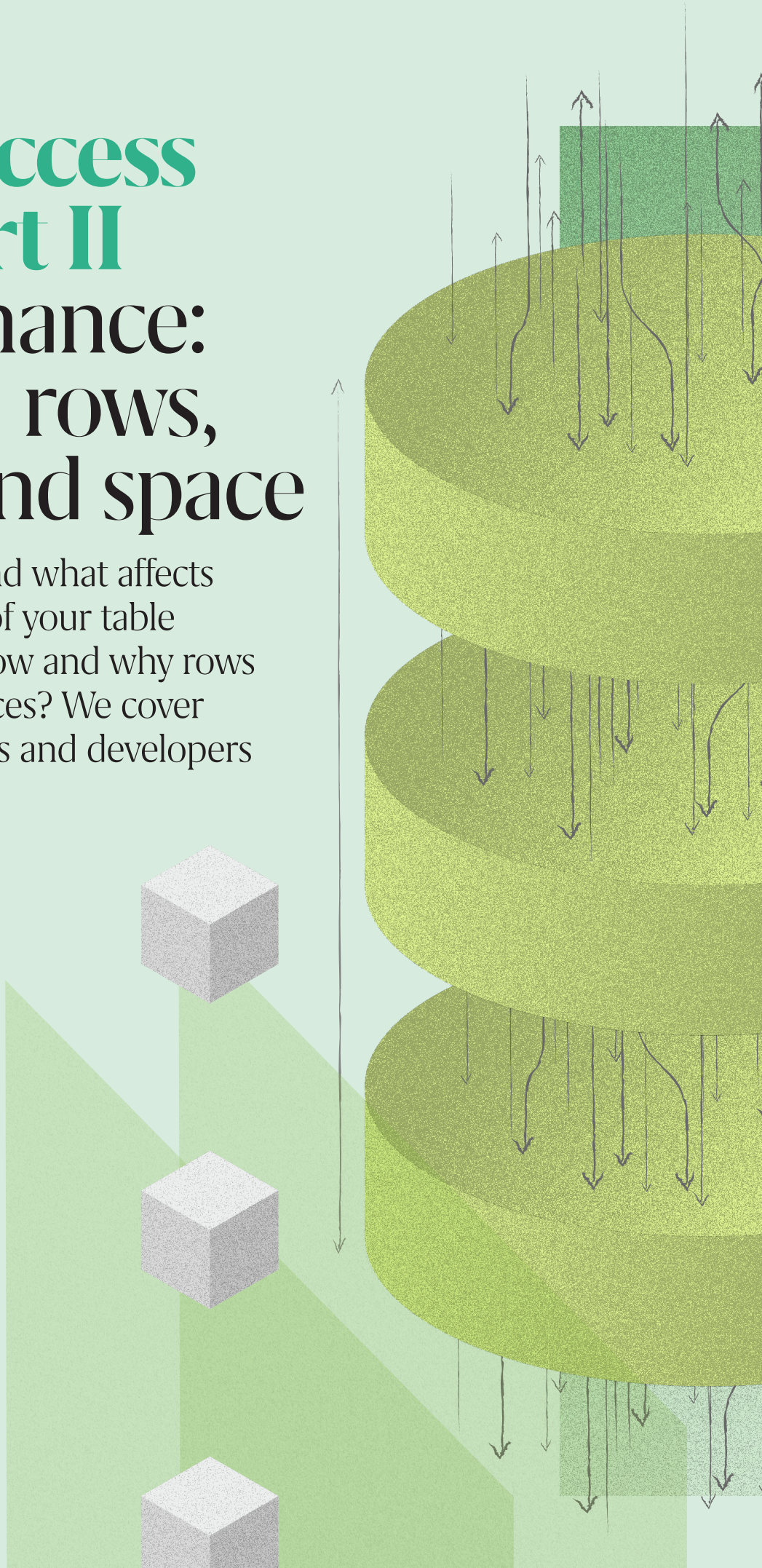
I hope Oracle will continue to work on stabilising the platform and making it more flexible, by at least removing the restrictions that are not of a purely technical nature.

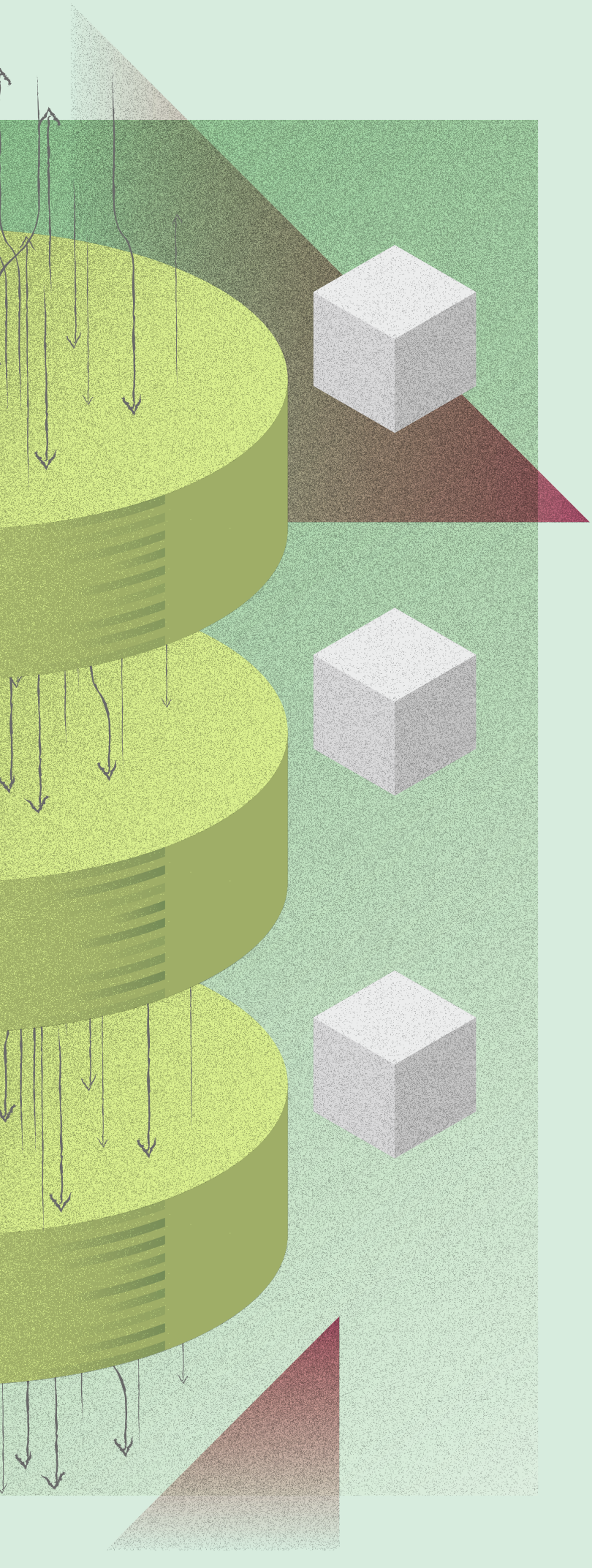
Table access full, part II

Performance: chained rows, DMLs and space

Want to understand what affects the performance of your table scans, including how and why rows get chained in pieces? We cover the basics for DBAs and developers

By Roger MacNicol





10-SECOND SUMMARY

- **This article follows on from last issue's basic look at Table Scans and how to tackle associated performance problems.**
- **Here, we take a closer look at how rows are actually stored and examine performance implications for scans and DMLs.**
- **We also examine three ways of releasing space back into your system to restore performance if you find things are slowing down.**

In the previous issue of *#PTK*, we looked at how Oracle table scans use memory. In this issue we will look a little deeper at how rows are actually stored, as well as the performance implications for scans and DMLs (data manipulation languages).

A single row of data has at least three distinct parts and is stored in units known as 'row-pieces'. Each row-piece can contain up to 255 columns.

A row must contain:

- A Head piece 'H' which anchors the rowid* for use in indexes
- A First piece 'F' which contains the start of the first column plus following columns
- A Last piece 'L' which contains the end of the final column and marks the end of the row.

All three of these may be in a single row-piece if it's a newly inserted row with fewer than 256 columns. The 'H', 'F', and 'L' refer to flags in the header of a row-piece, like this:

A row with more than 255 columns is always broken down into multiple row-pieces and when this happens it's known as row chaining. The chain is created by storing a rowid at the beginning of any row piece which has another row piece that comes after it with more columns in it. The rowid used for chaining the pieces

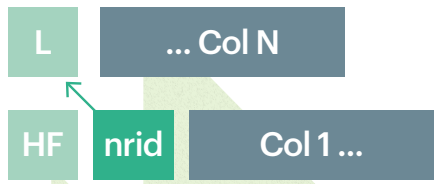
**Rowid is a 'row identifier' that precisely locates a single row's position in a particular block in a particular data file.*

together is known as a 'next rowid', which is abbreviated to 'NRID'.

When you ask Oracle to store a row with more than 255 columns on a new block, what happens depends on how it's inserted. For conventional inserts, Oracle starts at the last row-piece and writes that onto the block so it knows the rowid to use as the NRID for when it writes the next row-piece onto the block. As each row-piece is written, Oracle finds the next available space from the free list – so you can make no assumption about layout if there have been a lot of deletes in the table and space has become fragmented. If you have a mostly insert-based workload, the row-pieces will be written contiguously in reverse order from last to first piece.

For Direct-Path inserts, Oracle writes rows starting at the Head/First piece: since it controls the layout you can go back and fill in the NRIDs after laying out the row-pieces on the block. Direct Path inserts are guaranteed to write all the row-pieces contiguously in order, from the first piece to the last piece.

After a block has first been written, all the row-pieces will be contiguous on the block so chained rows should have minimal performance impact. For Direct Path loaded data, once a table scan finds the head piece it can walk forward through the block to get columns above 255; if a row piece is on the following block that block will probably already be in the buffer cache from the read-ahead or will already be in the adjacent Direct Read slot (see Part 1 in #PTK Issue 70) from multi-block I/O. For conventionally-inserted data, the scan has to follow the NRID chain which may or may not be contiguous on the block and so may have a greater impact on performance.



PERFORMANCE TIP

Using Direct Path loads can lead to faster query performance on tables with more than 255 columns! For example, if you have a table with 260 columns and want to run a report that only accesses columns near the front of the row... With conventional inserts Oracle will have written an 'L' row-piece of 255 columns and then an 'HF' row-piece with 5 columns, so accessing column 6 will cause a chained row fetch for every row. With Direct Path loads, Oracle will have written an 'HF' row-piece with 255 columns and then an 'L' row-piece with five columns, so using column 6 would not cause an additional chained row fetch.

A ROW'S LIFECYCLE AND PCTFREE

When you create a new table in Oracle it populates PCTFREE with its default value of 10%. This can (and sometimes should) be overridden explicitly in the Create Table grammar. We'll discuss why shortly, but first let's understand what they do:

```
CREATE TABLE mytab (c1 number) PCTFREE 10;
```

PCTFREE specifies how much empty space to leave for future updates when a block is first populated.

Now let's consider a row's lifecycle. Imagine an order processing table where various elements are not known when the row is first inserted and the transaction committed: for example, you may not know the processed date, the ship date, the tracking number, or delivery confirmation. Each of these may be filled by subsequent transactions as they become known. A row which was 150 bytes when first written might finally take up 180 bytes – i.e. be 20% larger. If you had specified a PCTFREE of 20%, each of those updates would have found room in the original block.

Had you left the default PCTFREE 10%, half of the updates would not have fitted in the original block and Oracle would have had to split the row into two row pieces in order to move the additional data to another block. In order to find a home for that new row piece the free list would have been checked and the new row piece may end up on a block stored some distance away from the original one.

But why does this matter? Let's say you now need to retrieve that row and need a column that's stored in the new row piece – the 'NRID' will be passed to a buffer cache get, and getting that block may result in another single block disk I/O.

You may not notice the performance impact when retrieving a single row but a full table scan which does an additional disk I/O for each row will have a significant impact on performance.

Now imagine you hadn't left enough space in the block. Then you update column 100 which causes the row piece to split and half the row has to move to a distant block to find space. And then you update column 50 and there still isn't space so the remaining row-piece gets split again and the new piece moves to a different distant block. I visited one customer whose update pattern had left every row fragmented in up to nine row-pieces, each with four to five columns in each row-piece. They needed eight additional block gets for every row and many of those turned into disk I/Os – they were not happy with the performance of single row retrieval or table scans!

PERFORMANCE TIPS

- **Plan out the lifecycle of your data from its initial insert to its final state and check the percentage increase in size (if any).**
- **If you have an insert-only workload where the data is never updated, it's safe to set PCTFREE to zero and your data will take less space and queries scan fewer blocks.**

If you realise you should have created a table with a larger PCTFREE, it's not too late. You can increase it with an Alter Table statement:

```
alter table mytab pctfree 30;
```

And you can see what the current value of these is for any table simply by looking at the user_tables view:

```
select table_name, pct_free, pct_used
from user_tables

where table_name is 'MYTAB';
```

MIGRATED ROWS

When you update a column near the front of the row, Oracle may decide to split the row between the Head piece and the First piece so as not to create a very short row piece with only a few columns in it. After the free-list has been searched for space the First piece will be stored in a different block. This is what's known as a 'Migrated' row.

Why does Oracle do this? Remember the Head piece is what anchors the rowid which is used by indexes to quickly locate the row. If Oracle moved the Head piece, it would have to update any indexes on the table with the new rowid. By leaving the Head piece where it is and only moving the first piece, the update proceeds faster and generates less redo.

HEAD PIECE SCANS vs FIRST PIECE SCANS

Do migrated rows affect performance? In general, a table scan (including parallel scans) is looking for First pieces as the beginning of the row and safely ignores Head pieces from migrated rows. But, if the scan requires the rowid to be returned because it's a Select For Update, a Searched DML, an Index build, or the user referred to the rowid in the query, then it starts looking for rows with their Head pieces because that defines the rowid. In these cases, each migrated row will cause another buffer cache get and possibly another disk I/O to retrieve the rest of the row.

DELETES

When rows are deleted and the free space on the block exceeds the PCTUSED value for the table, the block is added to the free-list and becomes available for new rows or row pieces to be written on it. If you continue to delete all the rows from the block it remains on the free-list – it doesn't get released back to the system for other segments to use. Each time you scan that table the empty block will still be read and looked at to see if it has any rows on it.

If you delete a large number of rows from a table with a searched delete, the segment still has all the same blocks and they all have to be scanned each time. (Note, for this reason you use Truncate rather than a Delete if you want to delete all the rows from a table, as Truncate does release space back to the system).

RESTORING PERFORMANCE

So how do you release space back to the system if a segment will continue to have fewer rows? And how do you defragment rows if updates have left them fragmented and are slowing the system down? There are three ways to do this, each of which has pros and cons.

1 Alter Table Shrink Space Compact

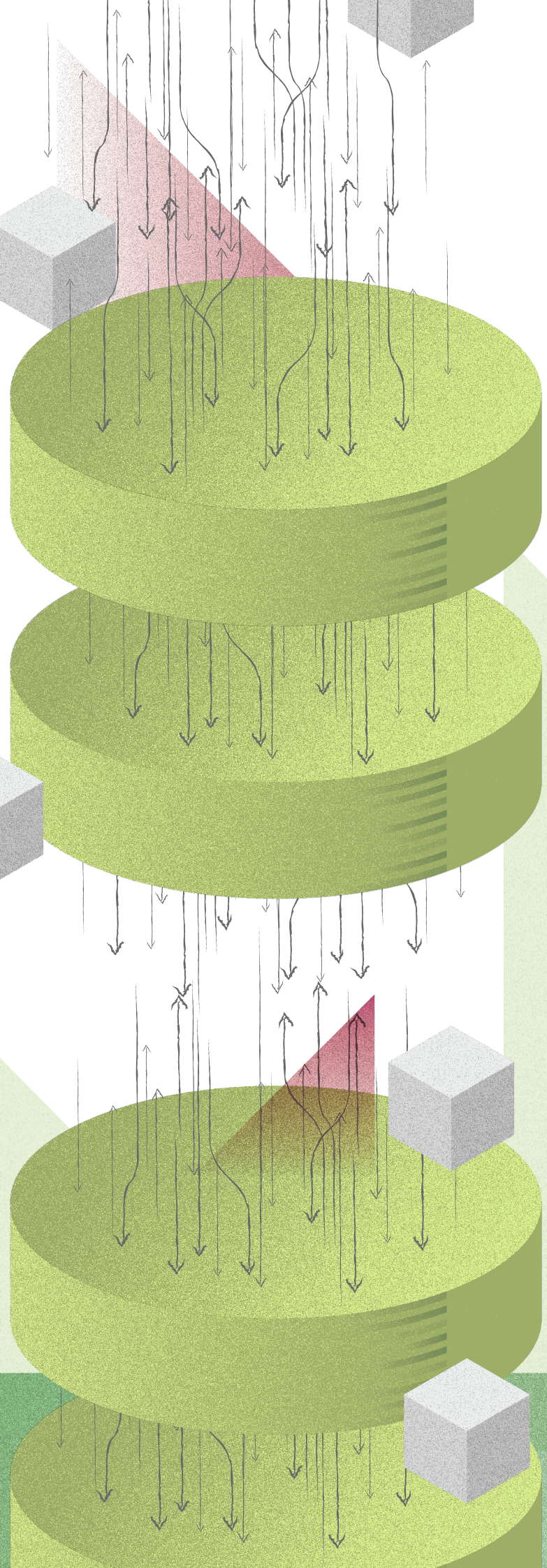
The sole purpose of Alter Table Shrink is to repack the row pieces into fewer blocks, and if there are only a few blocks to be released back this will be the fastest way. You use Alter Table Shrink Space Compact as an online operation that repacks the rows followed by Alter Table Shrink Space as a quick offline operation to release the space back to the system. This also has the advantage of not using any additional disk space. But, prior to 12.2 it did not try to keep row-pieces for a row together and could cause performance problems with Exadata Smart Scan.

2 Alter Table Move Online

Since 12.2, Oracle has supported moving a segment online which has to repack every row. However, it lays down a completely clean copy of each row just as if they had been inserted into a new table and so does a much better job at defragmenting rows than Alter Table Shrink can do. The disadvantage is that it takes twice the disk space while it runs.

3 Data Pump export and re-import

Data Pump can also create a segment with all the rows laid out as if they had just been inserted into the table and like Move it fully defragments each row. The advantage is that is does not take twice the disk space but it is a fully offline operation and requires halting use of that segment while it's repacked.



PERFORMANCE CHANGES IN 19C EXADATA

With 19c as part of simplifying space management, changes were made to try to prevent the creation of migrated rows in tables that have row movement enabled.

Enabling row movement tells Oracle that it's OK to change a row's rowid: this is required, for example, if you update the partitioning key of a row so Oracle has to move the row to a different partition; it's also required when using Alter Table Shrink to recompact a segment.

To enable row movement in an existing table you would use:

```
[alter table mytab enable row movement;]
```

In 19c, when a segment is stored on an Exadata tablespace and the table is not Index-Organized, not Clustered, and does not contain LONG columns, Oracle will try to defragment a row when processing an update and will try to avoid creating any new migrated rows.

It will only do this if it can find free space somewhere to store all of the row-pieces for the row contiguously. Since a row is likely to be scanned many more times than it is updated this ensures better performance over the long run.

ABOUT THE AUTHOR

Roger MacNicol is Architect, Data Storage Technology at Oracle

Designing an asynchronous application interface

How a rewrite reduced risk and increased reliability for one happy customer

By Robert S Jackson

10-SECOND SUMMARY

- Robert's team were challenged to take large volumes of historical data from a data warehouse and create a new transactional system that needed to keep in sync with daily updates.
- An asynchronous application interface was needed because of the business demands.
- Valuable lessons learned included: the importance of the design and thorough exploration of all possible error scenarios; deploying industry standard tools where possible, rather than coding from scratch; rewriting the interface to reduce risk and increase reliability.

Negotiating data feeds from your data warehouse can be arduous. The one we were dealing with was relatively large, housing terabytes of data and constantly growing. The warehouse had numerous batch load and cleansing jobs all running at scheduled times with many dependencies. Our application took the historical data and created a new transactional system which needed to keep in sync with daily updates (see Figure 1, below).

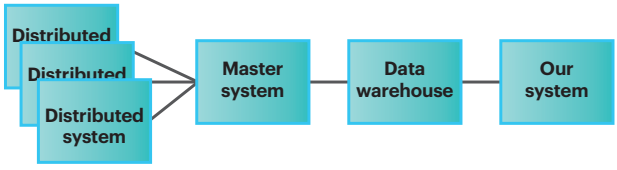


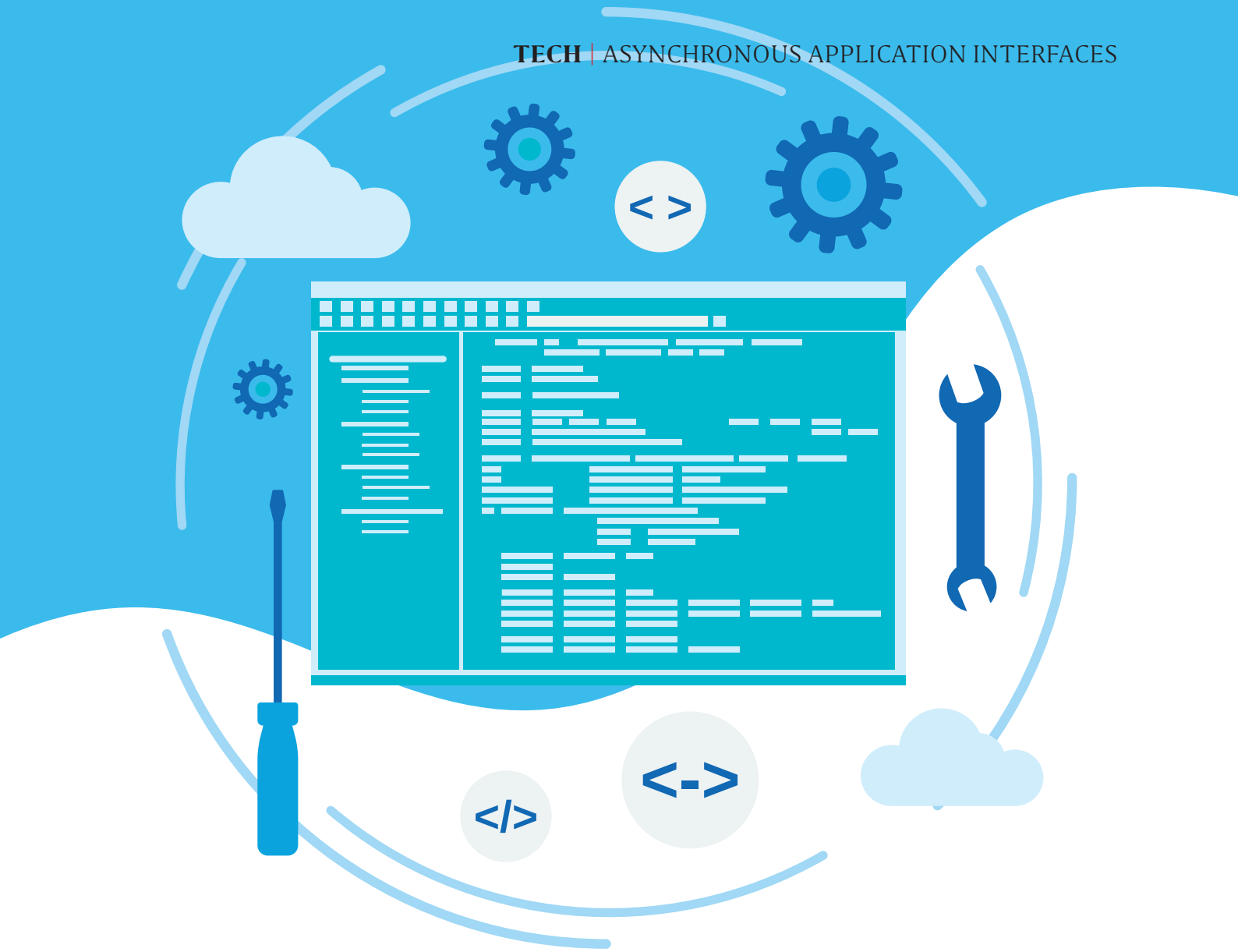
Figure 1

The rationale for application development came about because the monolithic master system could not be modified, and there was data which needed to be added and tracked at a more granular level.

When developing an application interface, there are two paths you can take – synchronous interface or asynchronous interface:

- **When using a synchronous interface** you can receive instant acknowledgement of success or failure. This type of interface depends heavily on the network infrastructure and requires both parties to be in constant communication. For example, records are inserted, updated, or deleted followed by a commit to complete discrete transactions. If an error occurs, you can add exception handling and retry the operation.
- **When using an asynchronous interface** the transactions occur in one direction and are processed in order. Batch processing is an asynchronous method. With asynchronous interfaces you do NOT receive an acknowledgement. The process continues until it completes. However, if there is an interruption or an error persists, there is the ability to restart it cleanly. For example, if a transaction is in error due to network connectivity or database issues where the database is down, the interface can handle it by failing and restarting at a later time.

The initial design and development of the interface was done quickly and on an infrastructure which was in flux, in that the network infrastructure was changing and being developed at the same time. For this reason decisions were made which were forced by security



policies and were not necessarily the most efficient method. The initial design included a whole host of products and operating systems including Solaris, Secure File Transfer Protocol (SFTP), UNIX cron jobs, dynamic KORN shell scripts, SQLPlus, Oracle and Teradata databases.

The decision to create the asynchronous interface was made because of organisational policy and was carried on through our rewrite. Because of the disconnected nature of the interface, timing was important. There were jobs which needed to complete on the data warehouse side before our interface files could be produced. When the files were produced, they had to be ingested into our system in date-time order.

Due to the fact that these systems were on different computers, sometimes the network connectivity was not available so files would be buffered up. Our initial interface could not handle multiple files and required human intervention to ensure files were processed in the correct order. This was a painful manual process and needed to be automated.

Once the network infrastructure became more stable, it gave us the ability to re-examine the interface and come up with a more maintainable and sturdy application interface.

DATA PUSH FROM TERADATA

Generating the six data files required a lot of processing time on the Teradata side. The job, depending on the system usage at the time, took three to four hours to process – and we dubbed the Teradata job which generated the files the 'Node Crusher'. With processing times like this and the finicky nature of the network we decided that the batch approach we had chosen was still a valid one. This also gave us the ability to restart the file when problems arose.

In the initial interface, Teradata would push six data files to our SFTP server. We would then unzip and use Oracle SQLLoader to populate our staging tables. This was problematic for a number of reasons:

- The file names of the zip and datafiles were a combination of filename and timestamp of when the file was created, for example INVENTORY201907191234.zip.
- There were KORN shell scripts which unzipped the six data files and created dynamic SQLLoad .par files so the

data files could be loaded into the staging tables.

SQLPlus was needed to see what the timestamp of the last data file processed was and write it to a file so the scripts could decide if that file had been processed or not. This was to ensure that in case of load failure due to network connectivity, files do not get processed multiple times. When a file was processed but the load routine crashed and did not clean up and remove the processed zip file we had to make sure that the file did not get processed again (see Figure 2).

In the new interface we used a more standard Extract Transform and Load (ETL) tool called Informatica. Informatica was able to perform all of the load operations we needed to get the data from the Teradata database into our Oracle staging tables using six workflows. This removed the complicated and temperamental KORN shell pieces, which resulted in a much cleaner and more efficient data push process.

We moved the data file creation date which we named the DERIVED_DATE into the staging tables, which allowed us to more easily track which files were awaiting processing. By adding the DERIVED_DATE to the staging tables we could hold multiple data loads grouped by DERIVED_DATE (see Figure 3).

In the early design phases we had discussed using Informatica to perform the entire load into our database structures but we eventually abandoned this due to the complicated business rules needed to reformat the data warehouse records into our transactional data structures. This allowed us to save time by reusing much of our Oracle load routine code which was stable and worked well with the existing staging tables.

PROCESSING FROM THE STAGING TABLES

On the Oracle side, we used a Packaged Procedure to read the data from the staging tables and insert, update, or delete records in our database structures. We created what we called a 'HANDSHAKE' table. This table would hold information about the Informatica load and data about processing on the Oracle side.

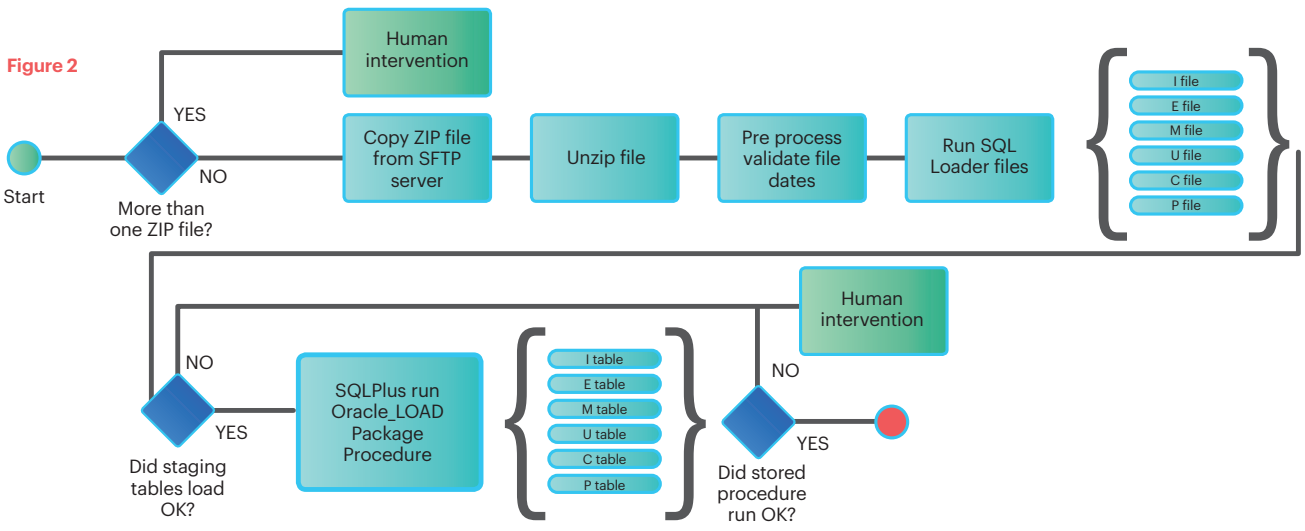
```
CREATE TABLE LOAD_INFORMATICA_LOG
(
 INFORMATICA_LOG_PK          NUMBER NOT NULL,
 INF_WORKFLOW_NAME           VARCHAR2(255 BYTE),
 INF_START_TS                DATE,
 INF_END_TS                  DATE,
 DERIVED_DT                  DATE,
 INF_RECORDS_LOADED          NUMBER(10),
 PROCESSED_DT                DATE
)
```

Before we could start our import side of the interface, we had to make sure that all six sets of data had been inserted. We used the following select to determine that all the data sets from the Teradata side had completed successfully.

```
V_ALL_FILES_EXIST := 'F';

SELECT *
 INTO V_ALL_FILES_EXIST
 FROM (SELECT 'T'
     FROM DUAL
     WHERE   EXISTS
           (SELECT COUNT (*)
            FROM informatica_log
            WHERE   INF_WORKFLOW_NAME = 'I'
               AND DECKALS_PROCESSED_DT IS NULL
               AND dp_derived_dt = DERIVED_DATE_REC.DP_DERIVED_DT
           HAVING COUNT (*) = 1)
       AND EXISTS
           (SELECT COUNT (*)
            FROM informatica_log
            WHERE   INF_WORKFLOW_NAME = 'E'
               AND DECKALS_PROCESSED_DT IS NULL
               AND dp_derived_dt = DERIVED_DATE_REC.DP_DERIVED_DT
           HAVING COUNT (*) = 1)
       AND EXISTS
           (SELECT COUNT (*)
            FROM informatica_log
            WHERE   INF_WORKFLOW_NAME = 'M'
               AND DECKALS_PROCESSED_DT IS NULL
               AND dp_derived_dt = DERIVED_DATE_REC.DP_DERIVED_DT
           HAVING COUNT (*) = 1)
       AND EXISTS
           (SELECT COUNT (*)
            FROM informatica_log
            WHERE   INF_WORKFLOW_NAME = 'U'
               AND DECKALS_PROCESSED_DT IS NULL
               AND dp_derived_dt = DERIVED_DATE_REC.DP_DERIVED_DT
           HAVING COUNT (*) = 1)
       AND EXISTS
           (SELECT COUNT (*)
            FROM informatica_log
            WHERE   INF_WORKFLOW_NAME = 'C'
               AND DECKALS_PROCESSED_DT IS NULL
               AND dp_derived_dt = DERIVED_DATE_REC.DP_DERIVED_DT
           HAVING COUNT (*) = 1)
       AND EXISTS
           (SELECT COUNT (*)
            FROM informatica_log
            WHERE   INF_WORKFLOW_NAME = 'P'
               AND DECKALS_PROCESSED_DT IS NULL
               AND dp_derived_dt = DERIVED_DATE_REC.DP_DERIVED_DT
           HAVING COUNT (*) = 1)
    UNION
    SELECT 'F' FROM DUAL
    ORDER BY 1 DESC)
 WHERE ROWNUM = 1;
```

Figure 2



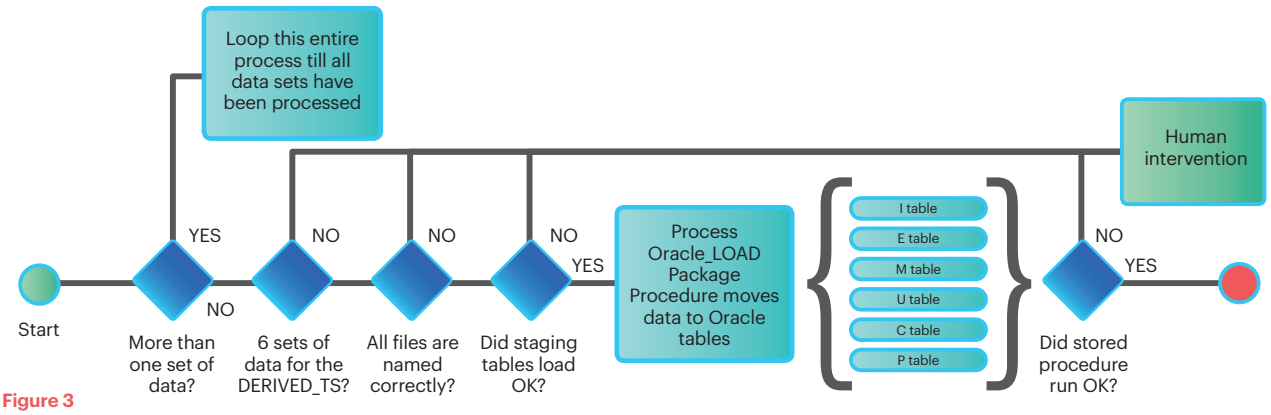


Figure 3

BUSINESS RULES

The data is fed into our master system from disconnected distributed systems (see figure 1). Since data can come into the data warehouse out of order, we depend on the data warehouse to provide us with a chronological history. It's crucial that when we load our transactional system that we keep that historical integrity. It's also very important that we process the data files we get from the data warehouse in the correct order, so we do a lot of validation to ensure that the data does get processed in the DERIVED_DATE order.

HOW DO YOU KNOW WHEN THE DATA IS READY?

Before the Informatica load runs, the process would delete from the staging tables where the DERIVED_DT was the same as the data about to be loaded. This ensured that if the informatica process had previously failed due to running out of spool space, Oracle tablespace, or any other reason, that the data would not be repeated and that it would be a clean load. Each Informatica workflow would populate the LOAD_INFORMATICA_LOG table upon successful processing of the dataset by entering the workflow name which just ran, the date and time it started, date and time it ended, the derived date that it was for, and the number of records loaded.

HOW DO YOU UPDATE THE HANDSHAKE TABLE TO SHOW YOU'VE FINISHED PROCESSING?

After successful loading of the staging tables, there should be six distinct workflow names all with the same derived date and valid start and end dates. At this point the Oracle load routine could start and process the data. Upon completion it would write a PROCESSED_DT to the LOAD_INFORMATICA_LOG and the load for that derived date would be completed.

REPROCESSING

In the old interface, reprocessing was generally a matter of manually processing a file when the interface on the Oracle side was down and Teradata had pushed more than one zip file, resulting in multiple zip files being staged on the SFTP server. This caused the Oracle side of the interface to halt and await human intervention.

In the rewritten interface the only time reprocessing has occurred is when the Teradata or Informatica processes fail. Since they automatically clean up before processing begins, the processes just need to be restarted, and the Oracle load processes take over when the Informatica piece completes and writes to the 'HANDSHAKE' table.

CONCLUSION

There are many lessons to be learned from this interface. Top of the list is the design and thorough exploration of all possible error scenarios. We had caught most of them, but one scenario we did not account for is when the Teradata process runs and no records are generated for one of the workflows. The Informatica workflow did not enter any record into the 'HANDSHAKE' table, resulting in less than six files being generated.

This, coupled with an oversight, and causing two files with the same workflow name to be generated, thus creating six files, caused lots of problems. Luckily it was discovered within the first few days of the stand-up of the interface and we were able to reload from backups and restart the new interface from day one again.

Always try to use industry standard tools, such as ETL, to perform functions they were made for instead of coding from scratch. It's frequently in a developer's nature to want to code something from scratch, but using Commercial Off-The-Shelf (COTS) tools makes your applications more maintainable and flexible for future upgrades.

The rewrite of the interface reduced our risk and increased the reliability of our application interface. Sometimes a solution is forced because of circumstance, but when given the time to correctly develop and architect a solution it makes for a much more stable product and happier customers. ☒

ABOUT THE AUTHOR

Robert Jackson is a Senior Software Engineer for KBR. He has been a defence contractor for 30 years and has been using Oracle tools for more than two decades.

Turn over now!

As usual, we've divided *#PTK* into two halves – one for UKOUG's Business Apps community and one for our Tech community.

They're both related, as are our two communities, and we're sure everyone will find each half useful.

Now flip over the magazine to see what we mean.

We'd love to hear what you think of our new look, what you like and dislike about this issue, and any ideas you have for future editions – after all, this is *your* magazine and a key benefit of UKOUG membership. So please send your comments and suggestions to: **editor@ukoug.org**

#PTKOnline

You can view this latest issue online and access the archive of *#PTK* and *Oracle Scene* editions here: **ukoug.org/ptk**