

Лекция по теме:

# Построение Standby Database на основе технологии Oracle Active Data Guard

Байдукалов Е.В,  
Санкт-Петербург, 2016

# Технологии Oracle для систем повышенной надежности

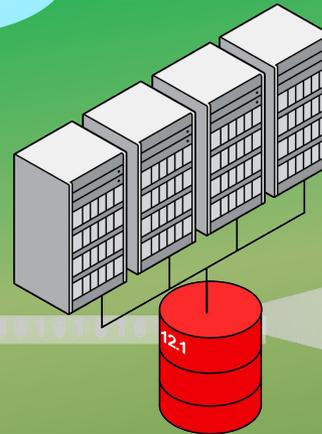
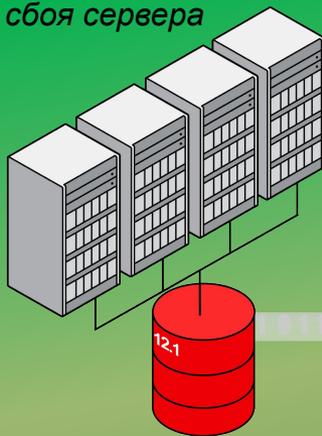


## RAC

- масштабируемость
- Защита от сбоя сервера

## Flashback

- Защита от ошибок человека



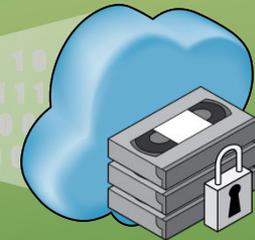
## Активная реплика

### Active Data Guard

- Защита от катастроф
- Нагрузка на чтение

### GoldenGate

- Active-active
- Heterogeneous



### RMAN, Oracle Secure Backup

- Backup to tape / cloud

ADG – это технология, обеспечивающая процесс односторонней репликации транзакций из основной БД на резервную БД посредством фоновых процессов сервера СУБД.

# Когда нужны системы повышенной готовности и системы повышенной надежности ?

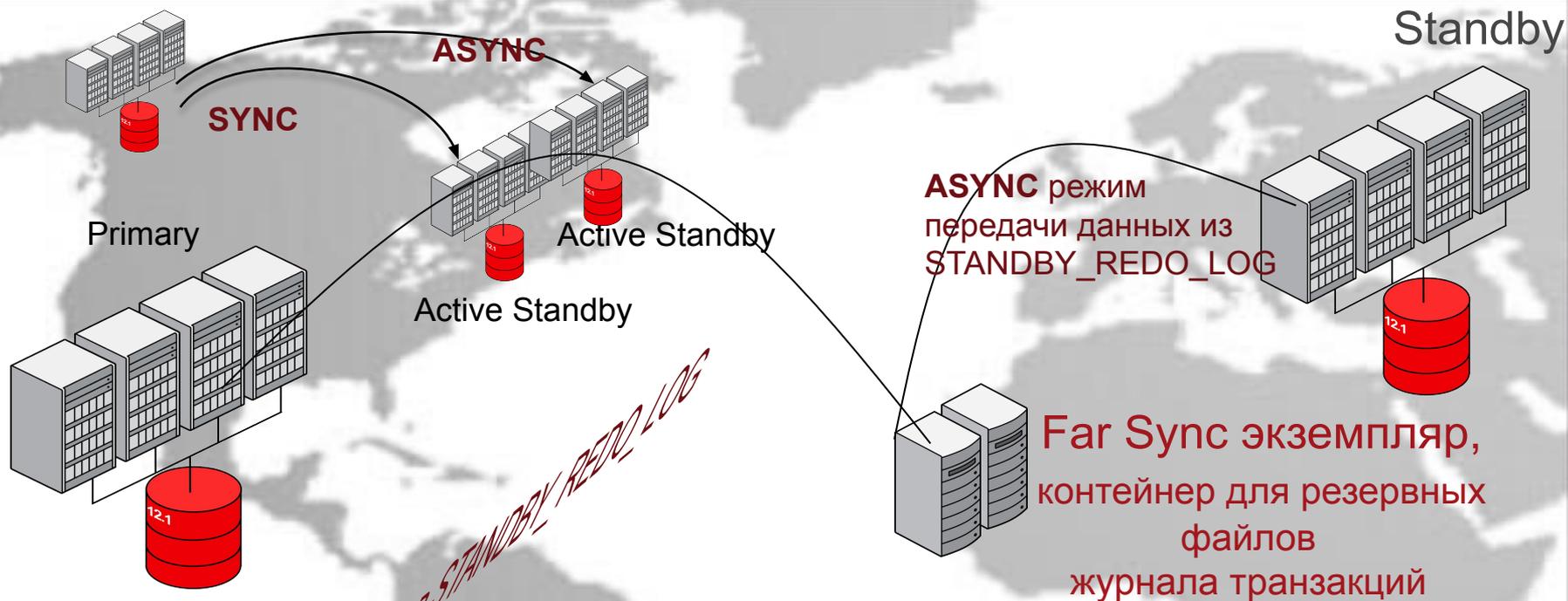
- ❖ Экономическая угроза существования бизнеса из-за потери данных.
- ❖ Недополученная выгода и конкурентное преимущество от потери данных.
- ❖ Недополученная прибыль, потеря доверия со стороны клиентов при простое системы.
- ❖ Простой рабочей силы и делопроизводства при не надежной работе приложений системы.

# Классификация приложений в системах повышенной надежности

Виды приложений	Времени простоя	RTO	RPO
Критичные	Недопустим	Секунды	Недопустимо
Бизнес-критичные	Минимальный	Минуты	Недопустимо
Уровня предприятия	Бизнес-час	Часы	Минимальная
Не критичные	Час	Дни	Не имеет значения

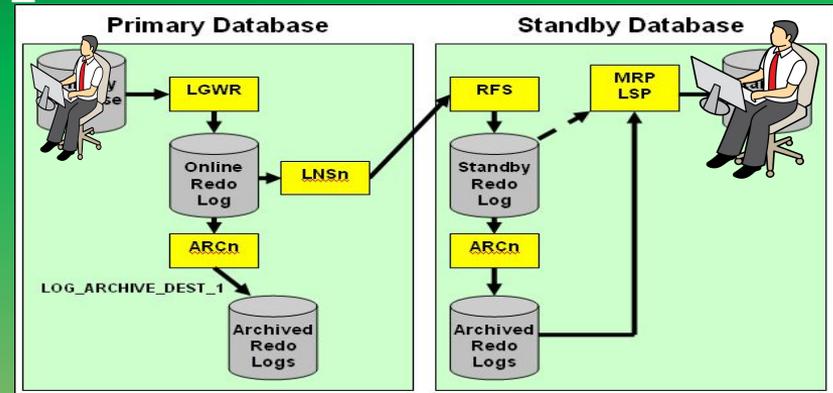
- ❖ RTO (recovery time objective) - показатель определяет допустимое **время простоя** в случае наступления катастрофического события.
- ❖ RPO (recovery point objective) - показатель определяет, какой **объем данных** вы можете себе позволить потерять в случае наступления катастрофического события.

# Oracle Active Data Guard 12c



Синхронный транспорт (SYNC) иногда упоминается как метод "без потери данных", потому что процесс LGWR не фиксирует транзакцию пока не подтвердится, что аналогичная запись есть на Standby DB. Если подтверждения по каким либо причинам не приходит, то это сказывается на работоспособности основного сервера, поэтому обычно используют асинхронный транспорт (позволяет не дожидаться ответа со Standby).

# Database в конфигурации DataGuard



## Фоновые процессы:

- ❖ LGWR Log Writer - копирует содержимое буфера журнала из памяти на диск.
- ❖ Новый процесс LNS ( LogWriter Network Server – сетевой сервер записи в журнал) избавляет процесс LogWriter от накладных расходов, связанных с передачей журнальных данных в удаленную резервную базу данных.
- ❖ При совершении транзакции создается redo log в области SGA, LGWR читает эту запись из redo log buffer и записывает ее в online redo log file - и ждет подтверждения от LNS. LNS читает ту же запись из буфера и передает ее в резервную базу данных с помощью Oracle Net Services, процесс Remote File Server (RFS) получает запись и записывает ее в Standby Redo Logs (SRL).

- ❖ Если **LNS** не успевает забирать запись из буфера , то он автоматически переходит к чтению и отправке записи из файла журнала транзакций вместо redo log buffer. После того, как LNS (LogWriter Network Server) сможет догнать LGWR, он опять возвращается к чтению прямо из буфера в SGA.
- ❖ Соотношение redo log buffer отслеживается с помощью представления X\$LOGBUF\_READHIST : низкий коэффициент указывает, что LNS читает из журнального файла вместо буфера ( на заметку, если это происходит регулярно, попробуйте увеличить размер буфера журнала).
- ❖ По мере того как процесс RFS записывает журнальные данные в SRL, MRP (Managed Recovery Process) читает данные из SRL и применяет изменения непосредственно к Standby DB.
- ❖ Процесс MRP может также переключиться на чтение из архивного журнала резервной базы данных, если SRL архивирован прежде, чем MRP может закончить чтение SRL (ситуация, которая может произойти - когда первичная база данных имеет очень высокую скорость генерации журнальных данных).

- ❖ Если вследствие **отказа** сети или отказов резервных серверов разрывается соединение первичной и резервных баз данных, то первичная база данных продолжает обрабатывать транзакции и накапливать журнальные данные, которые не могут быть отправлены в резервные базы данных до тех пор, пока не будет установлено новое сетевое подключение. В таком случае будет выполняться следующий сценарий:
- ❖ 1) Процесс ARCH в Primary постоянно будет **опрашивать** Standby, чтобы определить ее состояние.
- ❖ 2) Когда связь восстанавливается, то ARCH **опрашивает** standby control file (с помощью процесса RFS), чтобы определить последнюю версию журнального файла полученных от primary.
- ❖ 3) Data Guard узнает какие журнальные файлы нужны для синхронизации и сразу начинает их передавать с помощью **дополнительных** ARCH процессов.
- ❖ 4) После синхронизации LNS начинает работать в обычном режиме.

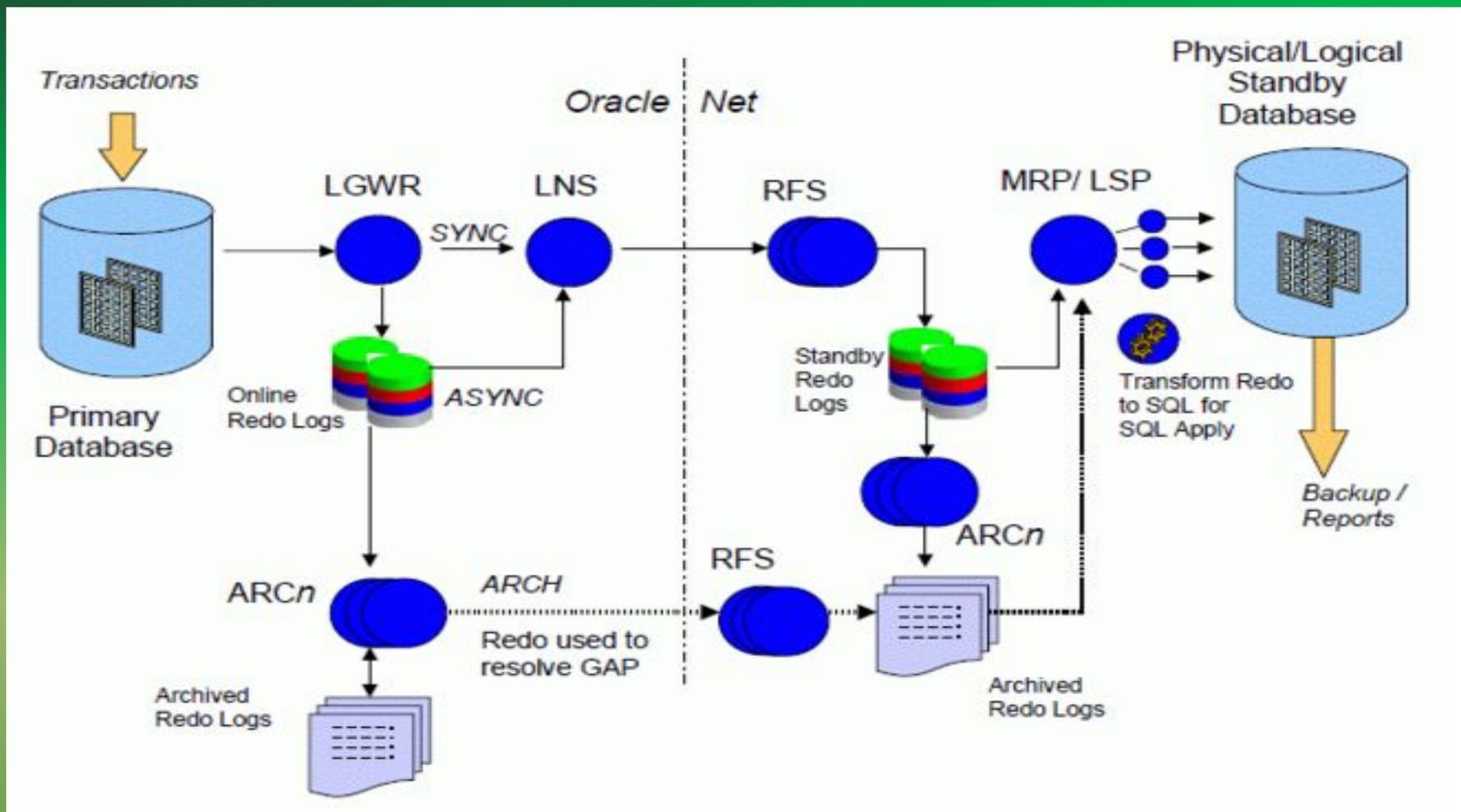


Иллюстрация работы фоновых процессов в синхронном и асинхронном режиме передачи информации на резервную базу данных и при разрыве связи.

# Методология конфигурирования DG

- I. До клонирования.* Цель – все сервера должны работать, как одна логическая машина.
- II. Клонирование.* Цель – создание такой же структуры и содержания, как на Primary DB.
- III. После клонирования.* Цель – запуск процесса репликации. Настройка DG Broker.  
Мониторинг.

# 1. До клонирования

- 1.1 Перевод базы данных в режим логирования;
- 1.2 Запуск резервного экземпляра.
- 1.3 Настройка сетевых файлов;
- 1.4 Настройка параметров `init.ora`;
- 1.5 Создание файла паролей;
- 1.6 Добавление `standby_redo_log` файлов;

# 1.1 Перевод базы данных в режим логирования

```
SQL> archive log list;
```

```
SQL> SELECT flashback_on, log_mode FROM v$database;
```

```
SQL> shutdown immediate;
```

```
SQL> startup mount;
```

```
SQL> alter database archivelog;
```

```
SQL> alter database open;
```

```
SQL> archive log list;
```

```
Database log mode      Archive Mode
```

```
Automatic archival    Enabled
```

```
SQL> alter database force logging;
```

```
SQL> select force_logging from v$database;
```

```
FORCE_LOGGING
```

```
-----  
YES
```

## 1.2 Настройка параметров init.ora

```
SQL> show parameter db_unique_name;
```

```
SQL> alter system set
```

```
    log_archive_config='dg_config=(spbstu,spbstu_stb)' scope=both;
```

```
// LOG_ARCHIVE_CONFIG - определяем имена экземпляров,  
    между которыми будет происходить обмен журналами. //
```

```
SQL> alter system set log_archive_dest_2='SERVICE=spbstu_stb  
    LGWR ASYNC
```

```
    VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
```

```
    db_unique_name=spbstu_stb' scope=both;
```

```
SQL> alter system set log_archive_dest_state_2=ENABLE  
    scope=both;
```

```
SQL> show parameter log_archive_dest_state_2
```

```
//log_archive_dest_2 - куда будут передаваться архивлоги -  
    файловой системе или сервису. Параметр ASYNC указывает,  
    что данные, сгенерированные транзакцией, не обязательно  
    должны быть получены на standby до завершения транзакции.
```

```
SQL> alter system set FAL_SERVER=spbstu_stb scope=both;
```

```
SQL> alter system set FAL_CLIENT=spbstu scope=both;
```

// fal\_client='spbstu' – этот параметр определяет, что когда экземпляр перейдет в режим standby, он будет являться клиентом для приема архивных журналов (fetch archive log).

fal\_server='spbstu\_stb' – определяет FAL (fetch archive log) сервер, с которого будет осуществляться передача архивных журналов. Параметры fal\_client и fal\_server работают только когда база запущена в standby режиме. //

```
SQL> alter system set standby_file_management='AUTO'  
scope=both;
```

// standby\_file\_management='AUTO' – задаем режим автоматического управления файлами в standby режиме. При таком значении параметра все создаваемые или удаляемые файлы основной базы будут автоматически создаваться или удаляться и на standby базе. //

```
SQL> SHOW PARAMETER ARCH;
```

NAME	TYPE	VALUE
archive_lag_target	integer	0
log_archive_config	string	DG_CONFIG=(spbstu, spbstu_stb)
log_archive_dest	string	
log_archive_dest_1	string	LOCATION=+ARCH VALID_FOR=(all_logfiles,all_roles) db_unique_name=spbstu
log_archive_dest_10	string	
log_archive_dest_11	string	
log_archive_dest_12	string	
log_archive_dest_13	string	
log_archive_dest_14	string	

NAME	TYPE	VALUE
log_archive_dest_15	string	
log_archive_dest_16	string	
log_archive_dest_17	string	
log_archive_dest_18	string	
log_archive_dest_19	string	
log_archive_dest_2	string	SERVICE=spbstu_stb LGWR ASYNC NOAFFIRM NET_TIMEOUT=30 VALID_FOR=( ONLINE_LOGFILES, PRIMARY_ROLE) DB_UNIQUE_NAME=spbstu_stb

## 1.3 Создание pfile для Standby DB

```
SQL> create pfile from spfile;
```

Для standby

```
SQL> create pfile= '/u01/app/oracle/pfilespbstu_stb.ora' from spfile;
```

Изменяем и передаем на резервный хост

```
*.db_unique_name='spbstu_stb'
```

```
*.fal_server='spbstu_stb'
```

```
*.fal_client='spbstu'
```

```
*.log_archive_config='DG_CONFIG=(spbstu, spbstu_stb)'
```

```
*.log_archive_dest_1='LOCATION=+ARCH
```

```
VALID_FOR=(all_logfiles,all_roles) db_unique_name=spbstu_spb'
```

```
*.log_archive_dest_2='SERVICE=spbstu LGWR ASYNC
```

```
VALID_FOR=( ONLINE_LOGFILES, PRIMARY_ROLE)
```

```
DB_UNIQUE_NAME=spbstu'
```

```
$ scp '/u01/app/oracle/pfilespbstu_stb.ora'
```

```
oracle@ol68: '/u01/app/oracle/'
```

# 1.4 Добавление standby\_redo\_log файлов

```
//ALTER SYSTEM SET STANDBY_FILE_MANAGEMENT=MANUAL  
SCOPE=BOTH;
```

```
ASMCMD> mkdir STANDBYLOG
```

```
ALTER DATABASE ADD STANDBY LOGFILE thread 1 GROUP 4  
'+DATA/SPBSTU/STANDBYLOG/stby_4.log' size 52428800 reuse ;
```

```
ALTER DATABASE ADD STANDBY LOGFILE thread 1 GROUP 5  
'+DATA/SPBSTU/STANDBYLOG/stby_5.log' size 52428800 reuse ;
```

```
ALTER DATABASE ADD STANDBY LOGFILE thread 1 GROUP 6  
'+DATA/SPBSTU/STANDBYLOG/stby_6.log' size 52428800 reuse ;
```

```
ALTER DATABASE ADD STANDBY LOGFILE thread 1 GROUP 7  
'+DATA/SPBSTU/STANDBYLOG/stby_7.log' size 52428800 reuse ;
```

```
SQL> select group#,status from v$standby_log;
```

```
SQL> select TYPE, MEMBER from v$logfile where TYPE='STANDBY';
```

Удаление (если потребовалось ) standby\_redo\_log файлов

```
SQL> alter database drop logfile group 7;
```

# 1.5 Создание файла паролей

primary

```
SQL> show parameter remote_login_passwordfile;
```

```
SQL> select USERNAME from v$pwfile_users;
```

Для получения возможности подключения к базе через файл паролей достаточно:

- ❖ Создать файл паролей ORAPWD FILE=filename  
PASSWORD=password ENTRIES=max\_users
- ❖ Установить параметр инициализации  
REMOTE\_LOGIN\_PASSWORDFILE в значение EXCLUSIVE. Это значение по умолчанию.
- ❖ И иметь для пользователя привилегии SYSDBA.

```
$> orapwd file=orapwspbstu password=oracle entries=7
```

Передача файла на Standby DB

```
$ scp $ORACLE_HOME/dbs/orapwspbstu  
oracle@192.168.10.103:$ORACLE_HOME/dbs/orapwspbstu_stb
```

standby

```
$ chmod 4640 $ORACLE_HOME/dbs/orapwspbstu_stb
```

# 1.7 Запуск Standby DB

```
$ sqlplus / as sysdba
```

```
SQL> startup nomount pfile='/.....ora'
```

```
SQL> show parameter db_unique_name;
```

```
SQL> select status from v$instance;
```

```
STATUS
```

```
-----  
STARTED
```

# 1.3 Oracle Net

## Primary

```
LISTENER2 =  
  (DESCRIPTION_LIST =  
    (DESCRIPTION =  
      (ADDRESS = (PROTOCOL = TCP)  
HOST = 192.168.10.102)(PORT = 1522))  
    )  
  )  
SID_LIST_LISTENER2 =  
  (SID_LIST =  
    (SID_DESC =  
      (GLOBAL_DBNAME = spbstu)  
      (SID_NAME = spbstu)  
      (ORACLE_HOME =  
/u01/app/oracle/product/12.1.0/db_1)  
    )  
  )  
)
```

## Standby

```
LISTENER2 =  
  (DESCRIPTION_LIST =  
    (DESCRIPTION =  
      (ADDRESS = (PROTOCOL = TCP)  
HOST = 192.168.10.103)(PORT = 1522))  
    )  
  )  
SID_LIST_LISTENER2 =  
  (SID_LIST =  
    (SID_DESC =  
      (GLOBAL_DBNAME = spbstu)  
      (SID_NAME = spbstu_stb)  
      (ORACLE_HOME =  
/u01/app/oracle/product/12.1.0/db_1)  
    )  
  )  
)
```

# Primary

## Standby

# tnsnames.ora Network Configuration File: /u01/app/oracle/product/12.1.0/db\_1/network/admin/tnsnames.ora

# Generated by Oracle configuration tools.

SPBSTU =

```
(DESCRIPTION =  
  (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.10.102)(PORT = 1522))  
  (CONNECT_DATA =  
    (SERVER = DEDICATED)  
    (SERVICE_NAME = spbstu)  
  )  
)
```

SPBSTU\_STB =

```
(DESCRIPTION =  
  (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.10.103)(PORT = 1522))  
  (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.10.103)(PORT = 1523))  
  (CONNECT_DATA =  
    (SERVER = DEDICATED)  
    (SERVICE_NAME = spbstu)  
    #(SID_NAME = spbstu)  
  (UR = A)  
  )  
)
```

**GLOBAL\_DBNAME** . Глобальное имя базы данных уникальным образом идентифицирует базу данных Oracle и имеет формат имя\_базы\_данных.домен\_базы\_данных — например, sales.us.acme.com. Поскольку никакие две базы данных в одном домене не могут иметь одинаковые имена, каждое глобальное имя базы данных уникально.

**SERVICE\_NAME** . В системе каждая база данных идентифицируется по ее имени службы, для указания которого служит параметр инициализации SERVICE\_NAMES. По умолчанию значение имени службы устанавливается соответствующим глобальному имени базы данных. Обратите внимание, что база данных может адресоваться более чем по одному имени службы. Это может быть реализовано, если нужно, чтобы различные наборы клиентов по-разному адресовались к базе данных для удовлетворения их конкретных потребностей. Например, для одной и той же базы данных можно создать два имени служб наподобие следующих: sales.us.acme.com finance.us.acme.com

**SID\_NAME** .Имя экземпляра базы данных указывается в файле инициализации (init.ora) в виде параметра INSTANCE\_NAME. Когда речь идет о системном идентификаторе (SID) Oracle, подразумевается просто экземпляр Oracle.

# Утилиты для проверки сети

```
$ ifconfig  
$ ping  
$ ping  
$ tnsping  
$ tnsping
```

```
[oracle@ol67 admin]$ tnsping spbstu_stb  
  
TNS Ping Utility for Linux: Version 12.1.0.1.0 - Production on 23-FEB-2016 12:05  
:20  
  
Copyright (c) 1997, 2013, Oracle. All rights reserved.  
  
Used parameter files:  
  
Used TNSNAMES adapter to resolve the alias  
Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = ol68)(PORT = 1522)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = spbstu_stb)))  
OK (210 msec)
```

## 2. Клонирование

- 2.1 Проверка существования директорий указанных в primary init file на Standby DB;
- 2.2 Установка параметра local listener;
- 2.3 Подключение к RMAN;
- 2.4 Проверка создания standby\_log\_file на Standby DB;
- 2.5 Перевод standby db в режим mount;

## 2.1 Проверка существования директорий указанных в primary init file на Standby

Primary

```
SQL> show parameter audit
```

```
audit_file_dest          /u01/oracle/admin/spbstu/adump
```

Standby

```
$ mkdir -p /u01/oracle/admin/spbstu/adump
```

## 2.2 Установка параметра local listener

### Primary

```
SQL> alter system set local_listener='(DESCRIPTION =(ADDRESS =  
  (PROTOCOL = TCP)(HOST = ..... )(PORT = 1522)))' scope=both;
```

### Standby

```
SQL> alter system set local_listener='(DESCRIPTION =(ADDRESS =  
  (PROTOCOL = TCP)(HOST = ..... )(PORT = 1522)))' scope=both;
```

- ❖ LOCAL\_LISTENER указывает имя сети, указывающее на адрес или список адресов Oracle Net местных слушателей (то есть, слушателей, которые работают на той же машине). Адрес или список адресов указан в TNSNAMES.ORA файле.

## 2.3 Подключение к RMAN

Standby

```
[oracle@ol67 ~]$ rman target sys/oracle@spbstu auxiliary  
sys/oracle@spbstu_stb
```

Recovery Manager: Release 12.1.0.1.0 - Production on Fri Mar 18  
04:53:21 2016

Copyright (c) 1982, 2013, Oracle and/or its affiliates. All rights  
reserved.

connected to target database: **SPBSTU** (DBID=2443390911)

connected to auxiliary database: **SPBSTU** (not mounted)

## Скрипт для RMAN

```
run{  
allocate channel chan1 type disk;  
allocate channel chan2 type disk;  
allocate auxiliary channel aux1 device type disk;  
allocate auxiliary channel aux2 device type disk;  
duplicate target database for standby from active database  
dorecover nofilenamecheck; }
```

- ❖ Параметр `nofilenamecheck` нужен, чтобы `rman` не ругался на повторяющиеся имена файлов (если мы используем одинаковую структуру каталогов на `primary` и `standby`).

## 2.4 Проверка создания standby\_log\_file на Standby

```
SQL> select * from v$standby_log;
```

## 2.5 Проверяем, что standby db в режим mount;

Standby

```
SQL> select name, db_unique_name, database_role,  
           protection_mode from v$database;
```

```
SQL> select name, controlfile_type, open_mode, log_mode  
           from v$database;
```

NAME	CONTROL	OPEN_MODE	LOG_MODE
SPBSTU	STANDBY	MOUNTED	ARCHIVELOG

Если не в mount :

```
SHUTDOWN IMMEDIATE;
```

```
STARTUP NOMOUNT;
```

```
ALTER DATABASE MOUNT STANDBY DATABASE;
```

## 3. После клонирования

- 3.1 Узнать max redo log на primary и max standby redo log на standby;
- 3.2 Запуск процесса MRPO;
- 3.3 Мониторинг Redo Apply;
- 3.4 Перевод базы данных в режим read only
- 3.5 Переключение ролей баз данных.

## 3.1 Узнать max redo log на primary и max standby redo log на standby

Выполняем на primary и standby, запрос который нам покажет количество архивлогов:

```
SQL> select max(sequence#) from v$archived_log;
```

Потом на primary, выполняем несколько раз команду:

```
SQL> ALTER SYSTEM SWITCH LOGFILE
```

Разница - Gap ( разрыв )

## 3.2 Запуск и останов процесса MRPO

- ❖ Переводим нашу standby базу в режим Real-time apply redo:  
SQL> alter database recover managed standby database using current logfile disconnect;

Или

SQL> alter database recover managed standby database using current logfile disconnect from session;

- ❖ Если мы не хотим использовать режим Real-time apply redo, а хотим дожидаться когда будет закончено формирование очередного архивного журнала на основном сервере и он будет передан на standby для применения сохраненных в нем транзакций, то нам необходимо переводить нашу standby базу в режим redo apply командой:

SQL> alter database recover managed standby database disconnect;

- ❖ Если что-то пошло не так, то для решения проблемы в первую очередь необходимо остановить «накатку» логов:

SQL> alter database recover managed standby database cancel;

## 3.3 Мониторинг Redo Apply

Проверяем

```
SQL> select recovery_mode from v$archive_dest_status;
```

```
SQL>select process, status from v$managed_standby;
```

```
SQL> select max(sequence#) from v$archived_log;
```

# Системные представления DG

View	На стороне Primary	На стороне Standby
V\$ARCHIVED_LOG	Какие логи отправляются	Полученные логи
V\$ARCHIVED_DEST_STATUS	Состояние arch процессов, куда они отправляют файлы, последний отправленный файл.	В каком режиме применяются логи.
V\$DATAGUARD_STATUS	-	Информация из alert log файла.
V\$MANAGED_STANDBY	-	Запущен ли процесс MRP.

```
9. 192.168.10.103 (oracle) 13. 192.168.10.102 (oracle) × +
SQL> select max(sequence#) from v$log_history;
MAX(SEQUENCE#)
-----
          51

SQL> select max(sequence#) from v$sarchived_log;
MAX(SEQUENCE#)
-----
          82

SQL> select controlfile_type, open_mode from v$database;
CONTROL  OPEN_MODE
-----
STANDBY MOUNTED

SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE using current logfile DISCONNECT FROM SESSION;
Database altered.

SQL> select process, status from v$managed_standby;
PROCESS    STATUS
-----
ARCH       CLOSING
ARCH       CLOSING
ARCH       CONNECTED
ARCH       CLOSING
RFS        IDLE
RFS        IDLE
RFS        IDLE
RFS        IDLE
MRP0       APPLYING_LOG

9 rows selected.

SQL> select recovery_mode from v$archive_dest_status;
RECOVERY_MODE
-----
MANAGED REAL TIME APPLY
IDLE
IDLE
```

## 3.4 Перевод standby в режим read only

```
SQL> shutdown immediate;
```

```
SQL> startup mount;
```

```
SQL> alter database open read only;
```

```
SQL> select name, open_mode, log_mode, database_role from  
v$database;
```

NAME	OPEN_MODE	LOG_MODE	DATABASE_ROLE
SPBSTU	READ ONLY WITH APPLY	ARCHIVELOG	PHYSICAL STANDBY

# Конфигутирование Data Guard Broker

primary и standby:

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_2=" ";
```

```
SQL> ALTER SYSTEM set  
  dg_broker_config_file1='+DATA/db_brocker1.dat' SCOPE=both;
```

```
SQL> ALTER SYSTEM set  
  dg_broker_config_file2='+ARCH/db_brocker2.dat' SCOPE=both;
```

```
SQL> ALTER SYSTEM SET dg_broker_start=TRUE SCOPE=both;
```

# Primary

```
$ dgmgrl
```

```
DGMGRL> CONNECT sys@spbstu
```

```
DGMGRL> CREATE CONFIGURATION 'spbstu' AS PRIMARY  
DATABASE IS 'spbstu' CONNECT IDENTIFIER IS primary;
```

# Standby

```
DGMGRL> ADD DATABASE 'spbstu_stb' AS CONNECT IDENTIFIER  
IS standby maintained as physical;
```

```
DGMGRL> ENABLE CONFIGURATION;
```

```
DGMGRL> show configuration
```

```
DGMGRL> show database spbstu
```

```
DGMGRL> show database spbstu_stb
```

Остановить bkoker

```
SQL> ALTER SYSTEM SET dg_broker_start=FALSE SCOPE=both;
```

Выключить конфигурацию:

```
DGMGRL> disable configuration;
```

Удалить конфигурацию:

```
DGMGRL> REMOVE CONFIGURATION;
```

Получить подробную информации по базе:

```
DGMGRL> show database verbose spbstu_stb
```

Получить подробную информации по экземпляру:

```
DGMGRL> show instance verbose spbstu on database spbstu_stb
```

## 3.5 Переключение ролей баз данных.

```
DGMGRL> switchover to 'SPBSTU_STB';
```

```
DGMGRL> switchover to 'SPBSTU';
```

# Режимы защиты

- В Data Guard предлагаются три режима защиты данных для балансировки стоимости, готовности, производительности и защищенности данных. Эти режимы определяют правила, управляющие поведением конфигурации Data Guard, и могут быть легко установлены, используя любой из доступных интерфейсов управления, например, если для первичной базы данных использовать следующий простой оператор SQL:

```
SQL> ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE  
      {PROTECTION |AVAILABILITY |PERFORMANCE};
```

- Чтобы определить подходящий режим защищенности данных, нужно взвесить свои бизнес-требования к защите данных и соотнести их с допустимым для пользователей временем ответа системы.

Режим	Риск потери данных	Метод транспортировки журнальных данных
Maximum Protection (Максимальная защищенность)	Нулевая потеря данных	SYNC
Maximum Availability (Максимальная готовность)	Нулевая потеря данных – в предположении, что до отказа не было никаких нарушений в синхронной передаче данных в то время, когда первичная база данных фиксировала (commit) транзакции	SYNC
Maximum Performance (Максимальная производительность)	Минимальная потеря данных – не более нескольких секунд – в зависимости от полосы пропускания сети	ASync

# Практика

Для закрепления полученного материала на практике слушателям предлагается выполнить ряд заданий по управлению процессом репликаций:

- ❖ Подготовить primary для создания потоковой репликации;
- ❖ Клонировать primary database на standby database;
- ❖ Запустить реплику;
- ❖ Остановить реплику;
- ❖ Снова запустить реплику.