Oracle Database Native Sharding

Technical Deep Dive [CON6532]

Srinagesh Battula, Mark Dilman
Oracle Product Development

Sudhi Vijayakumar,
Facebook DB Architect

Sept 20, 2016

Accelerate Your Digital Transformation in the Cloud
Safe Harbor Statement

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.
Announcing Oracle Database 12c Release 2 on Oracle Cloud

• Available now
  – Exadata Express Cloud Service

• Coming soon
  – Database Cloud Services
  – Exadata Cloud Machine

Oracle is presenting features for Oracle Database 12c Release 2 on Oracle Cloud. We will announce availability of the On-Prem release sometime after Open World.
Program Agenda

1. Sharding Overview
2. A Customer’s Perspective – Facebook
3. Sharded Database and Schema Creation
4. Data Dependent Routing
5. Demo
Program Agenda

1. Sharding Overview
2. A Customer’s Perspective – Facebook
3. Sharded Database and Schema Creation
4. Data Dependent Routing
5. Demo
Oracle Sharding – An Elastic Database Architecture

- Horizontal partitioning of data across up to 1000 independent Oracle Databases (shards)
- Shared-nothing hardware architecture
  - Each shard runs on commodity server
  - No shared storage
  - No clusterware
- Data is partitioned using a sharding key (i.e. account_id)

A single logical DB sharded into N physical Databases

Server A

Server B

Server N

Shard01:
Table 1

Shard02:
Table 1

Shardnn:
Table 1

Partitions

A
B

E
F

G
H

C
D

... ...

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.
Oracle Database Sharding – Benefits

**Linear Scalability**
Add shards online to increase database size and throughput. Online split and rebalance.

**Extreme Availability**
Shared-nothing hardware architecture. Fault of one shard has no impact on others.

**Geographic Distribution**
User defined data placement for performance, availability, DR or to meet regulatory requirements.
Native Database Sharding in Oracle – Capabilities

• Auto deployment of up to 1000 shards
  – Supports Data Guard and Oracle GoldenGate

• Multiple sharding methods
  – System Managed (Consistent Hash)
    • Uniform data distribution across shards → linear scalability
  – Composite – (Range-Consistent Hash or List-Consistent Hash)
    • Geo-distribution plus linear scalability

• Centralized schema maintenance
  – Native SQL for sharded and duplicated tables

• Routing
  – Direct routing from Connection pools
  – Proxy routing for Multi-shard queries

• Lifecycle Management
  – Elastic scaling with automatic rebalance
  – Shards may run different Oracle releases
Sharding – a Different Way to Scale

Frictionless linear scaling due to zero shared hardware or software

![Graph showing TPS vs. number of shards](image-url)
Sharding for Extreme Data Availability

Fault Isolation

1%

The portion of users who undergo brown-out at one time by an unplanned outage or database upgrade in a sharded database with 100 shards

Oracle database on commodity hardware
Sharding – a Superior Way to Distribute Data
Data Sovereignty and Data Proximity

Sharded Database
Linear Scalability & Geodistribution

Customers Americas

Customers Europe

Customers Asia

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. | Oracle Confidential – Internal/Restricted/Highly Restricted
Sharding – a Superior Way to Deploy Database
Flexible On-Premises or Cloud Deployment

On-Premises  Hybrid  Cloud
Automated Data Availability Configuration

Active Data Guard with Fast-Start Failover

GoldenGate ‘chunk-level’ active-active replication with automatic conflict detection/resolution (OGG 12.3)

Optionally – complement replication with Oracle RAC for server HA
Active-Active Sharded Database – Oracle GoldenGate 12.3

- Automatic GoldenGate deployment
  - Chunk-level replication within and across shardgroups

- Automatic CDR (conflict detection & resolution) to simplify active-active replication
  - Including insert-delete conflicts, JSON columns, LOB updates

- Shardgroups can have different number of shards, replication factor, database versions and H/W platforms
Data Modeling Requirements

Table Family and Duplicated Tables

• Hierarchical table family that consists of a root table and many child and grandchild tables

• Every table in the table family must contain the sharding _key
  – Sharding _key must be the leading column of the primary key of the root table

• The sharding method and sharding_key are determined based upon application and business requirements
  – Plan carefully...changing the method or key requires rebuilding of sharded database

• Identify tables that need to be duplicated on all shards
  – Common reference data (e.g. Products)
## Application Requirements

### Direct Routing
- OLTP workloads that require linear scale, high performance, and fault isolation must specify a `sharding_key` during connect
  - All data required for a transaction is contained within a single shard

### Proxy Routing
- The following workloads must be able to accept lesser SLAs for linear scale, performance, and fault isolation than OLTP transactions
  - Reporting workloads
  - Multi-shard queries
  - Applications unable to specify a `sharding_key`

### Database Connections
- Must use Oracle integrated connection pools (UCP, OCI, ODP.NET, JDBC)
- Must be able to separate workloads that use Direct Routing from those that use Proxy Routing
  - Each uses separate connection pools
Program Agenda

1. Sharding Overview
2. A Customer’s Perspective – Facebook
3. Sharded Database and Schema Creation
4. Data Dependent Routing
5. Demo
Oracle Database Native Sharding

Sudhi Vijayakumar
Database Systems Architect, Facebook Inc
Agenda

• Introduction
• Usage of Oracle in Facebook
• Current Challenges
• Application Use Case
• Evaluation
Facebook

Founded in 2004, Facebook's mission is to give people the power to share and make the world more open and connected. People use Facebook to stay connected with friends and family, to discover what's going on in the world, and to share and express what matters to them.
Who am I?

- 15+ Years in the industry
- Contributions to best selling books
- Metallica, Hans Zimmer and Baby Lullaby fan
- Voracious reader
- Productivity maniac
- Photographer
- www.facebook.com/kaddisudhi
- www.Kaddisudhi.com
- Attended Sharding Preview conducted by Oracle Beta Program Office
Oracle Usage in Facebook

[Diagram showing Oracle DB connected to various software components such as EBS, ADF, OSB, Hyperion, CCG, WCC, Agile, OBIEE, BPM, Apex, ePIS, APRO, Vertex, GRC, ODS, Scuba, and Hive.]
Current Challenges

- Monolithic architecture
- Custom requirements for our DC
- Painful deployment procedures
- Limitations for scaling
- Datasize growth
Benefits of Oracle Sharding

- Full power of Oracle RDBMS
- Deployments can be made automatic
- Open Compute Platform can be used for deployments
- Horizontal scaling based on demand/growth
- Reduce SPOF
- Reduce tech debt (apply patches in rolling manner, etc.)
- Uses familiar technology (Data Guard, GoldenGate)
How does data get into Oracle?

- Operational Data Store (ODS)
- 2 Billion Time Series Counters
- Scuba is slice and dice store
- In-memory data store
- High rate of ingestion
- Hive is used for warehouse
- They are not RDBMS
- Do not serve users and not ACID
Facebook Data Center

- Spread across the world
- Unique cooling tech
- Smaller power footprint
- Design based on Open Compute Project (OCP)
Application: DC Analytics

- Site Ops folks
- Key Metric areas
- Drill-down reports
- Trend Analysis
Workload Pattern

• Application specifies Sharding key
• Enables connection checkout
• Direct routing
• Limit cross-shard queries
Target Sharded Architecture

Various Data Sources

- Shard Group 1
- Shard Group 2
- Shard Group 3
Key Metrics for Oracle Sharding POC

- Support Facebook’s existing and growing ingestion rate
- Response times for queries
- Ease of Deployment
- Ease of Maintenance
- Iterative design changes
Program Agenda

1. Sharding Overview
2. A Customer’s Perspective – Facebook
3. Sharded Database and Schema Creation
4. Data Dependent Routing
5. Demo
Deployment of a System-Managed SDB with Data Guard

Region Availability_Domain1

Clients

Connection Pools

Shard Director shdir1,2

Shard Catalog shardcat

Region Availability_Domain2

Connection Pools

Shard Director shdir3,4

Shard Catalog shardcat_stdby

Shardgroup shgrp1

Primaries...

HA Standbys...

Shardgroup shgrp2

Data Guard Fast-Start Failover
Declarative Specification of SDB Configuration

Primary components

create shardcatalog -database <shardcat host>:1521:shardcat -user sdb_admin/passwd_sdb_admin
add gsm –gsm sharddirector1 -listener 1571 -pwd passwd_gsmcatuser -catalog <shardcat host>:1521:shardcat ...
add credential -credential oracle_cred -osaccount oracle -ospassword < >
add shardgroup -shardgroup shgrp1 -deploy_as primary -region Availability_Domain1
create shard -shardgroup shgrp1 -destination <host1> -credential oracle_cred ...

Duplicated components for HA deployed in separate availability domain
add gsm –gsm sharddirector2 -listener 1571 -pwd passwd_gsmcatuser -catalog <shardcat host>:1521:shardcat ...
add shardgroup -shardgroup shgrp2 -deploy_as active_standby -region Availability_Domain2
create shard -shardgroup shgrp2 -destination <host11> -credential oracle_cred ...

deploy
add service -service oltp_rw_srvc -role primary
Schema Creation – Sharded and Duplicated Tables

Database Tables

Customers

<table>
<thead>
<tr>
<th>Customer</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>Mary</td>
</tr>
<tr>
<td>456</td>
<td>John</td>
</tr>
<tr>
<td>999</td>
<td>Peter</td>
</tr>
</tbody>
</table>

Orders

<table>
<thead>
<tr>
<th>Order</th>
<th>Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>4001</td>
<td>123</td>
</tr>
<tr>
<td>4002</td>
<td>456</td>
</tr>
<tr>
<td>4003</td>
<td>999</td>
</tr>
<tr>
<td>4004</td>
<td>456</td>
</tr>
<tr>
<td>4005</td>
<td>456</td>
</tr>
</tbody>
</table>

Line Items

<table>
<thead>
<tr>
<th>Customer</th>
<th>Order</th>
<th>Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>4001</td>
<td>40011</td>
</tr>
<tr>
<td>123</td>
<td>4001</td>
<td>40013</td>
</tr>
<tr>
<td>456</td>
<td>4004</td>
<td>40014</td>
</tr>
<tr>
<td>999</td>
<td>4003</td>
<td>40015</td>
</tr>
<tr>
<td>999</td>
<td>4003</td>
<td>40016</td>
</tr>
</tbody>
</table>

Products

<table>
<thead>
<tr>
<th>SKU</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Coil</td>
</tr>
<tr>
<td>101</td>
<td>Piston</td>
</tr>
<tr>
<td>102</td>
<td>Belt</td>
</tr>
</tbody>
</table>

Shard 1

Shard 2

Shard N

Sets of related partitions of customers, orders & line items placed on a given tablespace[s]

Duplicated Tables
Mapping Related Partitions to a Chunk

Chunk is the Unit of Data Movement in a Sharded Database

- In simplest form, each chunk is a tablespace containing related partitions of the sharded tables, i.e. Chunk#1 contains Customers_P1, Orders_P1, etc
  - A chunk can also be a set of tablespaces that store related partitions – a separate tablespace for each partition of the table

- All data pertinent to a given sharding key resides in a single chunk
Contents of a Shard
Chunks of Sharded Tables and Full Copies of Duplicated Tables

- Chunk #1 Partitions:
  - Customers_P1 (1-1M)
  - Orders_P1
  - Lineitems_P1

- Chunk #60 Partitions:
  - Customers_P60 (59M-60M)
  - Orders_P60
  - Lineitems_P60

Products (Duplicated Table)

Shard 1
System Managed Sharding

- Based on partitioning by CONSISTENT HASH
- Range of hash values assigned to each chunk
- Data is uniformly distributed / re-sharded automatically
- Many relatively small equally sized chunks

+ Linear Scalability
+ Automatic balanced data distribution
- User has no control on location of data

- Customer facing applications sharded by customer_id
- For linear scalability
Tablespace Set

- **CREATE TABLESPACE SET tbs1;**
  - Automates creation of tablespaces and distributes them among shards
  - Number of tablespaces = Number of chunks
- Supports all tablespace clauses
Creating Sharded and Duplicated Tables

DDLs are executed on the Shard Catalog

CREATE TABLESPACE SET tbs1;

CREATE SHARED TABLE Customers
( CustId VARCHAR2(60) NOT NULL,
  FirstName VARCHAR2(60),
  LastName VARCHAR2(60),
  ...,
  CONSTRAINT pk_customers
        PRIMARY KEY(CustId)
)
PARTITION BY CONSISTENT HASH (CustId)
PARTITIONS AUTO
TABLESPACE SET tbs1;

CREATE DUPLICATED TABLE Products
( ProductId INTEGER PRIMARY KEY,
  Name VARCHAR2(128),
  LastPrice NUMBER(19,4),
  ...
) TABLESPACE products_tsp;

CREATE SHARED TABLE Orders
( OrderId INTEGER,
  CustId VARCHAR2(60),
  OrderDate TIMESTAMP,
  ...,
  CONSTRAINT pk_orders
        PRIMARY KEY (CustId, OrderId),
  CONSTRAINT fk_orders_parent
        FOREIGN KEY (CustId) REFERENCES Customers(CustId)
)
PARTITION BY REFERENCE (fk_orders_parent);

Automatically maps partitions to the tablespaces that were created.
Oracle Sharding for Document Store Applications

Supports Schemaless Application Development with JSON

```
CREATE TABLESPACE SET tbs1;

CREATE SHARDED TABLE Customers
( CustId VARCHAR2(60) NOT NULL,
  CustProfile CLOB,
  CONSTRAINT pk_customers PRIMARY KEY(CustId),
  CONSTRAINT cons_json CHECK (CustProfile IS JSON)
)
PARTITION BY CONSISTENT HASH (CustId)
PARTITIONS AUTO
TABLESPACE SET tbs1;
```

- Full power of SQL to JSON document stores plus linear scalability of SDB
Composite Sharding

• Provides two-levels of sharding
• The sharded database is divided into N sets of shards called shardspaces
  – Data is partitioned across shardspaces by LIST or RANGE on super-sharding key (e.g. geography)
  – Within each shardspace, data is partitioned across shards by CONSISTENT HASH using sharding key (e.g. customer id)

+ Best practice for geo-distribution or hybrid clouds with linear scalability

- Requires two sharding keys: super_sharding_key and sharding_key

➤ Billing system sharded by geo then by account_id
➤ For geo-distribution and linear scaling
Composite Sharding – Example

ADD SHARDSPACE –SHARDSPACE Americas –CHUNKS 120;

CREATE SHARD –SHARDSPACE Americas
    –DESTINATION <host1> –CREDENTIAL oracle_cred;
CREATE SHARD –SHARDSPACE Americas
    –DESTINATION <host2> –CREDENTIAL oracle_cred;
DEPLOY;
CREATE TABLESPACE SET tbs1 IN
    SHARDSPACE Americas;

ADD SHARDSPACE –SHARDSPACE Asia –CHUNKS 180;

CREATE SHARD –SHARDSPACE Asia
    –DESTINATION <host3> –CREDENTIAL oracle_cred;
CREATE SHARD –SHARDSPACE Asia
    –DESTINATION <host4> –CREDENTIAL oracle_cred;
CREATE SHARD –SHARDSPACE Asia
    –DESTINATION <host5> –CREDENTIAL oracle_cred;
DEPLOY;
CREATE TABLESPACE SET tbs2 IN
    SHARDSPACE Asia;
Composite Sharding - Example
Map Ranges of Sharding Keys to Partitions

CREATE SHARDED TABLE customers
( CustId VARCHAR2(60) NOT NULL,
    Geo VARCHAR2(8),
    CONSTRAINT cust_pk
        PRIMARY KEY(CustId)
)
PARTITIONSET BY LIST (Geo)
PARTITION BY CONSISTENT HASH (CustId)
PARTITIONS AUTO
(PARTITIONSET Americas VALUES (‘Americas‘)
    TABLESPACE SET tbs1,
    PARTITIONSET Asia VALUES (‘Asia‘)
    TABLESPACE SET tbs2) ;
Program Agenda

1. Sharding Overview
2. A Customer’s Perspective – Facebook
3. Sharded Database and Schema Creation
4. Data Dependent Routing
5. Demo
Routing in an Oracle Sharded Environment

• **Direct Routing** based on sharding_key
  – For workloads that specify sharding_key (e.g. customer_id) during connect
    • A transaction happens on a single shard
  – Enabled by enhancements to mid-tier connection pools and drivers
    • JDBC/UCP, OCI, and ODP.NET recognize sharding keys
      – PHP, Python, Perl, and Node.js leverage OCI support for sharding

• **Proxy Routing** via coordinator (shard catalog)
  – For workloads that cannot specify sharding_key as part of connection
  – Applies to reporting, batch jobs
  – Queries can span 1 or more or all shards
    • Performed in parallel across shards (e.g. aggregates on sales data)
UCP: Connection Request with a Shard Key

An Example of New Sharding APIs

• Pool creation: no Java application code change
• Shard-aware application gets a connection, using UCP Sharding APIs

OracleShardingKey keyMaryEmail =
   pds.createShardingKeyBuilder()
   .subkey("mary.smith@xyz.com", OracleType.VARCHAR2)
   .build();

Connection connection =
   pds.createConnectionBuilder()
   .shardingKey(keyMaryEmail)
   .build();
Session-Based Routing via Sharding Key

- Sharding key is provided by the applications at connection checkout.
- Shard Director looks up the sharding key and redirects client to the correct shard containing the data.
- Client executes SQL directly at shard.
- After all transactions for the given sharding key are executed, application must check-in the connection to the pool and check-out a new connection for another key.
Direct Routing via Sharding Key

**Fast Path for key-based access**

- Connection pool maintains the shard topology cache
  - Upon first connection to a shard
    - Connection pool retrieves all sharding key ranges in the shard
    - Connection pool caches the key range mappings
- DB request for a key that is in any of the cached key ranges goes directly to the shard (i.e., bypasses shard director)
Proxy Routing

Non-sharding Key Access & Multi-shard Queries

• Applications connect to Query coordinator/Shard Catalog
• Single Shard Queries
  – Coordinator parses SQL, extracts sharding key and routes query to the correct shard
• Multi Shard Queries
  – Coordinator rewrites the query so that most processing is done on the shards
  – Supports shard pruning and scatter-gather
  – Final aggregation performed on the coordinator
• For developer convenience, not for high performance
• Limited to System Managed Sharding in 12.2.0.1
Centralized Schema Management

connect to GDS$CATALOG service
alter session enable shard ddl;
create tablespace set ...
create user ...
create sharded table ...
...
Create duplicated table ...

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. |
Duplicated Table Maintenance

Master Table

Shard Catalog

Shard 1

Shard 2

Shard 3

Read-Only Materialized Views
Online Addition and Rebalancing of Shards

- A chunk is the unit of data movement when resharding.
- Chunks are automatically rebalanced when a new shard is added.
- Uses RMAN Incremental Backup & Transportable Tablespace.
Online Addition and Rebalancing of Shards

- A chunk is the unit of data movement when resharding
- Chunks are automatically rebalanced when a new shard is added
- Uses RMAN Incremental Backup & Transportable Tablespace
Lifecycle Management of SDB

- DBA can manually move or split a chunk from one shard to another
- When a new shard is added, chunks are automatically rebalanced
  - Uses RMAN Incremental Backup & Transportable Tablespace
- Connection pools get notified (via ONS) about split, move, add/remove shards, auto-resharding, read-only
  - Application can either reconnect or access read-only
- All MAA practices apply for BUR of shards and shard catalog with special considerations
- Can patch all shards with one command via opatchauto
- EM supports monitoring & management of SDB
Program Agenda

1. Sharding Overview
2. A Customer’s Perspective – Facebook
3. Sharded Database and Schema Creation
4. Data Dependent Routing
5. Demo
Oracle Sharding Demo on Oracle Bare Metal Cloud

• Objective
  – Elastically scale-out the Shared Database
    • Demonstrate linear scalability of Relational transactions
  – Achieve highest availability with MAA on Oracle Bare Metal Cloud

• Infrastructure
  – Each shard hosted on dedicated server

<table>
<thead>
<tr>
<th>Component</th>
<th>Resources per shard</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>36 Cores</td>
</tr>
<tr>
<td>Memory</td>
<td>512 GB</td>
</tr>
<tr>
<td>Flash</td>
<td>12.8 TB NVMe</td>
</tr>
<tr>
<td>Network</td>
<td>10 GbE</td>
</tr>
</tbody>
</table>

Demo Topology

Oracle Bare Metal Cloud

Availability Domain 1
(100 Primary Shards)

Shard-level Active Data Guard
(Max Availability)

Availability Domain 2
(100 HA Standby Shards)
Oracle Sharding Demo on Oracle Bare Metal Cloud

Conclusions

• Elastically scaled-out to 200 shards on Oracle Bare Metal Cloud
  – Demonstrated linear scalability of Relational transactions
  – Demonstrated **11 Million transactions per sec** that includes:
    • 4.5 Million Read-Write Transactions per sec across all 100 Primary shards
    • 6.5 Million Read-Only Transactions per sec across all 100 Active Standby shards

• MAA Sharding provides highest availability
  – Each shard is protected by **Data Guard Fast-Start Failover** across Availability Domains
  – Single Shard Failure resulted in 100% availability for 99% of the application
    • 1% of the application experienced only 15 seconds blackout
Oracle Native Database Sharding
Combines the best of RDBMS and NoSQL

• Multi-modal database – Relational, JSON, XML, etc.
• SQL and programmatic interfaces - PL-SQL, OCI, JDBC, etc.
• Schema flexibility & developer agility with JSON within Oracle Database
• Better consistency than NoSQL databases
• Easier application maintenance – schema in database instead of application
• Enterprise-class features of a mature DBMS
• Leverage in-house and world-wide Oracle DBA skillset
• Plus superlinear scalability & extreme availability
Oracle Native Database Sharding

• Complete platform for sharding an Oracle database
• Ideal for mission-critical transaction processing that require a higher level of scalability and availability than is possible with a single physical database
Oracle Sharding Sessions and Demos in OOW 2016

**Monday, Sept 19**
- 12:30 PM: High Availability and Sharding Deep Dive with Next-Generation Oracle Database [CON6183], Moscone South-102
- 5.30 PM: Deep Dive into Oracle GoldenGate 12.3 for Oracle Database 12c Release 2 [CON6555], Moscone South-102
- 5.30 PM: Best Practices for Application Performance and Scalability [CON6538], Park Central - Metropolitan

**Tuesday, Sept 20**
- 12.15 PM: Oracle Database 12c Release2 JDBC and UCP Features and Best Practices [CON6454], Park Central - Olympic
- 4:00 PM: Oracle Database Sharding Technical Deep Dive with Oracle Development [CON6532], Moscone South-102

**Wednesday, Sept 21**
- 11:00 AM: Oracle Maximum Availability Arch Best Practices For Oracle Database 12c [CON6537], Moscone South-102

**Thursday, Sept 22, Park Central – Concordia**
- 1.15 PM: What’s New for .NET Developers in Oracle Database [CON6577], Park Central - Concordia

**Demos**

- **Sharding Demo Booth:**
  - SDB-032, Database High Availability, Moscone South Demogrounds
- **Sharding Demos on Oracle Bare Metal Cloud**
  - Cloud Infrastructure Booth, Outside Keynote Hall
  - “Supercharge your database with Oracle IaaS platform” Demo Pod, Main Expo Hall

**Contact:**

Oracle Sharding Product Management

Nagesh Battula  srinagesh.battula@oracle.com

Nick Wagner  nick.wagner@oracle.com
Integrated Cloud
Applications & Platform Services