The Cloud-First Strategy of Oracle Database 12c Release 2

Publication Date: 31 Jan 2017  |   Product code: IT0014-003194

Tony Baer
Summary

Catalyst
The cloud is getting harder to ignore; according to Ovum's latest ICT Enterprise Insights data, 40–45% of all enterprises expect to raise their cloud spend in 2017. The cloud has become harder for Oracle to ignore; Oracle forecasts roughly 80% of its client base will embrace cloud in the next decade. Against that backdrop, it is not surprising that Oracle Database 12c Release 2 (or as we will abbreviate it, Database 12.2c) is Exhibit A for Oracle's emerging cloud-first strategy. When this report was published, Database 12.2c was generally available on all Oracle Cloud Database Services including the new Exadata Express Cloud Service – that is the entry level for Oracle Database as a Service. We expect that availability of Database 12.2c on-premise will come in mid-2017.

Ovum view
Oracle has delivered a major database update that keeps current with cloud deployment, multimodel, and big data integration trends. The major differentiator is Oracle's market-leading approach to database multitenancy (PDBs, or pluggable database containers): for enterprise customers, Oracle’s approach provides a good building block for keeping pricing competitive. Also of note is Oracle’s embrace of sharding, adopting a technology that provides a lighter-weight, more modern alternative for distributed internet-scale transaction processing to Oracle RACs (Real Application Clusters). Greater integration with Hadoop provides new paths for blending big data analytics.

Key messages
- Oracle Database 12c Release 2 is a poster child for Oracle's new "cloud-first" product strategy.
- Many, but not all, of the enhancements to Database 12.2c are cloud-driven. But features like sharding are likely suited for cloud because that is where the extreme workloads that demand sharding are more often than not likely to be run.
- Database 12.2c has a number of enhancements that improve its multimodel data support and integration with big data analytics.

Recommendations

Recommendations for enterprises
Given Oracle's cloud-first strategy for Database 12.2c, it should not be surprising that cloud readiness is the highlight of the release. Admittedly, the headline – expansion of the number of PDBs (Oracle’s approach to cloud multitenancy) is multiplied nearly twentyfold – will not affect enterprise customers directly. But it will have a very important, indirect impact on customers: Oracle can deliver Database 12.2c far more efficiently and economically, allowing it to become very price-competitive with Amazon's own managed Oracle database services. And for customers with multiple instances that they want to move to the cloud, PDBs will reduce their DBA and administrative overhead. Features such as sharding will provide customers with more modern, lighter-weight alternatives for internet-
scale transaction processing compared to Oracle RAC; initially this will be useful for new workloads rather than moving over existing ones, which would have to be modified to move to sharding. (This will be fine later once your team grows accustomed to sharding and works out the kinks.)

The other major theme focuses on big data analytics. There are new capabilities such as "approximate results," partitioned external tables, graph, and JSON processing capabilities that are associated with handling analytics at scale and with nontraditional data sources. Oracle is not alone or way ahead of the pack in blending its database with Hadoop and taking on more NoSQL-like capabilities; most data platforms are also becoming multimodel. But that is not the point; what is important is that Database 12.2c is better situated to take on cloud-oriented workloads and become a first-class citizen in the multipolar, hybrid big data processing environments that are becoming the norm.

Recommendations for vendors

Do not take Oracle database in the cloud for granted: Database 12.2c makes Oracle more competitive in this area. PDBs are an innovative approach for managing multitenancy that gives Oracle the opportunity to become cost-competitive in the cloud – whether it reaches the economies of scale to be the cost leader will depend on business execution (how well it scales its customer base) and related innovations for cloud networking and storage infrastructure. While Oracle, as a new player in cloud, has nowhere near the cloud market presence of Amazon or Microsoft, its FY17 Q1 SaaS and PaaS revenue surge of 77% year over year provides good starting momentum. Oracle also came out swinging with $175/month pricing for Database 12.2c on the new entry-level Exadata Express Cloud Service. The addition of sharding provides Oracle another foothold for competing for distributed, cloud transaction processing workloads – but it will not put NoSQL players like MongoDB or DataStax out of business. Enhancements to multimodel data place Oracle in the middle of the pack – it is competitive with IBM and Teradata, and ahead of Microsoft.

Scaling out for the cloud

Cloud-first

Oracle has made the second release of Database 12c a poster child for cloud-first development with general availability on all Oracle Cloud Database Services, including the new Exadata Express Cloud Service. Oracle at this point is not committing to when it will release Database 12.2c for on-premise, but we expect phased rollouts beginning in mid-2017.

As noted above, at first blush, this release is all about features that scale and make Oracle database more efficient in the cloud, meaning Oracle can price cloud delivery of its database more attractively. But there are other features in this release that will appeal to on-premise and cloud customers alike. The highlights, which we will spotlight, concern extensibility for different data types (e.g., JSON, text, and graph) and big data support that will appeal to existing on-premise customers. And while we will not go into exhaustive detail about each feature, the new release addresses hundreds of core, operational enhancements to the database in areas ranging from availability to improved data compression, indexing, partitioning, diagnostics, manageability, performance, security, and others.
Expanding the pluggable container envelope

To recap, pluggable databases are Oracle’s twist on multitenancy; rather than manage multitenancy in the application, Oracle manages it in the database. That translates to far more efficiency in managing database tenants.

More PDBs

Database 12.2c packs far more PDBs than the initial release; the limit goes from 252 PDBs up to 4,096 per instance. PDBs are clearly useful for customers seeking to consolidate their Oracle instances, and there may be a few that need to consolidate at such extreme scale. But for most enterprise customers, the benefits of the higher PDB density will be more indirect; they will benefit as customers of the Oracle Cloud, where Oracle is able to price and deliver database-as-a-service more cost-effectively and efficiently.

Oracle’s first database cloud service leveraging its multitenant architecture is Exadata Express Cloud Service. It offers Oracle 12.2c Enterprise Edition and most options running on the database-optimized Exadata infrastructure with 20GB of data and full 24x7 customer support for $175 per month. Exadata Express service is limited to scale to 50GB of data and one processor core, so it is positioned by Oracle for dev/test and departmental production workloads. Customers needing more power can upgrade to the Database Enterprise or Exadata Cloud services.

PDB "hot cloning" improves portability

Cloning is useful when trying to create new databases, such as test/development environments and/or product or test environments in the cloud. Database 12.2c allows PDBs to be cloned, refreshed, and relocated without taking the system down. Cloning capability is not new to Oracle, but the key to this release is that "hot cloning" mode is now enabled, thereby avoiding downtime.

PDB refresh allows cloned databases to stay in sync by refreshing updates (not the entire database) on the fly. PDB relocate allows a PDB to be physically moved from one database container to another with no application downtime. This feature is useful when shifting pluggable databases back and forth between different servers, data centers, and compute zones (in the cloud), and, of course, moving data from on-premise to the cloud (and vice versa). Such a feature could provide an alternative for "bursting" a single database from on-premise to the cloud to handle demand growth. This is not the same as bursting workloads to the cloud, however.

Related to cloning and relocating PDBs is managing database resources when consolidating multiple PDBs into a CDB. Database 12.2c applies more resource management features by enabling prioritization of available CPU, I/O, and memory resources that can align with business workload priorities.

Securing data in the cloud

Ironically, while security has often been cited as a concern for enterprises considering cloud deployment, increasingly it has become the case that cloud providers are rising to the challenge and are often providing even more robust, updated security measures compared to what enterprises implement on their own. Regardless of how secure cloud infrastructure is, many enterprises draw the line at key management for encryption; their policies require keys to be managed on-premise. Along with Database 12.2c, Oracle Key Vault 12.2 has also been updated to support this requirement. Specifically, it now accommodates management of encryption keys for cloud-deployed data on-
premise as long as there is a persistent secure shell network connection that is maintained between
the data center and the Oracle Public Cloud. This capability is critical for two reasons. First, it allows
organizations to manage encryption keys under one umbrella (as long as their cloud instances are in
the Oracle Public Cloud). Secondly, this capability is essential for organizations whose policies
stipulate that encryption be managed from inside the firewall.

**Introducing sharding**

Oracle is borrowing a feature popular in the NoSQL/internet distributed database world, and one that
is often associated with databases deployed in the cloud that must handle the extremes of internet-
scale traffic and distributed deployment.

Sharding allows a database to be horizontally partitioned to run across multiple nodes on a "shared
nothing" basis (each node is self-contained with its own compute and storage). It is the OLTP
equivalent of the shared-nothing, massively parallel (MPP) server architectures used for analytics.
Sharding tends to be associated with internet businesses whose online transaction systems must be
physically distributed to handle traffic; but it is also useful for high-availability/disaster recovery
scenarios or hybrid deployments that spread individual databases across multiple nodes, data
centers, and/or on-premise and cloud (discussed below). In Database 12.2c, Oracle supports up to
100 shards. The Oracle Public Cloud will fully automate sharding of Database 12.2c; by contrast,
sharding for on-premise installations of Database 12.2c (when it is released) will require some manual
effort.

Until now, Oracle's answer to scale-out has been Oracle RACs, a technology introduced roughly 15
years ago that involved a cluster manager (Oracle Clusterware) to act as traffic cop, allowing multiple
servers to share data storage and orchestrate reads and writes. Operationally, sharding provides a
much lower overhead approach; instead of having a cluster manager act as a hub to direct traffic,
data is apportioned to clusters and processing goes where the data is. But it does require changes to
the application layer so that workloads are automatically routed to the right shards containing the
pertinent data. RAC, on the other hand, is transparent to application workloads and is suitable for
workloads that cannot be sharded.

Oracle is not the first to offer sharding managed by the database (as opposed to the application or
middleware); it is a common feature of operational databases such as MongoDB, Cassandra, and
MySQL. But it is the first enterprise-class SQL database to do so.

**Operational enhancements**

**Less administrative overhead**

While simplicity is not a term that is normally associated with enterprise databases, the introduction of
multitenant pluggable databases in the 12c generation of Oracle Database has made some important
contributions in that direction. This benefit will especially affect organizations deploying multiple
databases that can now be consolidated under a common administrative umbrella. Configuration
steps normally performed by DBAs are largely eliminated after the first database has been deployed,
because all pluggable databases are managed "as one" in a virtualized container database (CDB)
and leverage the same settings for housekeeping functions such as security, replication, and
recovery.

© Ovum. All rights reserved. Unauthorized reproduction prohibited.
Enhancements for in-memory

In July 2014, the initial release of Oracle Database 12c introduced a memory-based, compressed column store for real-time analytics that sits alongside the disk-based row store (for OLTP). Database 12.2c adds a number of in-memory enhancements, including faster joins, expressions, and JSON, along with in-memory on standby databases and fast start. On Exadata storage, the Database 12.2c in-memory data store format and processing has been extended onto Exadata's flash cache storage, allowing for a much larger in-memory footprint. For large enterprise database deployments, tiering of "hot" (frequently used) and "cold" (rarely used) data is an important capability for optimizing the operational cost and performance of the database. Oracle Database 12c automated the process of moving data based on its temperature, but in the first release, only supported movement of data between different classes of disk. With Database 12.2c, automation now extends to memory; this is obviously critical for effective use of Database 12c's in-memory column store.

Comparison with other databases

Because in-memory processing has become practically a checklist item (Ovum views this as part of a larger trend where tiering is being extended from traditional storage media to silicon), it makes sense to compare notes with some of Oracle's rivals. Admittedly, there remain differences regarding which databases optimize in-memory for analytics or OLTP; most do the former, but it is not universal.

For instance, IBM BLU Acceleration for DB2 automatically tiers "hot" data from disk-based row storage and persists it as compressed, columnar in-memory storage. Teradata Intelligent Memory takes a similar approach, but it does not necessarily convert data from row to columnar; instead, it persists hot data in-memory in its original format, which could be row or logical columns. Meanwhile, Microsoft SQL Server 2016 supports in-memory for OLTP, but supports column store indexes for analytics in-memory. Nonetheless, SQL Server requires the DBA to designate the specially stored tables. Conversely, SAP HANA, which was designed as a 100% in-memory database, attacks the task from the opposite starting point of tiering cold data back to disk. SAP HANA offers dynamic tiering, but the tiering is controlled, not by the database, but by the application (or a stored procedure).

Extensibility

Extending multimodel support

Our report Hadoop, SQL, and NoSQL: No Longer an Either-Or Question identified a clear trend toward multimodel databases. Oracle has been no stranger to this trend, having supported a growing array of data types from text to rich media, XML, JSON, and spatial data. Database 12.2c improves the fidelity of JSON support and introduces a new graph engine.

Better-fidelity JSON

With the original release of Database 12c, Oracle increased support for heterogeneous data types; this is part of a general trend that Ovum has identified for databases becoming more hybrid in their capabilities. Oracle, and other household-brand SQL databases, have added JSON support in the past few years, but the JSON was of limited fidelity: JSON documents, which typically have complex hierarchical, nested structures, were usually flattened to single wide columns. Oracle treats JSON as
variable character (VARCHAR, CLOB, or BLOB) objects that would be accessed through "PATH" statements.

Highlights of JSON enhancements in Database 12.2c include:

- **In-memory support** – Attributes from JSON documents are extracted and stored as columns in-memory, providing the performance improvements that until now were restricted to SQL relational data.
- **JSON Data Guide** – This feature examines a collection of JSON documents, and infers a SQL "schema" view. While this feature does not change the core approach of Oracle (and most other relational databases) to make JSON look like SQL, it makes it easier for the SQL developer who would otherwise be unfamiliar with JSON formats.
- **New search expressions** – These expressions can drill down to individual members of a JSON array, providing access to details that might otherwise be obscured with traditional SQL-on-JSON approaches that flatten JSON data structures to single columns.
- **PL/SQL support** – This makes JSON more accessible to SQL developers. With the new release, PL/SQL adds several features and expressions for querying JSON data that support specific JSON object types, and perform joins of JSON documents and SQL tables.

Oracle is not alone in raising its game with JSON; IBM (DB2) and Teradata also store data in JSON format and allow access via API (which allows granular access to nested data) or using SQL via JDBC (which requires JSON documents to be flattened into a single column of a relational table). Snowflake, a new cloud-based data warehousing platform, differentiates by treating JSON data as a special data type that allows query of nested data. By contrast, Microsoft SQL Server only stores JSON as text, rather than as a special data type.

**Oracle Spatial and Graph**

Oracle combines geospatial and graph into the same data engine. For Database 12.2c, it adds support of PGX, a parallel graph technology first developed for Hadoop.

Graph engines are useful for representing complex many-to-many interrelationships. Any single node can have relationships (termed "edges") with different properties with one or more nodes. There are a number of open source and proprietary graph engines; however, while graph engines have become useful for representing entities with complex relationships, graph databases (e.g., Neo4J) have found only specialized markets because graph databases are complex to understand. But as an embedded engine powering use cases (e.g., charting the interrelationships among social networks tribes, the complex operations in the Internet of Things, or patterns of fraud, and providing the underlying models for master data management), graph technology is growing in use, as long as the practitioner does not have to manage data in it directly.

The PGX graph engine features improved performance; Oracle claims its proprietary graph engine, which is optimized for the Infiniband networks acting as the interconnect for Exadata, outperforms GraphX (a popular graph engine with the Spark computing engine) by up to 160x, thanks to its approach to partitioning compute (which is a challenge for graph problems). Although the Oracle graph engine is proprietary, it uses the open source TinkerPop interface, which provides a standard means of modeling graph domains, and a standard analytic language (Gremlin); so Oracle's implementation will look pretty standard when accessed through its Java APIs.
Partitioned external tables

As the data platform world has grown more multipolar, data will be scattered across multiple targets; the need arises for a means to query across all those targets. This is not a new idea, but the overhead of early approaches to federated query and data virtualization limited the take-up.

Oracle Big Data SQL provides one approach for querying Oracle Database and Hadoop with a query that projects Hadoop tables (Hive and HDFS) as virtualized external tables to Oracle database. With Database 12.2c, a complementary approach for partitioned external tables lets you persist virtualized views. Partitioned external tables let you map tables in Hive and HDFS to your Oracle database instance; a related feature is support of partition-pruning for faster query processing. While Big Data SQL provides this view on the fly for the query, the new partitioned external tables view makes it suited for materializing views in logical data warehouses.

And while we are on the topic of Big Data SQL, the new 3.0 version, released alongside database 12.2c, can now target any instance of Cloudera or Hortonworks Hadoop platforms. It is no longer restricted to running against Oracle's Big Data Appliance (which packages Cloudera Enterprise) and Exadata, making it more widely available to Oracle customers.

Approximate query processing

When conducting exploratory queries, it may not always be necessary to get the exact answer. The benefit is that getting a rough approximation takes less time because the query is not processed as thoroughly, or less data is targeted. Larger data sets compound the need for such capabilities; especially when data in Hadoop is involved. Commercial databases have long provided the option to query samples of data, but Database 12.2 uses sophisticated, dynamic statistical algorithms to compute the approximate query results. Database 12.2c adds the capability to shortcut queries by count, percentile, or median.

Appendix

Methodology

This report was compiled through discussions with Oracle and several reference customers, culminating with information from the analyst database summit conducted in conjunction with Oracle OpenWorld 2016.

Further reading

"Oracle’s cloud momentum nets new customers," IT0022-000784 (September 2016)

"Oracle's Big Data analytics portfolio gains critical mass," IT0014-003084 (December 2015)

"Oracle bets big on microservices but what will the payoff be in telecoms?" IT0012-000184 (November 2016)

"Oracle introduces new capabilities for the cloud infrastructure market," IT0022-000751 (October 2016)

"Oracle reaching critical mass with SaaS business," IT0014-003101 (February 2016)
The Cloud-First Strategy of Oracle Database 12c Release 2

Hadoop, SQL, and NoSQL – No Longer an Either-Or Question, IT0014-002937 (September 2014)

Author
Tony Baer, Principal Analyst, Information Management
tony.baer@ovum.com

Ovum Consulting
We hope that this analysis will help you make informed and imaginative business decisions. If you have further requirements, Ovum’s consulting team may be able to help you. For more information about Ovum’s consulting capabilities, please contact us directly at consulting@ovum.com.

Copyright notice and disclaimer
The contents of this product are protected by international copyright laws, database rights and other intellectual property rights. The owner of these rights is Informa Telecoms and Media Limited, our affiliates or other third party licensors. All product and company names and logos contained within or appearing on this product are the trademarks, service marks or trading names of their respective owners, including Informa Telecoms and Media Limited. This product may not be copied, reproduced, distributed or transmitted in any form or by any means without the prior permission of Informa Telecoms and Media Limited.

Whilst reasonable efforts have been made to ensure that the information and content of this product was correct as at the date of first publication, neither Informa Telecoms and Media Limited nor any person engaged or employed by Informa Telecoms and Media Limited accepts any liability for any errors, omissions or other inaccuracies. Readers should independently verify any facts and figures as no liability can be accepted in this regard – readers assume full responsibility and risk accordingly for their use of such information and content.

Any views and/or opinions expressed in this product by individual authors or contributors are their personal views and/or opinions and do not necessarily reflect the views and/or opinions of Informa Telecoms and Media Limited.
CONTACT US

www.ovum.com
analystsupport@ovum.com

INTERNATIONAL OFFICES

Beijing
Dubai
Hong Kong
Hyderabad
Johannesburg
London
Melbourne
New York
San Francisco
Sao Paulo
Tokyo