18c ORACLE Database

Oracle Sharding 18c

New Features

ORACLE WHITE PAPER | JULY 2018

ORACLE

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Introduction

When the Oracle Sharding feature was introduced in Oracle Database 12c Release 2, the focus was on internet scale applications that require linear scalability and fault isolation. These applications deal with 100s of millions or billions of users, massive concurrent user base and extremely large databases. With Oracle 18c, Oracle Sharding has been made more inclusive to applications and various other use cases by increasing its flexibility and strategies.

Here is the list of the Oracle Database 18c Sharding features:

1. User-defined Sharding
2. PDB Sharding
3. RAC Sharding
4. JSON & Spatial capabilities and Multi-shard query enhancements
5. Midtier Sharding

This white paper on Oracle Sharding 18c New Features is intended for Enterprise Architects, Database Architects, Database Administrators, Application Architects and those who are involved in the design and architecture of distributed database systems.

User-defined Sharding

In Oracle Database 12c Release 2, we introduced two sharding methods - *system-managed sharding and composite sharding*. System-managed sharding is based on partitioning by consistent hash. This sharding method randomly and evenly distributes data across shards and automatically redistributes it when shards are added to or removed from the sharded database. Consistent Hash is good for application where there are millions and even billions of values of the sharding key and it is not practical to manage them individually. This method enables linear scalability of transactions, concurrent users and the database capacity.

With composite sharding method, data is first partitioned by list or range (super_sharding_key) and then further partitioned by consistent hash (sharding_key). The two levels of sharding make it possible to map data to a set of shards, and then automatically maintain balanced distribution of data across that set of shards. Composite sharding is ideal for global data distribution where shards are placed in each geography and within a given geography data is uniformly distributed. Another use case for this method is - a set of powerful shards dedicated to Gold class users and set of low powered shards for Silver class users. Within each class of users, data is uniformly distributed.

But what if there are much less distinct values of the sharding key, maybe thousands, and the customer wants tighter control in mapping data to shards. For such cases, in Oracle Database 18c, we introduced *user-defined sharding* – a sharding method which is based on partitioning by RANGE or LIST and allows the user to explicitly specify mapping of data to shards. User-defined sharding is ideal for performance, regulatory, or other reasons when the user needs to store related data in the same shard and have full control on moving data between shards.

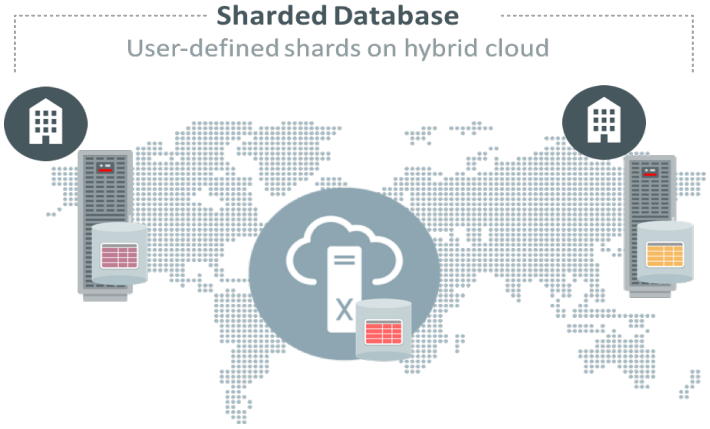

Figure 1. Oracle Sharded Database with User-defined sharding

User-controlled data distribution provides:

- Regulatory compliance - data can remain in the country of origin
- Hybrid clouds - some shards are on-premises and other shards in the cloud
- Cloud bursting - ability to move data from on-premises to the cloud during peak seasons
- Increased visibility into planned maintenance - when a shard needs to be brought down for maintenance, the administrator knows exactly which data will not be available.
- Each shard can have different hardware and HA configuration.
- More efficient range queries

With user-defined sharding, the user has the control to maintain balanced data distribution.

PDB Sharding

From Oracle Database 18c onwards, all Oracle databases will be container (CDB) or pluggable (PDB) databases by default. This PDB Sharding feature adds support for the ability to use PDBs as shards or catalog databases within the sharding architecture. PDB Sharding capability renders good manageability benefits for a sharded database. For example - consolidation of shards, manage many as one, database upgrades etc.

In Oracle Database 18c, we support shard as a single PDB in a given CDB. In addition to a single shard PDB, a CDB can contain other non-shard PDBs.

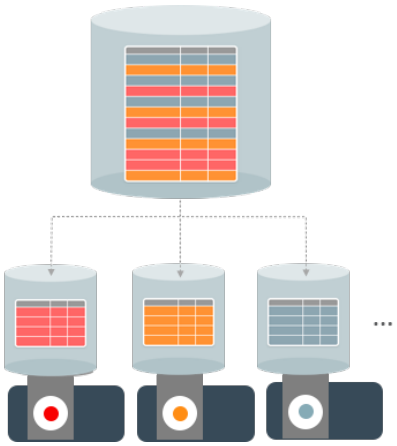

Figure 2. Oracle Sharded Database with pluggable databases as shards

RAC Sharding

RAC Sharding is a feature that enables to logically affinitize table partitions to Oracle RAC instances. This reduces block pings across instances while yielding better cache affinity.

This feature which is applicable to non-sharded RAC databases, takes advantage of direct routing API of Oracle Sharding. The application must use integrated Oracle clients such as Oracle Universal Connection Pool (UCP), Oracle Call Interface (OCI) Session Pool, Oracle Data Provider for .NET (ODP.NET) Connection Pool etc. Requests that specify sharding key are routed to the instance that logically holds the corresponding partition while the requests that don't specify sharding key still work transparently.

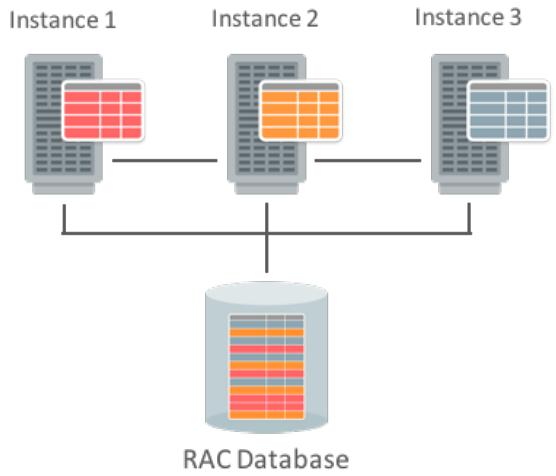

Figure 3. High-performance for shard-aware RAC applications with RAC Sharding feature

This capability empowers RAC with the performance and scalability of a Sharded Database with minimal application changes. The application just specifies the sharding key (as part of the connection check-out) for the most performance critical operations. The RAC Sharding feature is enabled on the RAC database by executing the following:

```
alter system enable affinity <TableName>;
```

No changes to the database schema are required. There is no requirement to deploy Sharding infrastructure as well.

Midtier Sharding

In Oracle Sharding 18c, we added the ability to affinitize midtiers with shards. This eliminates a bottleneck in Oracle Sharding 12c Release 2 that required each connection pool to establish a connection to all shards. In Oracle 18c Sharding, we have created "swim lanes" – the application tier, and (optionally) the web tier, are sharded in the same manner as the database.

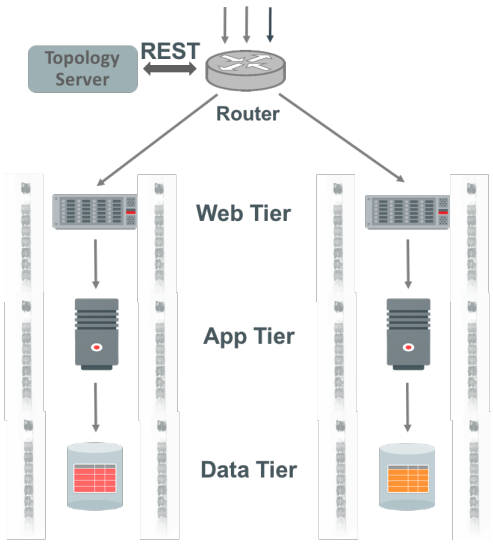

Figure 4. Swim-lanes across tiers via Midtier Sharding

To support this functionality, we provide a topology server, which has the Oracle sharded database (SDB) topology information, with a REST API. Customer's routing tier can make a REST API call to the topology server and enables routing of the request to the midtier associated with the sharding key. This API takes a value of a sharding key and returns the identifier of a swim lane to which this value belongs.

The "swim lane" architecture enabled via the Midtier Sharding:

- Provides even better fault isolation and scalability
- Reduces the number of database connections

- Improves midtier cache locality
- Eliminates chatty midtier-to-database connections across datacenters

Multi-shard Query Enhancements

Oracle Sharding 18c multi-shard queries (via proxy routing) have been enhanced to allow for composite and user-defined sharding methods as well as system managed sharding. For multi-shard queries, one can set different consistency levels by the initialization parameter MULTISHARD_QUERY_DATA_CONSISTENCY.

The query explain plan is also enhanced to display the information for all shards participating in the multi-shard query.

For centralized diagnostics, the SQL SHARDS() clause can be leveraged to query the V$, DBA/USER/ALL views, dictionary tables across all shards. For example:

```
SQL> select ora_shard_id, count(*) from shards(v$sql) where
ora_shard_id in (1,21) group by ora_shard_id;

ORA_SHARD_ID   COUNT(*)
------------ ----------
           1        746
          21        738
```

JSON, LOBs & Spatial Capabilities

Oracle Sharding 18c release enables JSON operators that generate temporary LOBs, large JSON documents (requiring LOB Storage), Spatial Objects, Index and Operators and Persistent LOBs to be used in a sharded database.

Summary

Oracle Database 18c with Oracle Sharding is a globally distributed multimodel (relational and document) cloud-native (and on-premises) DBMS. It is built on shared-nothing architecture in which data is horizontally partitioned across databases that share no hardware or software. It provides linear scalability, fault isolation, and geographic data distribution for shard-amenable applications. Oracle Sharding does all this while rendering the strong consistency, full power of SQL and the Oracle Database ecosystem. You can deploy a sharded database and incrementally add shards to elastically scale your transactions, database capacity, and concurrent users. Oracle Sharding 18c new features – (such as user-defined sharding, RAC sharding, Oracle Multitenant support, midtier sharding) now allow the expansion of the applicability of Sharding to various other customer use cases.

References

1. Oracle Sharding OTN Page – http://www.oracle.com/goto/oraclesharding

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

blogs.oracle.com/oracle
facebook.com/oracle
twitter.com/oraclesharding
oracle.com/goto/oraclesharding

Hardware and Software, Engineered to Work Together