

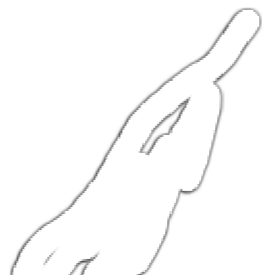
Intelligent Workload Management across Database Replicas

August 2015

Author:
Ritika Nevatia

Supervisor:
Prasanth Kothuri

CERN openlab Summer Student Report 2015



Abstract

Load management in Oracle Databases is something performed manually at CERN currently. The aim of the project is to automate this project using a tool released by Oracle called – Global Data Services. The major objective is to study and provide feedback regarding the functionality, configuration options, usability and performance of the product for automatic distribution of workloads between replicated environments for intelligent workload management across database replicas in Oracle using this tool.

Table of Contents

1	Introduction	4
2	Oracle Databases	5
2.1	Introduction	5
2.2	Server Processes	5
2.3	Memory Structure	6
3	Oracle Data Guard	7
4	Oracle Global Data Service	8
4.1	Introduction	8
4.2	Why GDS	8
4.3	Key features	9
4.4	How GDS Works	9
4.4.1	Services	9
4.4.2	Working	9
4.5	GDS Components	10
4.5.1	GDS Pool	11
4.5.2	GDS Region	11
4.5.3	Global Service Manager	11
4.5.4	GDS Catalog	12
4.5.5	Oracle Notification Servers	12
4.6	The process	13
4.7	Use Cases	13
4.7.1	Load Balancing for Replicated Databases	14
4.7.2	Service Failover in Replicas	15
4.8	Configuration	15
4.9	List of Service attributes	18
4.10	Limitations	18
5	Conclusion	18

1 Introduction

Oracle databases use various technologies like Active Data Guard for physical replication and GoldenGate for logical replication and consistency. However there existed no tool to balance the load on each of the primary or the replica(s) of the database. This is why Global Data Services (GDS) may be useful. It is important to validate the functionality of GDS and assess the suitability of the product before deploying in CERN database infrastructure.

2 Oracle Databases

2.1 Introduction

An Oracle database is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information. A database server is the key to solving the problems of information management. In general, a server reliably manages a large amount of data in a multiuser environment so that many users can concurrently access the same data. All this is accomplished while delivering high performance. A database server also prevents unauthorized access and provides efficient solutions for failure recovery.

Oracle Database is the first database designed for enterprise grid computing, the most flexible and cost effective way to manage information and applications. Enterprise grid computing creates large pools of industry-standard, modular storage and servers. With this architecture, each new system can be rapidly provisioned from the pool of components. There is no need for peak workloads, because capacity can be easily added or reallocated from the resource pools as needed.

The database has logical structures and physical structures. Because the physical and logical structures are separate, the physical storage of data can be managed without affecting the access to logical storage structures.

2.2 Server Processes

Server processes are of two kinds:

- Dedicated
 - One user process
 - Each user process accesses its own code to use the Global Area for responses
- Shared
 - Multiple user process
 - Requests from each user process are sent to the dispatcher which adds these requests to a Request Queue. When its turn comes, the user process gains access to code and gets added to the Response Queue. Once it gets the response, the dispatcher then sends the response to the respective user process.

2.3 Memory Structures

Memory can be managed in two ways:

- Automatic
- Manual

Memory structures are of two kinds:

- System Global Area
- Program Global Area

Other components of the database are:

- Control Files
- Redo Logs
- Archived Redo Logs
- Tablespace

3 Oracle Data Guard

Oracle Data Guard ensures high availability, data protection, and disaster recovery for enterprise data. Data Guard provides a comprehensive set of services that create, maintain, manage, and monitor one or more standby databases to enable production Oracle databases to survive disasters and data corruptions. Data Guard maintains these standby databases as transactional consistent copies of the production database. Then, if the production database becomes unavailable because of a planned or an unplanned outage, Data Guard can switch any standby database to the production role, minimizing the downtime associated with the outage. Data Guard can be used with traditional backup, restoration, and cluster techniques to provide a high level of data protection and data availability.

With Data Guard, administrators can optionally improve production database performance by offloading resource-intensive backup and reporting operations to standby systems.

4 Oracle Global Data Service

4.1 Introduction

Many enterprises consolidate their information technology infrastructure to improve business efficiency. Database consolidation is a critical part of this process. However, most organizations must still maintain local and remote replicas of their databases for reasons that include:

- Business continuity and disaster recovery
- High availability
- Performance optimization for local clients
- Content localization and caching
- Compliance with local laws

In any set of database replicas, some database servers may have a slow query response time because of high load or high network latency, while other servers capable of offering a faster response time may be under-utilized. Optimal query performance and resource utilization across a set of database replicas requires a workload management solution that provides dynamic load balancing of client connections and workload requests across the replicas.

Many enterprises use a home-grown solution for workload management across database replicas. These solutions cannot provide critical functionality such as run-time load balancing and reliable database service failover between replicas because they are not fully integrated with Oracle software.

4.2 Why GDS?

- Higher Availability
- More Scalability
- Better Performance
- Improved Manageability

4.3 Key Features

Work Load Management	Centralized Framework
Work Load Routing	Region based Replication lag based
Failover	Inter database service failover
Role Based Global Service	Takes care of role transition via Data Guard
Load Balancing	Run-time Connection

4.4 How GDS works?

4.4.1

Services

Oracle Database provides a powerful workload management feature called database services. Database services are named representations of one or more database instances. Database services allow you to group database workloads, ensure that client requests are routed to the optimal instance that offers a service, and provide high availability by transparently failing client connections over to surviving instances when a planned or unplanned instance outage occurs.

4.4.2 Working

Oracle Global Data Services (GDS) implements the Oracle Database service model across a set of replicated databases known as a Global Data Services configuration.

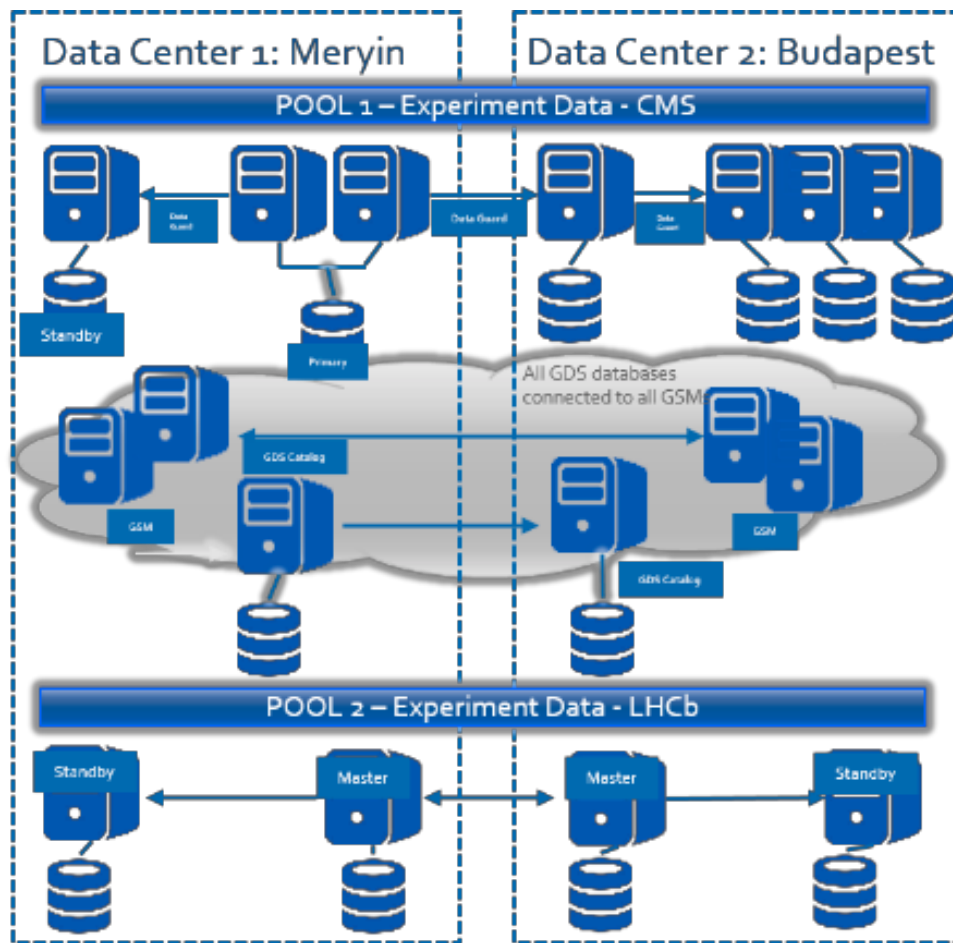
A Global Data Services configuration looks like a virtual multi-instance database to database clients. It provides client access through global services, which are functionally similar to the local database services provided by single-instance or Oracle Real Application Clusters (Oracle RAC) databases. Local and global services both provide load balancing, high availability, and resource management. The essential difference between global services and local services is that global services span the instances of multiple databases, whereas local services span the instances of a single database.

A Global Data Services configuration and its global services are created and managed using the GDSCCTL command-line interface, which is similar to the SRVCTL command-line interface used to manage an Oracle RAC database and its services.

A Global Data Services configuration can consist of any combination of multi-instance or single-instance Oracle databases hosted on heterogeneous or non-heterogeneous server platforms. Oracle Data Guard, Oracle GoldenGate, or any other database replication technology, can be used to synchronize the databases in a Global Data Services configuration.

Global Data Services is a highly effective solution for automatic workload management across a set of replicated databases, whether used with many widely distributed databases and clients or with a single database, a local replica and a few clients.

4.5 GDS Components



4.5.1 GDS Pool

- Subset of services in same administrative domain
 - Simplifies service management
 - Higher security (different admin for each pool)
 - One DB – One Pool
 - DBs in the pool need not provide same service
 - All DBs providing same service need to be in same pool

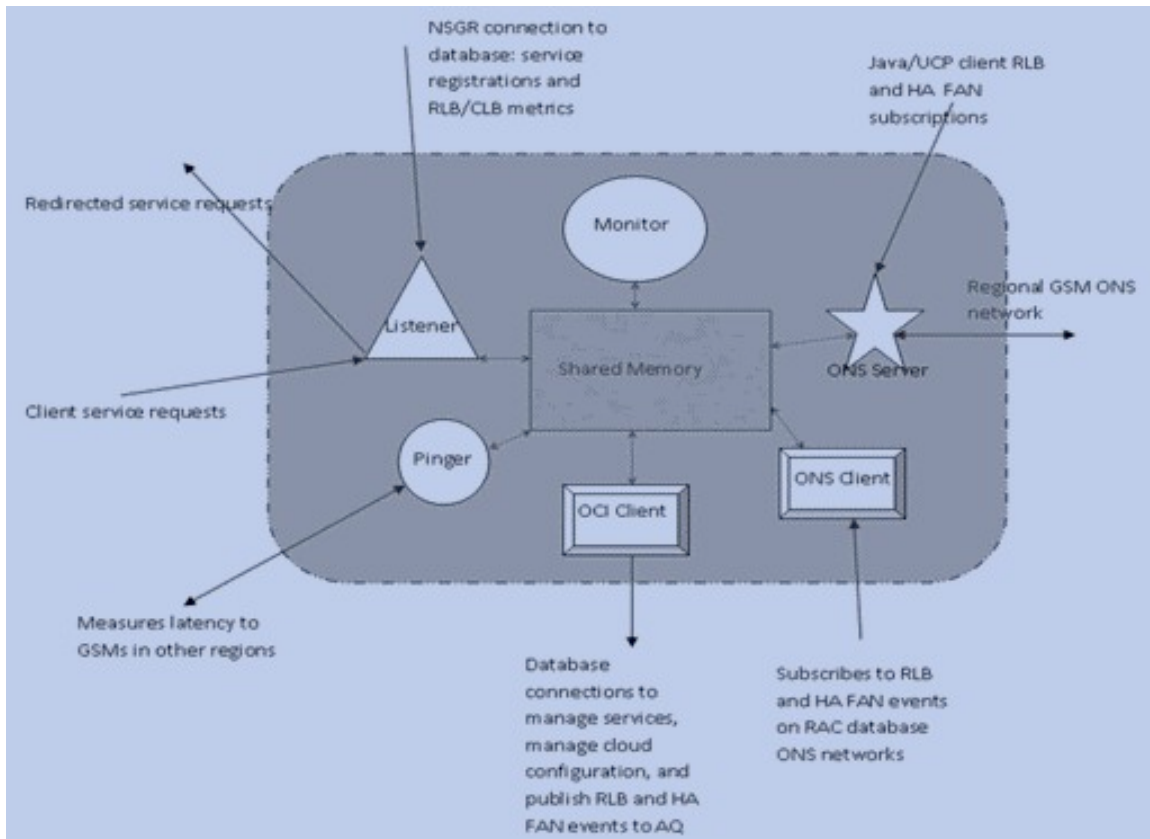
4.5.2 GDS Region

- Subset of services in same network domain
 - Low network latency
 - Region: LAN or MAN
 - One region – multiple pools
 - One pool – multiple regions
 - Buddy regions for GSM

4.5.3 Global Service Manager

GSM analogous to remote listener in Oracle RAC

- Regional listener for connecting to global services
- Connect time load balancing
- Manage global services across regions
- Collect performance metrics, measure network latency
- Create run-time load balancing advisory
- Monitor availability of DB instances/global service and notify their failure
- Each region should have atleast one GSM (Recommended: 3 GSM per region)



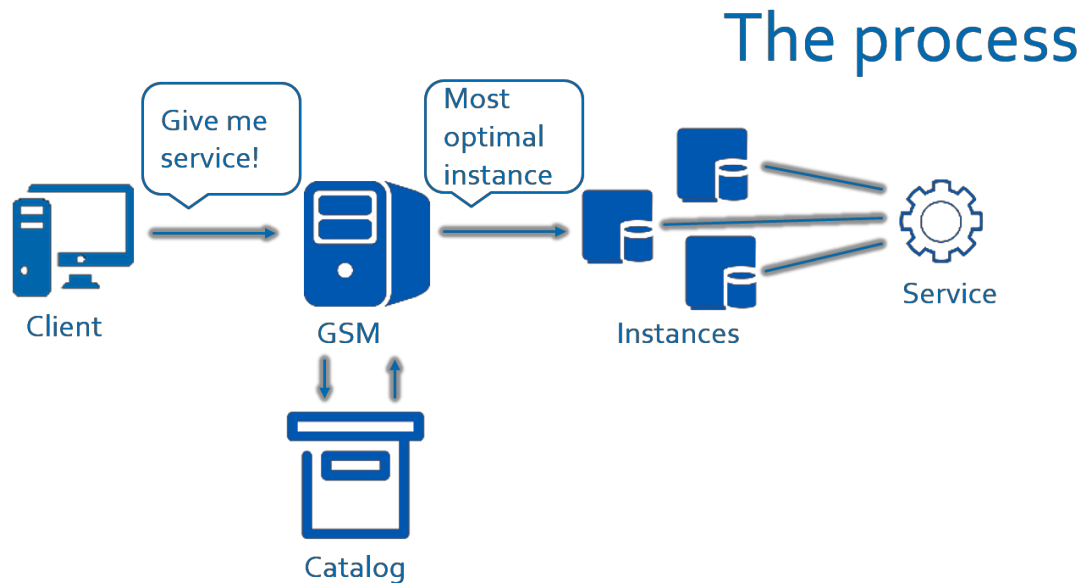
4.5.4 GDS Catalog

- Repository to store config data
- Catalog must reside in 12c DB or above
- DB can be in/out of the GDS configuration
- Maybe co-hosted with RMAN or Oracle Enterprise Manager Catalogs
- Oracle RAC/Data Guard/Clusterware should be used to enhance availability of DB where GDS Catalog resides

4.5.5 Oracle Notification Servers

- Used to receive run-time load balancing advisory and high availability events from GSM
- Co-located with each GSM
- All ONS are interconnected
- Clients subscribe to ONS and receive FAN notifications

4.6 The process



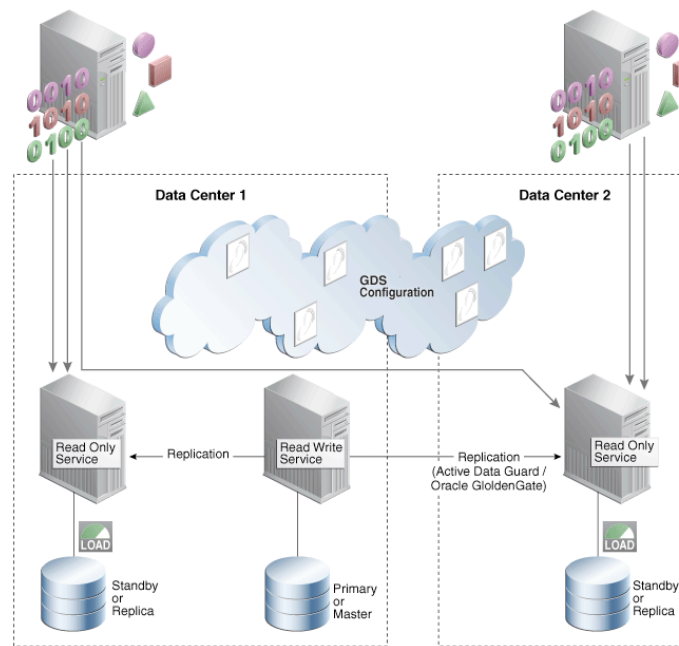
The client makes a request to the GSM asking for a service. The GSM, with the help of the catalog searches for the most optimal instance for a given service and returns it to the client.

4.7 Use Cases

There are five major use cases:

- Load Balancing for Replicated Databases
- Service Failover for Replicated Databases
- Region Affinity in Oracle GoldenGate Multi-Master
- Load Balancing in Oracle GoldenGate Multi-Master
- Balancing Oracle Active Data Guard and Oracle GoldenGate Reader Farms

4.7.1 Load Balancing for Replicated Database

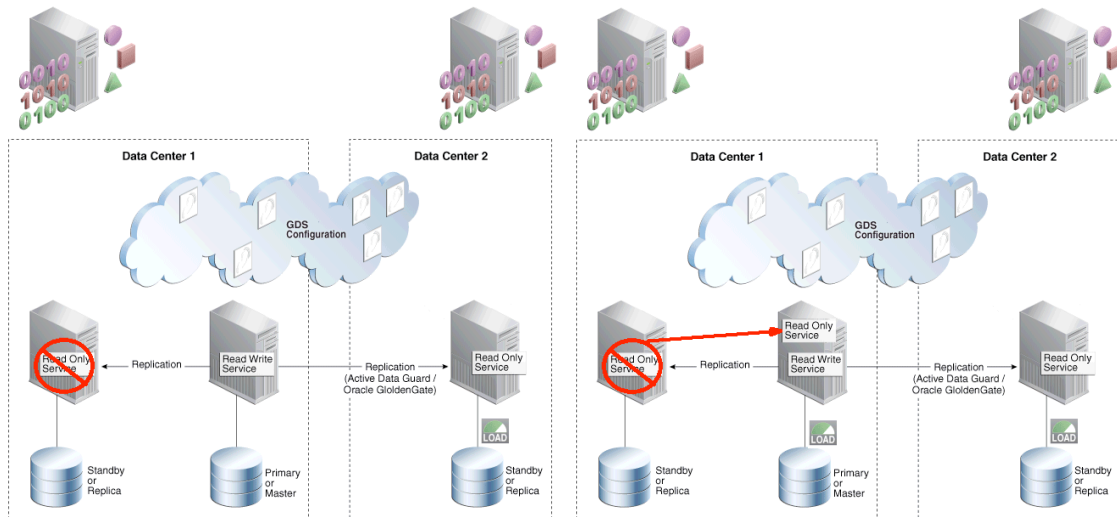


GDS considers the following points while balancing the load:

- Load metrics
- Region affinity
- Network latency
- Load balancing goals

And then makes an appropriate decision to balance loads on replicated database. This was successfully tested on three physical machines with Oracle 12c.

4.7.2 Service Failovers in replicas



Let's say the replica at Data Centre 1 fails. All queries being redirected to the replica will not receive a reply due to the shutdown. In this case, GDS distributes the traffic effectively on the primary database and redirects the queries there.

This test case was also successfully tested by shutting down both primary and standby database on different occasions.

4.8 Configuration Details

Steps to configure GDS are as follows (with commands):

1. Checking all versions:
 - a. `rpm -qa | grep make-3.81`
 - b. `rpm -qa | grep binutils-2.17.50.0.6`
 - c. `rpm -qa | grep gcc`
 - d. `rpm -qa | grep libaio`
 - e. `rpm -qa | grep libstdc++`
 - f. `rpm -qa | grep sysstat`
 - g. `rpm -qa | grep compat-libstdc++`
 - h. `rpm -qa | grep libgcc`
 - i. `rpm -qa | grep libstdc++-devel`
 - j. `rpm -qa | grep glibc-devel`
 - k. `rpm -qa | grep gcc-c++`
 - l. `rpm -qa | grep glibc`
 - m. `rpm -qa | grep ksh`
2. Make a directory to install GDS, download the package, unzip it

3. Install
 - a. set DISPLAY=137.138.35.9:0.0
 - b. ./runInstaller
 - c. Follow steps in the display box
4. Create a GDS user
 - a. Log into xxserverxx
 - b. sudo su – oracle
 - c. set_env
 - d. sqlsys
 - e. sys@db:SQL> create user gds identified by "xxx" default tablespace tools temporary tablespace temp;
 - f. sys@db:SQL> alter user gds quota 10g on tools;
 - g. sys@db:SQL> grant gsmadmin_role to gds;
5. Creating GDS Catalog
 - a. Go to oracle/product/gsm/bin
 - b. ./gdsctl
 - c. GDSCTL>create catalog -database db -user gds/xxx
6. Give access to gsmcatuser
 - a. Log into DBA account
 - b. sys@db:SQL> alter user gsmcatuser account unlock;
 - c. sys@db:SQL> ALTER USER gsmcatuser IDENTIFIED BY xxx;
7. Create GSM
 - a. GDSCTL>add gsm -gsm mgsm -listener 10121 -catalog db
 - b. GDSCTL>modify gsm -gsm gsm1 -listener 10140

```
rlwrap ./gdsctl
```

Tip: Run gdsctl with rlwrap to save a log of commands
8. Connect to gds
 - a. connect gds@db-v:10121:xxxx1
 - b. connect gds@db
9. Set time zone for database
10. Create Pool
 - a. GDSCTL> add gdspool -dbpool hr -users gds
11. Create Region
 - a. GDSCTL> add region –region west,east
12. Create Databases
 - a. add database -connect db -region REGIONORA -gdspool DBPOOLORA

- b. add database -connect db -region REGIONORA -gdspool DBPOOLORA

13. Create Services

- a. add service -gdspool DBPOOLORA -service erts_rw -clbgoal LONG - commit_outcome TRUE -preferred db -available db5 - commit_outcome TRUE -notification TRUE -policy automatic - failoverdelay 10 -failoverretry 6 - session_state DYNAMIC -failovertime TRANSACTION
- b. add service -gdspool DBPOOLORA -service erts_r -clbgoal LONG - commit_outcome TRUE -preferred db5 -available db4 - commit_outcome TRUE -notification TRUE -policy automatic - failoverdelay 10 -failoverretry 6 - failovertime TRANSACTION - session_state DYNAMIC
- c. add service -gdspool DBPOOLORA -service erts_r2 -clbgoal LONG - commit_outcome TRUE -preferred db, db5 - commit_outcome TRUE -notification TRUE -policy automatic - failoverdelay 10 - failoverretry 6 -failovertime TRANSACTION - session_state DYNAMIC

14. Start Service

- a. start service -service erts_rw
- b. start service -service erts_r
- c. start service -service erts_r2

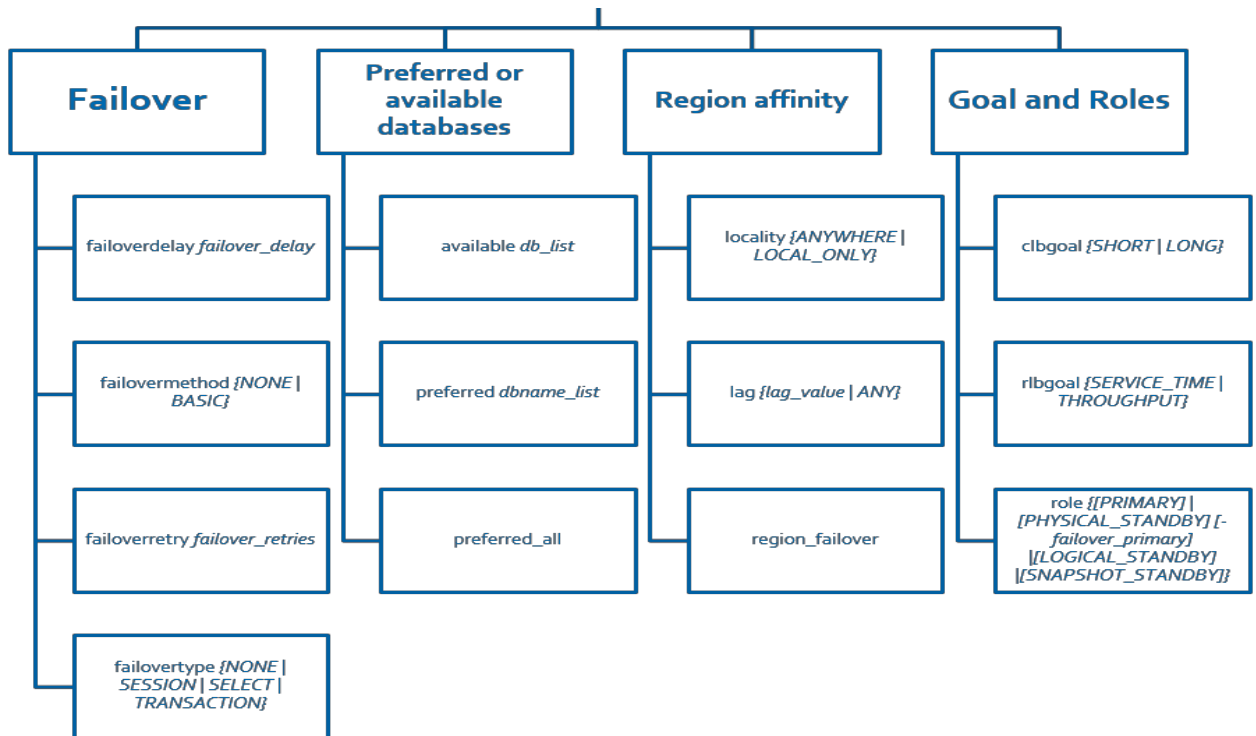
15. Tip (Instance needs modification sometimes):

- a. GDSCTL>modify service -service erts_rw -database db - modify_instances -preferred xxx -gdspool erts

16. Connect your clients to the services using the service names, and get most optimal instances returned for every query.

4.9 List of service attributes

The figure below lists an overview of available attributes to define a service:



4.10 Limitations

The reason why Global Data Services, despite being so useful, is not yet actively used are:

1. Requires all databases upgraded to Oracle 12C
2. Limited documentation, hence limited usage which goes into a cycle

5 Conclusion

Global Data Services, as a product works according to the description given in Oracle documentation and is useful for intelligent workload management.