

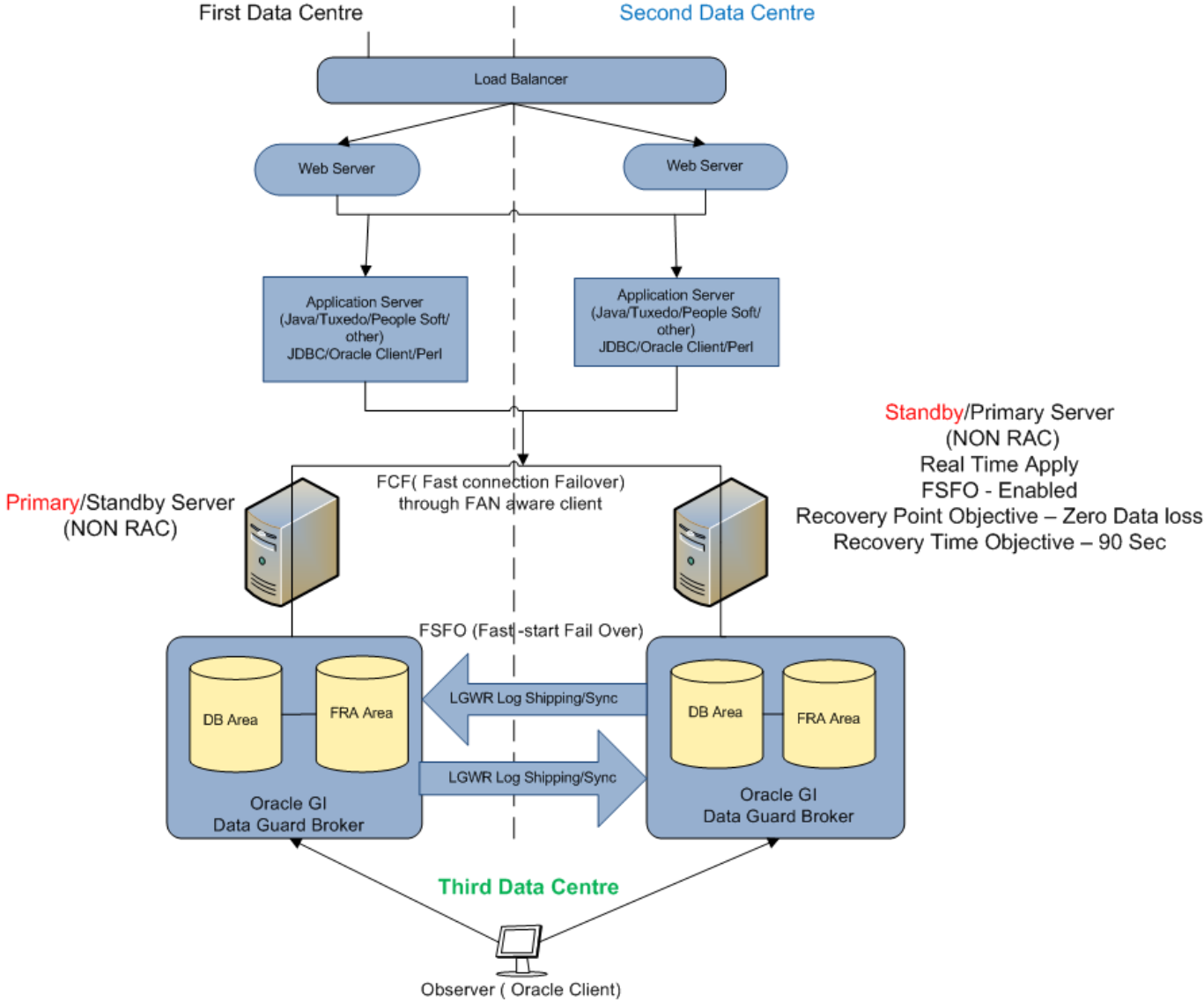
# **Oracle in Maximum Availability Architecture**

**GLOBAL DATA SERVICES, DATA GUARD 12C WITH STANDALONE DATABASE**

# Agenda

- Overview old database architecture and challenges faced
- New oracle database architecture
- Global Data Services (GDS) - Overview
- GDS Deployment
- GDS Configuration – GSM, Catalog, Region and Global Services
- GDS – Test Cases
- GDS Restrictions and Licence
- Lessons Learned
- Future plans

# GI Upgrade – DB/Apps Architecture

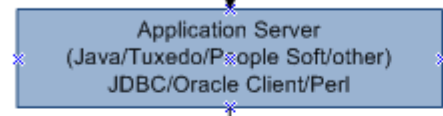


## The Challenges Compare GI vs GDS

- Per database server GI installation/Setup –Central Server
- Manage services in each database server – Manage central
- Service Location on each database server – Central service management
- Service Location transparency to client – total server transparent
- Connection string to individual db server – Can be unique except service name
- Region between database servers – Can be define globally

First Data Centre

Second Data Centre



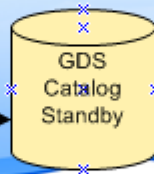
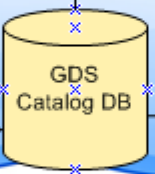
Oracle Global Data Services

All GDS Databases connected to all GSM's

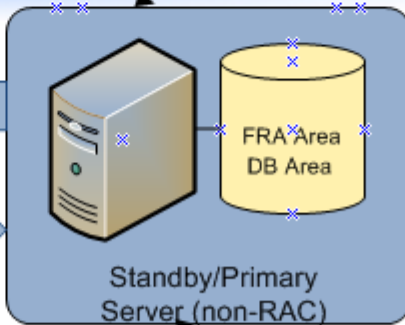
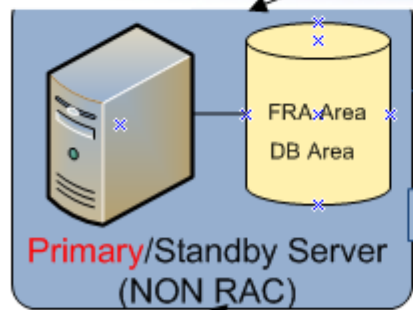
GSM 1



GSM 2



FCF through FAN aware client

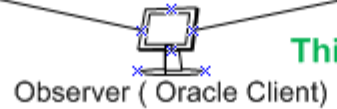


LGWR Log Shipping/Sync

LGWR Log Shipping/Sync

Real Time Apply  
FSFO - Enabled  
RPO - Zero Data loss  
RTO - 90 Sec

Third Data Centre



# Global Data Services Overview

GDS is a complete automated workload management solution for replicated databases (using replication technologies such as Active Data Guard, Oracle Golden Gate etc.).

- Centralized Service Management
- Workload Routing ( Region and Lag bases)
- Inter- Database service failover
- Role based services
- Load Balancing ( connect time and Run- time)

# What GDS is and is not????

- GDS does not itself perform data replication
- GDS does not perform database failovers
- GDS does not replace the functionality of Oracle Clusterware and Oracle RAC
- Oracle Maximum Availability Architecture (MAA) validation will be performed with Active Data Guard and Oracle GoldenGate
- For connect- time load balancing, GDS supports all clients. For Run- time load balancing, GDS supports the connection- pool based 12.1 clients such as JDBC,OCI,ODP.NET, WebLogic. JDBC 11.2 clients are supported but with some limitations

# GDS Configuration in a Nutshell

- Global Service Manager (GSM)
  - Regional listener to the incoming database connections
  - Perform connect time/Run time load balancing
  - Inter – database Service failover and Management
- GDS Catalog
  - Stores GDS Configuration meta data
- GDSCTL
  - CLI to admin GDS configs
- GDS Pool
  - Database that offer a common set of global Services
- GDS Region
  - group of databases and clients in close N/w ( east, west)
- Global Service
  - DB service provided by Multiple db's with replicated database



# GDS Deployment

- Install GSM software on GSM Servers
  - Minimum of 1 GSM per region (recommend 3 GSM/region)
- Setup GDS Administrator accounts & privileges
  - Connect to container as sysdba (GSM catalog database)
  - Create user c##gsmadmin
  - Create gsmadmin\_role to c##gsmadmin
- Configure GDS
  - Create GDS Catalog
  - Add GSM's, regions, Pool, Databases and Global Services
- Set up Client Connectivity

# GDS - Catalog & GSM

- Configure GDS
- Create GDS catalog
  - GDSCTL> connect c##gsmadmin/<pwd>@GSM
  - GDSCTL> create catalog -database GSM -userC##gsmadmin/<pwd>
- Add GSMs, Regions, Pools, Databases, Global Services
  - ALTER USER gsmcatuser identified by <pwd> ACCOUNT UNLOCK; (on db & GSM)
  - GDSCTL> connect gsmcatuser/<pwd>@GSM
  - GDSCTL> add gsm -gsm TDC\_GSM1 -listener 1522 -catalog GSM -pwd <pwd>
  - GDSCTL> start gsm -gsm TDC\_GSM1

# GDS - Region & Pool

- Connect to GDS catalog

```
GDSCTL> connect gsmcatuser/<pwd>@GSM
```

- Adding GDS Region

```
GDSCTL> add region -region NZ (north,south)
```

```
GDSCTL> config region
```

```
GDSCTL> remove region -region NZ
```

- Adding GDS Pool

```
GDSCTL> add gdspool -gdspool CS_POOL
```

```
GDSCTL> config gdspool
```

```
GDSCTL> remove gdspool -gspool CS_POOL
```

# GDS - DB or DGBroker

- Adding a Database to GDS Pool (standalone connection)

```
GDSCTL> add database -connect -region NZ -gdspool CS_POOL
```

- Adding a Dgbroker to GDS Pool ( for High Availability)

```
GDSCTL> add brokerconfig -connect CSDB - gdspool CS_POOL -pwd <gsmuser> -region NZ
```

```
GDSCTL> remove brokerconfig -gdspool CS_POOL
```

```
GDSCTL> synchronize brokerconfig -gdspool CS_POOL
```

# GDS – Global Service

- Adding service for standalone Database (single instance)

```
GDSCTL> add service –service CSDB –gdspool CS_POOL options: - –preferred_all - role  
GDSCTL> start service –service CSDB –gdspool CS_POOL
```

- Adding a dgbroker to GDS Pool ( for High Availability – Role based)

```
GDSCTL> add service –eservice CS_HA – gdspool CS_POOL –role PRIMARY  
–notification TRUE –fialovertime SELECT –fialovermethod BASIC –pdbname <PDB>  
GDSCTL> start service –service CS_HA –gdspool CS_POOL  
GDSCTL> stop service –service CS_HA –gdspool CS_POOL  
GDSCTL> remove service –service CS_HA –gdspool CS_POOL
```

# GDS – Global Service options

- Lag Tolerance (-lag in sec/ANY)

- It specifies the maximum lag that a provider of this service may have.
- keyword ANY indicate that there is no upper limit

```
GDSCTL> add service -esrvce CS_HA - gdspool CS_POOL -role physical standby -lag 180
```

- Locality ANYWHERE (-locality ANYWHERE | LOCAL\_ONLY)

- Client connections and work requests are routed to any region for load balancing or failover(ANYWHERE)
- Regardless of load, GDS will not route to databases in other regions (LOCAL\_ONLY – region\_failover)

```
GDSCTL> add service -esrvce CS_HA - gdspool CS_POOL -role physical standby -  
locality ANYWHERE
```

```
GDSCTL> add service -esrvce CS_HA - gdspool CS_POOL -role physical standby -  
locality LOCAL_ONLY - region_failover
```

# GDS – Global Service options

- CLBGOAL (-clbgoal SHORT | LONG)
  - GDS supports connect-time load balancing for all clients
  - SHORT for run-time load balancing
  - LONG for long running connections such as batch jobs
  
- RLBGOAL (load balancing)
  - Run-time load balancing Goal
  - It support SERVICE\_TIME balance or THROUGHPUT to balance the connection

# GDS - Client Connections

- Client connections can be define different passions
- Configuring Client Failover using OCI / JDBC / Perl & SQL \*Net

Option :-1

Its direct connection to database server using GDS service

```
AUXXPXD_S=
(DESCRIPTION_LIST=
  (LOAD_BALANCE=off)
  (FAILOVER=on)
  (DESCRIPTION =
    (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcp)(HOST=oraxxx01.db.auckland.ac.nz)(PORT=1521))
      (CONNECT_DATA=(SERVICE_NAME =
DB_NAME_HA.CS_POOL.oracloud))
    )
  )
)
```

Note: the HOST is physical database host and database listener port



# GDS - Client Connections

Option :-2

Its a connection through GSM to database

```
AUXPRD_GSM=
(DESCRIPTION_LIST=
  (FAILOVER=on)
  (DESCRIPTION =
    (CONNECT_TIMEOUT=20)(RETRY_COUNT=5)
    (ADDRESS_LIST=
      (LOAD_BALANCE=on)
      (ADDRESS=(PROTOCOL=TCP)(HOST=GSM01.its.auckland.ac.nz)(PORT=1522))
      (ADDRESS=(PROTOCOL=TCP)(HOST=GSM02.its.auckland.ac.nz)(PORT=1522))
    )
    (CONNECT_DATA=(SERVICE_NAME = DB_NAME_ha.cs_pool.oracle))
  )
)
```

# GDS - Client Connections

Option :-3

```
AUXTST=
(DESCRIPTION_LIST=
  (LOAD_BALANCE=off)
  (FAILOVER=on)
  (DESCRIPTION=
    (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
    (ADDRESS_LIST=(ADDRESS=(PROTOCOL=tcp)(HOST=GSM01-vip.db.auckland.ac.nz)
      (PORT=1522)))
    (CONNECT_DATA=(SERVICE_NAME = DB_NAME_ha.cs_pool.oracle))
  )
  (DESCRIPTION =
    (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
    (ADDRESS_LIST=(ADDRESS=(PROTOCOL=tcp)(HOST=Host-primary.db.auckland.ac.nz)
      (PORT=1521)))
    (CONNECT_DATA=(SERVICE_NAME = DB_NAME))
  )
  (DESCRIPTION=
    (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
    (ADDRESS_LIST=(ADDRESS=(PROTOCOL=tcp)(HOST=Host-drs.db.auckland.ac.nz)
      (PORT=1521)))
    (CONNECT_DATA=(SERVICE_NAME = DB_NAME))
  )
)
```

# GDS Best Help

```
GDSCTL>help  
  
add brokerconfig      add database          add gdspool  
add gsm               add invitednode      add invitedsubnet  
add region           add service           config  
config database      config gdspool        config gsm  
config region        config service        config vncr  
configure            connect              create catalog  
databases            delete catalog        disable service  
enable service       exit                  export catalog  
help                 import catalog        modify catalog  
modify database      modify gdspool        modify gsm  
modify region        modify service        quit  
relocate service     remove brokerconfig  remove database  
remove gdspool       remove gsm            remove invitednode  
remove invitedsubnet remove region         remove service  
services             set gsm               set inbound_connect_level  
set log_level        set outbound_connect_level set trace_level  
start gsm            start service         status  
status database     status gsm            status service  
stop gsm             stop service          sync brokerconfig  
sync database       synchronize brokerconfig synchronize database  
validate             validate catalog  
GDSCTL>█
```

# Test cases

Test Case	Result
Crash Primary/Standby	Global Service failovers to replica
Active Data Guard- Switchover	Data Guard role change
Active Data Guard- Lag	Route workload based on lag
Crash GDS Catalog	All services remain available
Load Balancing	Two GSM servers – faster return for Query
Relocate Service	GDSCTL> Relocate services ....
Sessions Drain	FAN post planned downtime event( Reason –user) <ul style="list-style-type: none"><li>- New work is redirected by GSM listener immediately</li><li>- Idle sessions are released</li><li>- Active sessions are released</li></ul>



# GDS Restrictions, Licence and Compatibility

## One GDS configuration manages

- 300 Database Instances
- 1000 Global Services
- 20 GDS Pools
- 10 GDS Regions
- 5 GSMs per Region
- 100 Mid-tier connections pools

## Database in GDS

- Must be licensed for Oracle Active Data Guard or Oracle GoldenGate
- Must be oracle Database EE 12.1 or greater
- Can be single or RAC
- Can be oracle Multitenant or Non- Multitenant
- Can run on oracle EXADATA OR ODA



## Lessons Learned And Improved Achieved

- Cleared concept between failover/switch over in GDS
- Missing permission after NON-CDB to CDB conversion  
/oracle/product/12.1.0.2/rdbms/admin/catgwmcat.sql
- Service using GSMCATUSER at CDB level
  
- Applications use GDS to maximize performance and availability
- Business continuity and Disaster recovery
- Central service management
- Mitigate down time during planned and unplanned outages

## Future Plans

- Database upgrade plans – 12.2 across the environment with big file type and some automation at ASM level
- Pool Limit problem
- Consolidation via multitenancy OR RAC
- Golden Gate DR set up for extract/replicate processes

Question/Answers  
??