**ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**Khoa công nghệ thông tin**

**ĐỀ TÀI:**

**CÔNG NGHỆ DATAGUARD TRONG DỰ PHÒNG THẢM HỌA DỮ LIỆU**

**Sinh viên thực tập**

**Vương Sỹ Sơn**

**Hà Nội, 02/2018**

**MỤC LỤC**

[**Mở đầu 3**](#_Toc508141049)

[**ORACLE DATABASE 3**](#_Toc508141050)

[**I. Tổng quan 3**](#_Toc508141051)

[**II. Các thành phần của Oracle Database Server 4**](#_Toc508141052)

[**1. Oracle Instance 4**](#_Toc508141053)

[**2. Oracle Database 8**](#_Toc508141054)

[**SERVICE ORACLE DATA GUARD 11G 12**](#_Toc508141055)

[**I. Giới thiệu 12**](#_Toc508141056)

[**II. Kiến trúc Data Guard 13**](#_Toc508141057)

[**1. Mô hình 13**](#_Toc508141058)

[**2. Primary Database 14**](#_Toc508141059)

[**3. Standby Database 14**](#_Toc508141060)

[**4. Data Guard Protection Modes 16**](#_Toc508141061)

[**5. Oracle Data Guard Processes 17**](#_Toc508141062)

[**6. Data Guard Services 19**](#_Toc508141063)

[**III. Cấu hình Oracle Data Guard 24**](#_Toc508141064)

[**1. Yêu cầu cài đặt 24**](#_Toc508141065)

[**2. Cấu hình 25**](#_Toc508141066)

[**3. Thay đổi Protection Mode (thực hiện trên server Primary) 32**](#_Toc508141067)

[**4. Switchover 33**](#_Toc508141068)

[**5. Failover 34**](#_Toc508141069)

[**6. Snapshot Standby Database 35**](#_Toc508141070)

[**7. Logical Standby Database 36**](#_Toc508141071)

[**Các tài liệu tham khảo 38**](#_Toc508141072)

# **Mở đầu**

1. **Đặt vấn đề**

Trong thời kỳ bùng nổ công nghệ thông tin hiện nay, nhu cầu về dữ liệu ngày càng trở nên quan trọng đối với doanh nghiệp. Trong đó, CSDL nơi lưu trữ những dữ liệu quan trọng nhất của doanh nghiệp vì nó chứa các thông tin về khách hàng, tài chính, nghiệp vụ… phục vụ cho các ứng dụng sản xuất kinh doanh của doanh nghiệp. Ngoài ra, CSDL còn được sử dụng cho mục đích phân tích và khai thác dữ liệu phục vụ nhu cầu phân tích thị trường, phát triển sản phẩm, định hướng phát triển doanh nghiệp trong tương lai.

CSDL có thể nói là thành phần quan trọng nhất của hệ thống công nghệ thông tin, nơi lưu trữ thông tin tác nghiệp của một hệ thống công nghệ thông tin. Trong chiến lược an toàn thông tin của tổ chức thì an toàn dữ liệu và đảm bảo tính sẵn sàng của dữ liệu là yếu tố thực sự quan trọng. Trong một thế giới mà sự hài lòng khách hàng đôi khi đến từ những vấn đề rất nhỏ, hoạt động của hệ thống công nghệ thông tin nói chung và CSDL nói riêng chính là giúp cho tổ chức nâng cao uy tín và khả năng cạnh tranh.

Để việc vận hành cho CSDL hoạt động tốt, thì việc phòng ngừa gián đoạn hệ thống và mất mát dữ liệu dựa trên nguyên lý sử dụng dư thừa các hệ thống và phần mềm để loại bỏ các điểm lỗi đơn (lỗi 1 phần dẫn đến ngừng hoạt động của cả hệ thống) là công tác cực kỳ quan trọng. Nguyên lý sử dụng dư thừa áp dụng cho các CSDL quan trọng. Lỗi của người quản trị, dữ liệu bị hỏng do hệ thố ng hoặc phần mềm hoặc hỏng toàn bộ site có thể ảnh hưởng đến tính liên tục của CSDL. Kể cả CSDL được cấu hình cluster chạy trên nhiều máy chủ cũng lộ ra các điểm lỗi đơn nếu không được bảo vệ một cách thích hợp. Trong khi CSDL được cấu hình cluster có thể cung cấp tính sẵn sàng cao rất tốt ở mức máy chủ thì một hệ thống phụ thuộc chạy CSDL đơn kết nối với thiết bị lưu trữ là rất nguy hiểm. Cách duy nhất để phòng ngừa các ảnh hưởng của các điểm lỗi đơn là phải có bản sao đầy đủ của CSDL chạy trên hệ thống khác và lý tưởng nhất là được triển khai ở địa điểm thứ 2 và có thể sẵn sàng truy cập nếu CSDL chính ngừng phục vụ vì bất kỳ lý do gì. Oracle Data Guard là giải pháp sẵn sàng để loại bỏ các điểm lỗi đơn đối với các CSDL Oracle quan trọng.

1. **Lý do chọn đề tài**

Oracle Data Guard cung cấp bảo vệ dữ liệu thời gian thực và tính sẵn sàng trong khi loại bỏ thỏa hiệp vốn có với các giải pháp khác cho Oracle Database. Nó cho phép không mất dữ liệu khắc phục thảm họa (DR) qua bất kỳ khoảng cách mà không ảnh hưởng hiệu suất cơ sở dữ liệu. Nó sửa chữa tham nhũng vật lý mà không ảnh hưởng đến tính sẵn sàng và tiết kiệm băng thông mạng mà không cần mục đích đặc biệt  các thiết bị mạng. Nó làm giảm thời gian chết cho Oracle nâng cấp cơ sở dữ liệu mà không dễ bị lỗi thủ tục bằng tay. Nó làm tăng lợi tức đầu tư trong DR hệ thống sử dụng sự đơn giản của sao chép vật lý.

1. **Đối tượng nghiên cứu**

Đối tượng nghiên cứu: Oralce Database, Oracle Data Guard

1. **Mục đích**

Nghiên cứu kiến trúc Oracle Database và công nghệ Oracle Data Guard.

Các bước tiến hành cài đặt Oracle Data Guard.

1. **Ý nghĩa**

Đề tài giúp người đọc hiểu được lý thuyết về các thành phần của hệ quản trị cơ sở dữ liệu Oracle, kiến trúc của Oracle Data Guard và làm thế nào để sử dụng Oracle Data Guard để giúp bảo vệ cơ sở dữ liệu Oracle chống lại sự ngưng hoạt động theo kế hoạch và không có kế hoạch.

# **ORACLE DATABASE**

**Giới thiệu về Oracle Database**

## **I. Tổng quan**

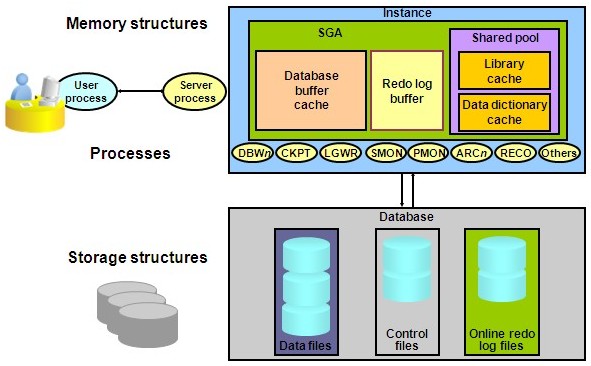
- Năm 1977 một chàng trai trẻ có tên là Lawrence Ellision sau khi đã đọc và tìm hiểu về mô hình quan hệ (relational model) của công ty máy tính IBM đã quyết định thành lập công ty Oracle. Sau đó hai năm (1979), công ty này đã phát hành sản phẩm đầu tay của mình chính là CSDL Oracle phiên bản 2.0, nó không những là một sản phẩm CSDL quan hệ mà còn tích hợp ngôn ngữ truy vấn có cấu trúc SQL. Năm 1983 phiên bản 3 được phát hành với các thay đổi trong ngôn ngữ SQL, mở rộng hiệu suất và các cải tiến. Năm 1985 phiên bản 5 ra đời đánh dấu một sự kiện quan trọng vì nó được áp dụng theo công nghệ khách chủ. Năm 1997 Oracle giới thiệu phiên bản 8, tiếp đến là phiên bản 8i (ver 8.1.5) hỗ trợ nhiều tính năng mới và việc phát triển ứng dụng CSDL trên Internet. Và với phiên bản Oracle 12c ra đời năm 2014 với nhiều tính năng nổi bật như hỗ trợ điện toán đám mây, dữ liệu lớn (big data),… , Oracle đã có thể cung cấp đa dạng các giải pháp lưu trữ và xử lý dữ liệu chuyên nghiệp cho các khách hàng theo nhiều hướng ứng dụng khác nhau.

**- Oracle Database**(hay còn gọi là**Oracle RDBMS** hoặc đơn giản là Oracle)**,** là 1**hệ quản trị cơ sở dữ liệu quan hệ**, được phát triển và phân phối bởi tập đoàn Oracle.

- Oracle Database là một tập các đơn vị dữ liệu. Mục đích của cơ sở dữ liệu là là lưu trữ và tìm lại được các thông tin được lưu trữ. Máy chủ cơ sử dữ liệu là chìa khóa của việc giải quyết vấn đề quản lý thông tin. Thông thường một máy chủ máy chủ có thể quản lý một lớn dữ liệu trong môi trường có nhiều người dùng có thể đồng thời truy cập cùng dữ liệu. Tất cả được hoàn thành vói hiệu năng cao. Máy chủ cơ sở dữ liệu có thể ngăn cản truy cập trái phép và cung cấp giải pháp hiệu quả trong việc láy lại dữ liệu khi có bất kỳ sự cố nào.

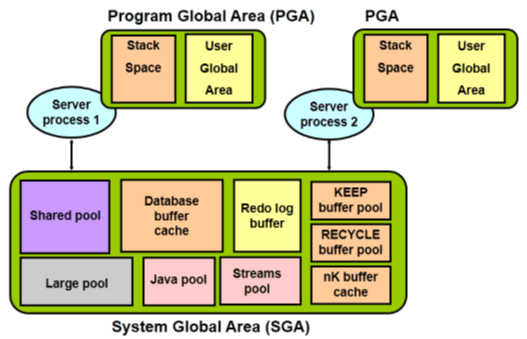
- Oracle Database là cơ sở dữ liệu đầu tiên được thiết kế cho máy tính lưới doanh nghiệp, tính linh hoạt và chi phí hiệu quả nhất để quản lý thông tin và ứng dụng. Hệ thống điện toán doanh nghiệp tạo ra các vùng chứa quy mô lớn và các máy chủ. Với kiến ​​trúc này, mỗi hệ thống mới có thể được cung cấp nhanh chóng từ các thành phần. Không cần khối lượng công việc tối đa, vì khả năng có thể dễ dàng thêm hoặc phân bổ lại từ các nguồn tài nguyên nếu cần.

## **II. Các thành phần của Oracle Database Server**



Kiến trúc của Oracle Database Server bao gồm 2 phần chính là: Oracle Instance và Oracle Database.

1. **Oracle Instance**
2. Cấu trúc bộ nhớ (Memory Structure)



Có 2 cấu trúc bộ nhớ cơ bản trong một Instance: SGA và PGA

* SGA (System Global Area): vùng bộ nhớ để lưu trữ dữ liệu và các thông tin điều khiển của Oracle Server, được khởi động khi một Instance khởi động.

+ Shared pool: vùng nhớ dành cho các đoạn SQL vừa được thực thi gần nhất và những định nghĩa dữ liệu được dùng gần nhất. Bao gồm: Library cache, Data dictionary, Server result cache.

+ Database buffer cache: chứa các [block dữ liệu](#Datablock) được đọc từ [Data File](#datafile). Khi một session cần dữ liệu, Server process sẽ đọc thông tin từ Database buffer cache để lấy các block dữ liệu cần thiết. Nếu các block dữ liệu không có trong Database buffer cache thì Server process mới đọc dữ liệu từ data file.

+ **Redo Log Buffer:** Là một bản ghi tạm thời, ghi lại tất cả những thay đổi trên các Data Block với mục đích chính là để phục hồi dữ liệu. Được thực hiện bởi các background process.

+ Large pool: Cung cấp một vùng nhớ lớn, được cấp phát cho các trường hợp như: vùng bộ nhớ cho UGA (User Global Area), xử lý I/O, sao lưu và phục hồi hệ thống.

+ Java Pool và Streams Pool: Java Pool là vùng dùng cho các Procedure viết bằng Java. Stream Pool dùng cho Oracle Stream.

* PGA (Program Global Area): một phần cấu trúc bộ nhớ dành cho User process kết nối tới một Instance, bao gồm dữ liệu và thông tin điều khiển cho một Server hoặc một Background process, được khởi động khi Server process khởi động.

1. Các tiến trình (Process Architecture)

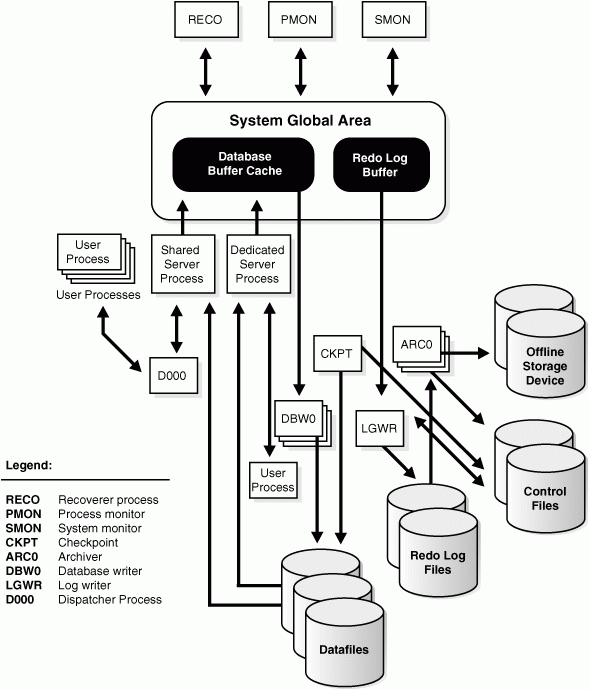
* Server process: là các process được sinh ra để phục vụ request từ user process.

+ Cấu hình dedicated server: một server process sẽ phục vụ cho một user process.

+ Cấu hình shared server: một server process sẽ phục vụ cho nhiều user process.

+ Nhiệm vụ:

* Phân tích và chạy các câu lệnh SQL từ client (user process).
* Đọc dữ liệu cần thiết từ disk lên Database buffer cache nếu chưa có.
* Trả kết quả về cho client.
* Background process: quản lý memory, process, quản lý I/O, giao tiếp giữa các thành phần…



Có các background process:

+ Database writer process (DBWn)

+ Log Writer process (LGWR)

+ Checkpoint process (CKPT)

+ System monitor process (SMON)

+ Recoverer process (RECO)

+ Job queue coordinator (CJQ0)

+ Job slave processes (Jnnn)

+ Archiver processes (ARCn)

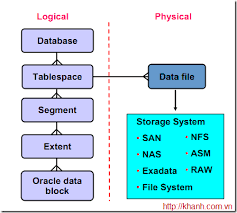
+ Queue monitor processes (QMNn)

1. **Oracle Database**

Vai trò của Oracle Database là lưu trữ và cung cấp thông tin cho người dùng thông qua.

Oracle Database quản lý thông tin lưu trữ thông qua hai thành phần lưu trữ vật lý và logic.

Thành phần lưu trữ vật lý là các tập tin. Thành phần lưu trữ logic là các không gian bảng (table space).



1. Cấu trúc vật lý của database

Dữ liệu trong Oracle Database được tổ chức lưu trữ và quản lý bởi 3 loại tệp tin: data file, control file, redo log file.

* Data File

+ Mỗi một Oracle database đều có thể có một hay nhiều datafiles. Các database datafiles chứa toàn bộ dữ liệu trong database. Các dữ liệu thuộc cấu trúc logic của database như tables hay indexes đều được lưu trữ dưới dạng vật lý trong các datafiles của database.

+ Tính chất:

* Mỗi datafile chỉ có thể được sử dụng trong một database.
* datafiles có một số tính chất cho phép tự động mở rộng kích thước mỗi khi database hết chỗ lưu trữ dữ liệu.
* Một hay nhiều datafiles tạo nên một đơn vị lưu trữ logic của database gọi là tablespace.
* Một datafile chỉ thuộc về một tablespace.

+ Dữ liệu trong một datafile có thể đọc ra và lưu vào vùng nhớ bộ đệm của Oracle. Trong trường hợp thông tin yêu cầu không có trong cache memory hiện thời, nó sẽ được đọc trực tiếp từ các datafiles ra và lưu trữ vào trong bộ nhớ.

* Redo Log File

+ Mỗi một Oracle Database có 2 hay nhiều Redo Log File, tập hợp những File này trong một CSDL được gọi là Redo Log của CSDL đó. Một Redo Log tạo thành những Redo Entry, còn gọi là Redo Record.

+ Chức năng chính của Redo Log là dùng để ghi lại những thông tin trong CSDL bị thay đổi.

+ Trong trường hợp Database gặp sự cố trước khi dữ liệu thay đổi được ghi nhận chắc chắn xuống các Data File thì những thông tin trong Redo Log sẽ được sử dụng trong việc phục hồi dữ liệu về trạng thái trước khi dữ liệu thay đổi.

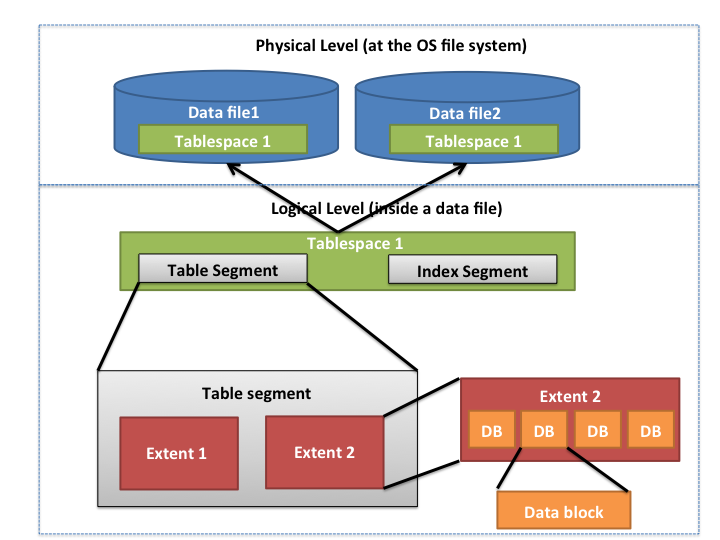
* Control File

+ Mỗi một Oracle Database có 1 Control File, dùng để chứa tất cả các thông tin về cấu trúc vật lý của một Database là tên Database, nơi lưu trữ của Data File và Redo Log File, Timestamp tạo Database. Mỗi Control File chỉ dùng cho một Database duy nhất.

+ Mỗi khi nào một instance của Oracle database được mở, control file của nó sẽ được sử dụng để xác định data files và các redo log files đi kèm.

+ Khi các thành phần vật lý cả database bị thay đổi (ví dụ như, tạo mới datafile hay redo log file), Control file sẽ được tự động thay đổi tương ứng bởi Oracle.

+ Control file cũng được sử dụng đến khi thực hiện khôi phục lại dữ liệu.

1. Cấu trúc logic 

Cấu trúc logic của Oracle database bao gồm các đối tượng tablespaces, schema objects, data blocks, extents, và segments.

* Tablespace:

+ Một database có thể có nhiều tablespace. Mỗi tablespace có thể tạo nên bởi nhiều datafiles.

+ Đặc điểm:

* Tablespace chỉ thuộc về một Database duy nhất.
* Một Tablespace bao gồm một hay nhiều tập tin vật lý dùng để lưu trữ dữ liệu, đó là Data File.

+ Trong Oracle Database luôn có một tablespace do Oracle tạo ra trong quá trình tạo DB có tên là system. Đây là vùng nhớ quan trọng, lưu trữ tất cả thông tin về Database như định nghĩa cấu trúc của các đổi tượng, thông tin cấp phát và sử dụng vùng nhớ của các đổi tượng,…

+ Tablespace được tạo nên bởi sự kết hợp của một hay nhiều đơn vị lưu trữ Logic gọi là Segment, một Segment được chia thành nhiều Extent và trong Extent thì có nhiều Data Block liên tục nhau.

* Schema và Schema Objects

Schema là tập hợp các đối tượng (objects) có trong database. Schema objects là các cấu trúc logic cho phép tham chiếu trực tiếp tới dữ liệu trong database. Schema objects bao gồm các cấu trúc như tables, views, sequences, stored procedures, synonyms, indexes, clusters, và database links.

* Data Blocks, Extents, Segments

+ Data Blocks

* Là mức phân cấp logic thấp nhất, các dữ liệu của Oracle database được lưu trữ trong các data blocks.
* Một data block tương ứng với một số lượng nhất định các bytes vật lý của database trong không gian đĩa cứng.
* Kích thước của một data block được chỉ ra cho mỗi Oracle database ngay khi database được tạo lập.
* Database sử dụng, cấp phát và giải phóng vùng không gian lưu trữ thông qua các Oracle data blocks.

+ Extents

* Là mức phân chia cao hơn về mặt logic các vùng không gian trong database
* Một extent bao gồm một số data blocks liên tiếp nhau, cùng được lưu trữ tại một thiết bị lưu giữ.
* Extent được sử dụng để lưu trữ các thông tin có cùng kiểu.

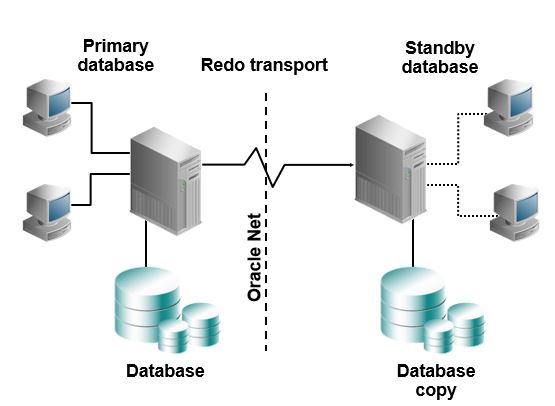
+ Segments

* Là mức phân chia cao hơn nữa về mặt logic các vùng không gian trong database.
* Một segment là một tập hợp các extents được cấp phát cho một cấu trúc logic.

# SERVICE ORACLE DATA GUARD 11G

## **I. Giới thiệu**

- Oracle Data Guard đảm bảo tính sẵn sàng cao, bảo vệ dữ liệu, và khôi phục sau sự cố. Data Guard cung cấp tất cả các dịch vụ khởi tạo, duy trì, quản lý và giám sát một hay nhiều standby database để Oracle database tiếp tục tồn tại sau sự cố, hư hỏng dữ liệu. Data Guard duy trì các CSDL dự phòng (Standby database) như là các bản sao của CSDL chính (Primary database). Nếu Primary database không sẵn sàng hoạt động, Data Guard có thể chuyển bất kỳ Standby database thành primary database, giảm thiểu thời gian ngưng hoạt động. Data Guard có thể sử dụng với sao lưu, khôi phục, và kỹ thuật cluster để tạo ra mức độ cao về bảo vệ dữ liệu và sẵn sàng dữ liệu.   
  
- Với Data Guard, có thể cải thiện hiệu năng của primary database bằng cách sao lưu và chạy các báo cáo trên hệ thống standby.



**Lợi ích của Oracle Data Guard**

* Đảm bảo an toàn dữ liệu trung tâm chính hoặc trung tâm từ xa trong trường hợp xảy ra thảm họa
* Máy chủ dự phòng cho phép truy xuất dữ liệu ở mức chỉ đọc trong khi đồng bộ với máy chủ chính
* Đảm bảo tính toàn vẹn của dữ liệu
* Tự động chuyển đổi vai trò giữa máy chính và máy dự phòng nhanh chóng
* Tăng hiệu suất máy chủ chính bằng việc chuyển các báo cáo sang máy chủ dự phòng
* Tận dùng tối đa đường truyền bằng tính năng nén dữ liệu
* Hỗ trợ chạy trên nền tảng các máy chủ Oracle, Exadata.
* Không giới hạn các loại dữ liệu, hệ thống lưu trữ và thao tác DDL và DML

**Primary Database**

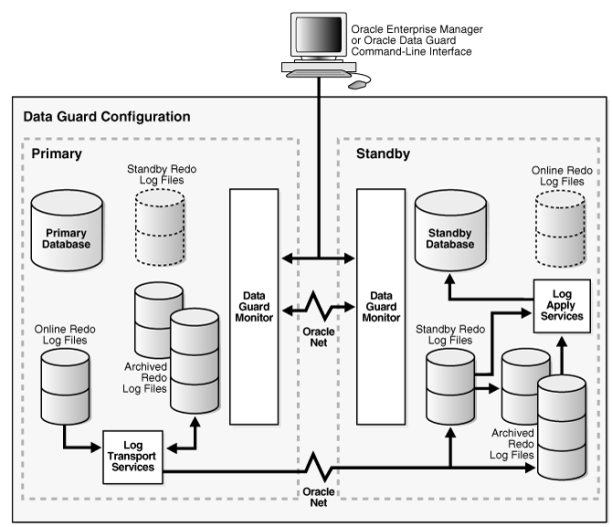
* Là cơ sở dữ liệu chính trong mô hình **Oracle Data Guard** mà các ứng dụng sẽ kết nối trực tiếp vào.
* Liên tục gửi những thay đổi dữ liệu sang Standby Database (*dưới dạng Redo data*).

**Standby Database**

* Hiểu một cách đơn giản CSDL dự phòng là bản copy của CSDL chính trong mô hình data guard. Như vậy sẽ có nhiều cách để tạo ra CSDL dự phòng từ CSDL chính.
* CSDL dự phòng sẽ nhận và apply redo data từ CSDL chính.
* CSDL chính và dự phòng có thể là Single hoặc RAC.
* Trong mô hình Oracle data guard, nếu CSDL chính gặp sự cố. Oracle data guard sẽ chuyển 1 trong các CSDL dự phòng thành CSDL chính. Như vậy, sẽ giảm thiểu thời gian chết của hệ thống.
* Tùy từng bài toán cụ thể **Oracle data guard** có thể triển khai cho việc an toàn dữ liệu, báo cáo và nâng cấp database.

## **II. Kiến trúc Data Guard**

1. **Mô hình**



**Giải thích:**

* Một CSDL được cấu hình Data Guard bao gồm 1 CSDL chính và nhiều CSDL khác được cấu hình ở chế độ standby
* Các CSDL được kết nối bởi nhau và không phụ thuộc vào vị trí địa lý, vd CSDL chính và dự phòng có thể nằm trên cùng một hệ thống hoặc có thể được đặt ở 2 khu vực khác nhau

**Công cụ quản lý:**

* SQL command
* Oracle Data Guard Broker

+ Oracle Enterprise Manager

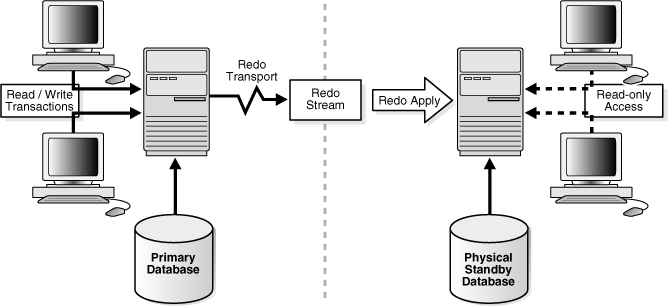
+ Oracle Data Guard Command-Line Interface

1. **Primary Database**

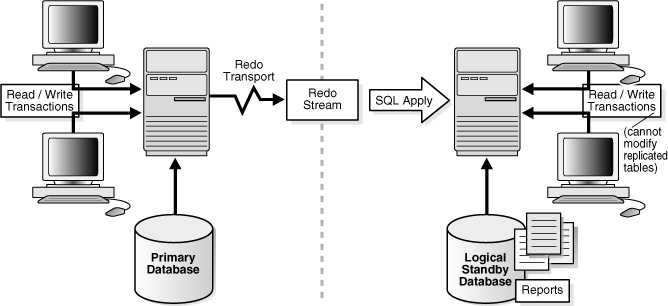
Là Cơ sở dữ liệu chính, được truy cập bởi hầu hết các ứng dụng.

1. **Standby Database**

* Là CSDL dự phòng, là một bản sao được tạo từ CSDL chính. Data Guard 10G cho phép tạo tối đa 9 CSDL dự phòng, còn bản 11G cho phép tạo 31 CSDL dự phòng
* Sau khi được khởi tạo, Data Guard sẽ tự động cập nhật dữ liệu từ CSDL chính sang CSDL dự phòng
* CSDL dự phòng được phân loại như sau
  1. Physical standby database



* Là bản sao chép vật lý đồng nhất của CSDL chính trên một block-for-block.
* Đồng bộ với CSDL chính thông qua việc nhận và apply redo data từ CSDL chính.
* Tùy từng bài toán mà Physical standby database có thể sử dụng cho trường hợp protection and reporting.
* Từ bản Oracle database 11g trở lên, physical standby database có thể sử dụng ở chế độ open read-only.
  1. Logical standby database



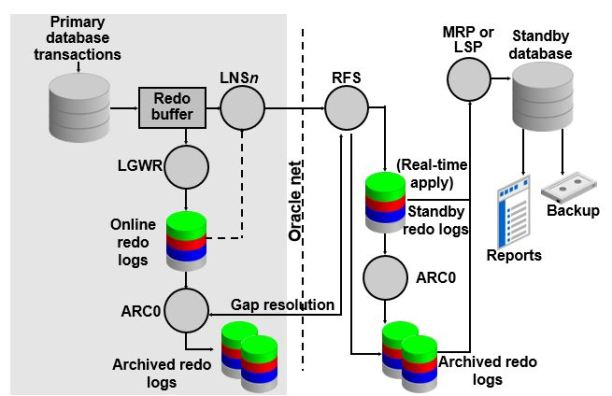
* Là bản sao chép của CSDL chính. Nó chứa các thông tin logic tương tự như CSDL chính. Mặc dù cấu tạo và kiến trúc của dữ liệu có thể khác nhau.
* Đồng bộ với CSDL chính thông qua apply redo. Khác với physical standby database, nó sẽ chuyển đổi dữ liệu trong redo nhận được từ CSDL chính thành các sql\_text và thực thi chúng (nó sử dụng LogMiner để đọc redo data nhận được từ CSDL chính).
* Tùy từng bài toán mà Logical standby database có thể sử dụng cho trường hợp protection, reporting và database upgrades.
  1. Snapshot standby database
* Là một CSDL được tạo bởi việc chuyển đổi một physical standby database vào một snapshot standby database.
* Nhận redo data từ CSDL chính nhưng không apply redo data cho đến khi nó chuyển đổi quay trở lại thành physical standby database.
* Thích hợp khi yêu cầu một phiên bản temporary, updatable của physical standby database.
* Có thể sử dụng cho việc testing.

1. **Data Guard Protection Modes**

Có 3 cơ chế bảo vệ dữ liệu trong Data Guard: Maximum Availability, Maximum Performance, Maximum Protection.

* 1. Maximum Availability
* Chế độ bảo vệ này cung cấp mức bảo vệ dữ liệu cao nhất có thể mà không ảnh hưởng đến tính sẵn sàng của CSDL chính.
* Transaction không được commit cho đến khi tất cả các redo data cần thiết để phục hồi transaction đó được ghi vào cả Online redo log và Standby redo log trên ít nhất một CSDL dự phòng đồng bộ.
* Nếu CSDL chính không thể ghi Redo stream cho ít nhất một CSDL dự phòng đồng bộ, nó sẽ hoạt động như ở chế độ Maximum performance để duy trì tính sẵn sàng của CSDL chính cho đến khi có thể ghi lại.
* Chế độ này đảm bảo dữ liệu không bị mất khi CSDL chính gặp lỗi, nhưng chỉ khi lỗi không làm ngăn chặn việc gửi các Redo data trước đó từ CSDL chính đến ít nhất một CSDL dự phòng.
  1. Maximum Performance
* Đây là chế độ bảo vệ mặc định khi cài Data Guard.
* Chế độ bảo vệ này cung cấp mức bảo vệ dữ liệu cao nhất có thể mà không ảnh hưởng đến hiệu suất của CSDL chính.
* Chế độ này được thực hiện bằng cách cho phép một Transaction commit trên CSDL chính ngay sau khi tất cả Redo data cần thiết để khôi phục Transaction được ghi vào Online redo log.
* Chế độ này cung cấp chế độ bảo vệ dữ liệu ít hơn một chút so với chế độ Maximum Availability và ảnh hưởng tối thiểu đến hiệu năng của CSDL chính.
  1. Maximum Protection
* Chế độ này đảm bảo dữ liệu không bị mất khi CSDL chính gặp lỗi.
* Để cung cấp chế độ này, Redo data cần thiết để phục hồi một Transaction phải được ghi vào cả Online redo log và Standby redo log trên ít nhất một CSDL dự phòng trước khi Commit transaction.
* Để đảm bảo rằng mất dữ liệu không bị mất khi hệ thống bị lỗi, CSDL chính sẽ shutdown.
* Transaction trên CSDL chính được bảo vệ ngay sau khi Data guard ghi các Redo data để lưu trữ liên tục trong tệp Standby redo log. Sau đó, xác nhận nhanh chóng được gửi trở lại để CSDL chính có thể tiến hành các transaction tiếp theo.
* Đây là chế độ an toàn cao nhất trong Data guard, nên tùy từng bài toán cụ thể chúng ta sẽ triển khai các chế độ Protection trong Data guard cho phù hợp.

1. **Oracle Data Guard Processes**

****

* 1. Process trên Primary database.

Trên CSDL chính sử dụng 2 process là Log Writer (LGWRn) và Archiver (ARCn):

* LGWRn:

+ Thu thập thông tin các Redo transaction và cập nhật vào Online Redo Logs trên CSDL chính.

+ Đối với Synchronous standby database: LGWR sẽ truyền Redo vào một tiến trình LNS (Log Network Server), và chuyển trực tiếp đến tiến trình RFS (Remote File Server) trên CSDL dự phòng. LGWR đợi xác nhận từ tiến trình LNS trước khi xác nhận commit.

+ Đối với asynchronous (ASYNC) standby databases: Tiến trình LNS đọc độc lập redo từ redo log buffer trên bộ nhớ hoặc trong online redo log file, và gửi các redo đến CSDL dự phòng.

* ARCn:

+ Tạo ra một bản sao của online redo log file vào archive redo log sử dụng cho việc phục hồi CSDL chính.

+ Chịu trách nhiệm chuyển redo data đến tiến trình RFS trên CSDL dự phòng và chủ động phát hiện xử lý GAP trên tất cả các CSDL dự phòng.

\* Nếu kết nối giữa CSDL chính và một hoặc nhiều CSDL dự phòng gặp sự cố, thì các redo data được tạo ra trên CSDL chính sẽ không thể gửi đến các CSDL dự phòng.

\* Khi một kết nối được thiết lập lại, oraccle data guard sẽ tự động tìm kiếm các Archived redo log files bị thiếu (gọi là GAP). Và sẽ tự động chuyển các Archived redo log files bị thiếu vào các CSDL dự phòng bằng cách sử dụng tiến trình ARCn (Arhiver process). Lúc này các CSDL dự phòng sẽ đồng bộ với CSDL chính mà không cần sự can thiệp của DBA.

* 1. Process trên Standby database

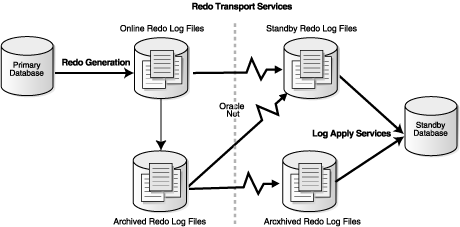
Trên CSDL chính sử dụng 4 process là Remote File Server (RFS), Archiver (ARCn), Managed Recovery (MRP) và Logical Standby (LSP):

* RFS: Nhận thông tin Redo log từ CSDL chính và có thể ghi redo vào Standby redo log Files hoặc Archived redo log Files.
* ARCn: Ghi các Redo log từ Standby redo logs vào Archived redo logs.
* MRP: Chỉ có trên Physical standby database, nó sẽ apply Redo log từ Archived redo log Files vào physical standby database.
* LSP: Chỉ có trên Logical standby database, nó sẽ apply Redo log từ Archived redo log Files vào logical standby database.
* Trong trường hợp Real Time Apply, cả MRP và LSP đều apply Redo log từ Standby redo log Files vào CSDL dự phòng.

1. **Data Guard Services**

Cơ chế cập nhật dữ liệu từ CSDL chính sang CSDL dự phòng được thực hiện bởi các services sau:

* 1. Redo Transport Services



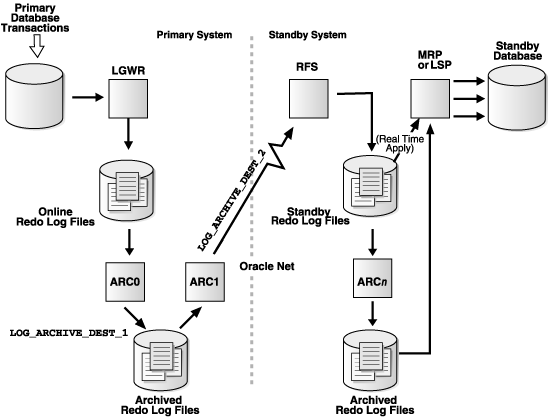
* Redo transport services kiểm soát tự động chuyển đổi của Redo data từ CSDL chính sang một hoặc nhiều điểm lưu trữ.
* Redo transport services thực hiện các công việc sau:

+ Chuyển các Redo data từ CSDL chính sang các CSDL dự phòng.

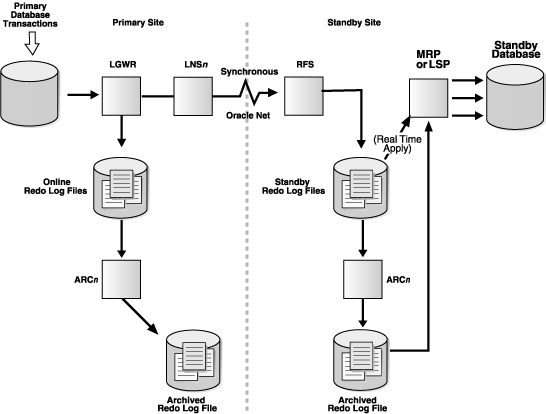
+ Quản lý tiến trình giải quyết lỗi thiếu các tệp của Archived log files.

+ Tự động phát hiện các Archived redo log files bị thiếu hoặc bị hỏng trên một CSDL dự phòng và tự động lấy lại từ CSDL chính hoặc một CSDL dự phòng khác.

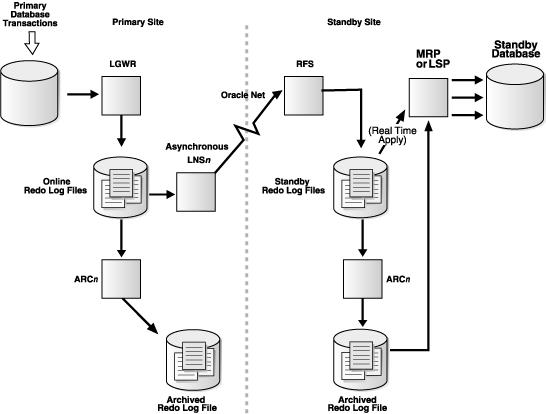
* + 1. Truyền Redo log sử dụng Archiver (ARCn) process



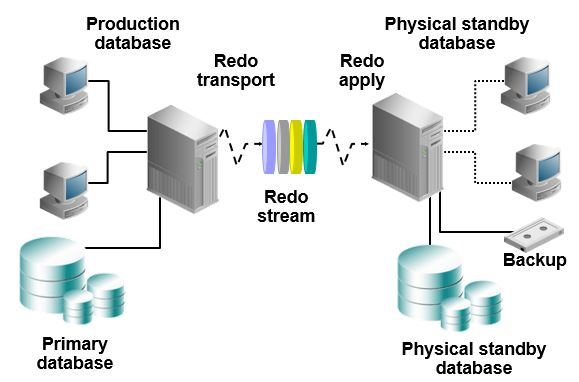
* Mặc định Transport service sử dụng ARCn process để truyền dữ liệu online Redo log.
* ARCn chỉ sử dụng khi thiết lập tham số Data Guard Protection Mode là Maximum Performance.
* Tiến trình này ARC0 sẽ tạo ra các Archive redo log.
* ARC1 sẽ truyền các Archive redo log được sinh ra từ CSDL chính sang CSDL dự phòng.
* Remote File Server (RFC process) trên máy dự phòng sẽ ghi các Archive redo log vào Standby Redo log trên máy dự phòng, Standby redo log file sẽ được tiến trình Archiver ghi vào Archived redo log file.
  + 1. Truyền Redo log sử dụng Log writer (LGWRn) process
* Có thể sử dụng với các thiết lập khác của Data Guard Protection Mode.
* Tiến trình này chuyển các Redo log sang CSDL dự phòng theo 2 phương thức Synchronize (SYNC) và Asynchronize (ASYNC), trến máy Stanby, tiến trình RFS ghi các redo log vào Stanby Redo log files.
  1. LGWRn sử dụng Synchronous Standby database



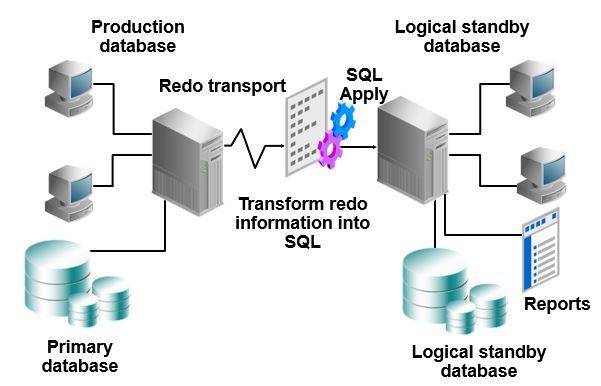
* Khi có thay đổi trên CSDL chính, tiến trình LGWR sẽ vừa ghi Redo log lên Online Redo log file, vừa chuyển trực tiếp Redo log sang tiến trình LNSn.
* Tiến trình LNSn sẽ đồng bộ sang CSDL dự phòng. Việc cập nhật dữ liệu phải được tiến hành tuần tự. Tiến trình mới sẽ không được thực hiện nếu vịêc cập nhật dữ liệu trước đó sang CSDL dự phòng chưa hoàn thành.
  1. LGWRn sử dụng Asynchronous Standby database



* Khi có thay đổi trên CSDL chính, tiến trình LGWR sẽ ghi dữ liệu lên Online Redo log files.
* Tiến trình LNSn (1 ..n) sẽ lấy Redo log từ Online Redo log Files hoặc Redo Buffer và đồng bộ sang CSDL dự phòng.
* Các thay đổi trên CSDL chínhsẽ được cập nhật liên tục sang database standby mà không cần đợi tiến trình cập nhật trước đó hoàn thành.
  1. Apply Services
* Redo data được truyền từ CSDL chính được ghi vào Standby redo log trên CSDL dự phòng. Apply services tự động áp dụng Redo data trên CSDL dự phòng để duy trì tính nhất quán với CSDL chính. Nó cũng cho phép truy cập dữ liệu ở chế độ read-only.
* Khác biệt chính giữa Physical và Logical standby databases là cách Apply services áp dụng archived redo data. Physical standby database áp dụng redo log bằng Redo Apply còn Logical standby database áp dụng redo log bằng SQL Apply.
  + 1. Redo Apply



* Được dùng cho Physical standby database.
* Công nghệ Redo apply áp dụng Redo data trên CSDL dự phòng bằng cách sử dụng các kỹ thuật khôi phục tiêu chuẩn của một Oracle Database.
* Redo apply sử dụng công cụ Oracle Media Recovery để apply các Redo data từ CSDL chính sang CSDL dự phòng.
  + 1. SQL Apply



* Được dùng cho Logical standby database.
* Trước tiên biến đổi dữ liệu redo đã nhận từ Primay database thành các câu lệnh SQL và sau đó thực thi các câu lệnh SQL đó trên CSDL dự phòng.
  1. Role Transitions

Thay đổi vai trò của Database từ CSDL dự phòng sang CSDL chính hoặc ngược lại bằng cách sử dụng switchover hoặc failover.

* + 1. Switchover

Switchover là sự đảo ngược vai trò giữa CSDL chính và một trong các CSDL dự phòng. Việc chuyển đổi đảm bảo không mất dữ liệu. Switchover thường được được sử dụng trong trường hợp chúng ta có kế hoạch chuẩn bị trước cho việc bảo trì hệ thống (software hoặc hardware).

* + 1. Failover

Trong môi trường Oracle data guard, khi CSDL chính gặp sự cố thì sẽ thực hiện Failover. Failover chỉ được thực hiện trong trường hợp CSDL chính gặp sự cố, và thực hiện Failover trên một CSDL dự phòng được lựa chọn để chuyển đổi vai trò của nó sang CSDL chính. Database Administrator có thể định cấu hình Data Guard để đảm bảo không mất dữ liệu.

## **III. Cấu hình Oracle Data Guard**

### **Yêu cầu cài đặt**

Môi trường: CentOS 6.5

Phiên bản Oracle: 11g (11.2.0.1)

* Máy chủ Primary
  + Hostname: db1
  + ORACLE\_SID: SANERO
  + IPADDRESS: 192.168.10.101
  + Mount point:

Filesystem Size Used Avail Use% Mounted on

/dev/sda5 15G 8.3G 5.5G 60% /

tmpfs 940M 467M 474M 50% /dev/shm

/dev/sda1 20G 172M 19G 1% /archivelog

/dev/sda7 12G 158M 11G 2% /backup

/dev/sda2 20G 1.7G 17G 10% /redo01

/dev/sda3 20G 1.7G 17G 10% /redo02

/dev/sda10 31G 11G 19G 36% /u01

/dev/sda6 15G 165M 14G 2% /u02

/dev/sda8 12G 385M 11G 4% /var

* Máy chủ Standby
  + Hostname: db2
  + ORACLE\_SID: SANERO
  + IPADDRESS: 192.168.10.102
  + Mount point:

Filesystem Size Used Avail Use% Mounted on

/dev/sda5 15G 8.3G 5.5G 60% /

tmpfs 940M 467M 474M 50% /dev/shm

/dev/sda1 20G 172M 19G 1% /archivelog

/dev/sda7 12G 158M 11G 2% /backup

/dev/sda2 20G 1.7G 17G 10% /redo01

/dev/sda3 20G 1.7G 17G 10% /redo02

/dev/sda10 31G 8.5G 21G 27% /u01

/dev/sda6 15G 165M 14G 2% /u02

/dev/sda8 12G 385M 11G 4% /var

* + Lưu ý: Trên máy chủ Standby chỉ cài software, không cài database.

### **Cấu hình**

* 1. Cấu hình trên máy chủ Primary
  2. Logging
* Kiểm tra log mode và force logging của CSDL chính

SQL> Select database\_role, log\_mode, force\_logging, flashback\_on from v$database;

DATABASE\_ROLE, LOG\_MODE, FORCE\_LOGGING, FLASHBACK\_ON

-----------------------------------------------------------------------------------------------

PRIMARY NOARCHIVELOG NO NO

* Nếu database chưa chạy ở chế độ archive log thì Enable Archiving trên CSDL chính như sau ( lưu ý: có downtime):

SQL> shutdown immediate;

SQL> startup mount;

SQL> alter database archivelog;

SQL> alter database open;

Check lại: SQL> archive log list;

* Enable Force Logging:

SQL> alter database force logging;

* 1. Tạo các Standby Redo logfile group

SQL> alter database add standby logfile group 4 ('/redo01/stb\_redo01\_1.log','/redo02/stb\_redo01\_2.log') size 512M;

SQL> alter database add standby logfile group 5 ('/redo01/stb\_redo02\_1.log','/redo02/stb\_redo02\_2.log') size 512M;

SQL> alter database add standby logfile group 6 ('/redo01/stb\_redo03\_1.log','/redo02/stb\_redo03\_2.log') size 512M;

SQL> alter database add standby logfile group 7 ('/redo01/stb\_redo04\_1.log','/redo02/stb\_redo04\_2.log') size 512M;

*Lưu ý:*

* Dung lượng file log standby phải bằng dung lượng file redo log.
* Số lượng standby log tối thiểu cần tạo được tính như sau: (threads)\*(groups Per Threads + 1).
  + Với DB chạy đơn thì số thread = 1, với DB RAC nhiều node thì số thread = số node.
  + Group per thread: theo guideline của Tập đoàn thì mỗi thread cần có tối thiếu 3 group.
  + Trong trường hợp cấu hình dataguard chạy đơn thông thường cần 4 standby logfile group.
  1. Thay đổi tham số trong file pfile của database: initSANERO.ora
* Tạo backup cho database

$ rman target /

RMAN> backup database plus archivelog;

* Thay đổi Initialization Parameter trong file $ORACLE\_BASE/dbs/initSANERO.ora

(có thể sử dụng sql command trong sqlplus để thay đổi tham số)

\*.db\_name='SANERO'

\*.db\_unique\_name='PRIMARY'

\*.fal\_client='TO\_PRIMARY'

\*.fal\_server='TO\_STANDBY'

\*.log\_archive\_config='DG\_CONFIG=(PRIMARY,STANDBY)'

\*.LOG\_ARCHIVE\_DEST\_1='LOCATION=/archivelog/ VALID\_FOR=(ALL\_LOGFILES,ALL\_ROLES) DB\_UNIQUE\_NAME=PRIMARY'

\*.LOG\_ARCHIVE\_DEST\_2='SERVICE=TO\_STANDBY LGWR ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=STANDBY'

\*.log\_archive\_dest\_state\_1='ENABLE'

\*.log\_archive\_dest\_state\_2='ENABLE'

\*.LOG\_ARCHIVE\_FORMAT='SANERO\_%t\_%s\_%r.arc'

\*.LOG\_ARCHIVE\_MAX\_PROCESSES=30

\*.STANDBY\_FILE\_MANAGEMENT='AUTO'

\*.service\_names='PRIMARY'

\*.db\_create\_online\_log\_dest\_1='/redo01'

\*.db\_create\_online\_log\_dest\_2='/redo02'

* Thêm 2 tham số sau trong trường hợp ORACLE\_SID (db\_name) của primary và CSDL dự phòng khác nhau:

*Ví dụ: db\_name trên primary là PRIMARY còn standby là STANDBY*

\*.DB\_FILE\_NAME\_CONVERT='STANDBY','PRIMARY'

\*.LOG\_FILE\_NAME\_CONVERT='STANDBY','PRIMARY'

* 1. Restart database

SQL> shutdown immediate;

SQL> startup nomount pfile='/u01/app/oracle/product/11.2.0/db\_1/dbs/initSANERO.ora';

SQL> create spfile from pfile;

* 1. Tạo standby controlfile

SQL> alter database mount;

SQL> alter database create standby controlfile as '/u01/app/oracle/oradata/SANERO/controlstandby01.ctl';

SQL> shutdown abort;

*Lưu ý:*

* Copy controlstandby01.ctl để tạo 2 standby controlfile.
* Shutdown database để đảm bảo datafile của primary và CSDL dự phòng giống nhau (bước này có thể thực hiện ngay trước khi copy datafile từ máy primary sang máy standby)
  1. Cấu hình Service và Listener
* Cấu hình tnsnames.ora và listener.ora trên cả 2 server giống nhau.
* Có thể cấu hình Service và Listener bằng Network Configuration Utility (netca) hoặc manually.
* Thay đổi phần màu đỏ phù hợp với IPADDRESS của từng server.

1. Service (tnsnames.ora)

SANERO =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.10.101)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = SHARED)

(SERVICE\_NAME = SANERO)

)

)

#--------------DG------------------------

TO\_STANDBY =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.10.102)(PORT = 1521))

(CONNECT\_DATA =

(SERVICE\_NAME = SANERO)

)

)

TO\_PRIMARY =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.10.101)(PORT = 1521))

(CONNECT\_DATA =

(SERVICE\_NAME = SANERO)

)

)

1. Listener (listener.ora)

LISTENER =

(DESCRIPTION\_LIST =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.10.101)(PORT = 1521))

)

)

SID\_LIST\_LISTENER =

(SID\_LIST =

(SID\_DESC =

(SID\_NAME = PLSExtProc)

(ORACLE\_HOME = /u01/app/oracle/product/11.2.0/db\_1)

(PROGRAM = extproc)

)

(SID\_DESC =

(SID\_NAME = SANERO)

(ORACLE\_HOME = /u01/app/oracle/product/11.2.0/db\_1)

(GLOBAL\_DBNAME = SANERO)

)

)

ADR\_BASE\_LISTENER = /u01/app/oracle

1. Start listener trên cả 2 server

$lsnrctl stop

$lsnrctl start

1. Check tnsping trên cả 2 server

* Trên server Primary: $tnsping TO\_STANDBY
* Trên server Standby: $tnsping TO\_PRIMARY
  1. Cấu hình trên máy chủ Standby

1. Tạo các thư mục giống như trên server Primary

$ mkdir -p /u01/app/oracle/oradata/SANERO

$ mkdir -p /u01/app/oracle/admin/SANERO/adump

$ mkdir -p /u01/app/oracle/admin/SANERO/pfile

$ mkdir -p /u01/app/oracle/admin/SANERO/dpdump

$ mkdir -p /u01/app/oracle/diag

$ mkdir -p /u01/app/oracle/flash\_recovery\_area/SANERO

1. Clone database từ server Primary sang server Standby

* Copy online redo logs từ server Primary sang server Standby vào các thư mục tương ứng

$ scp -r /redo01/stb\* oracle@192.168.10.102:/redo01/

$ scp -r /redo02/stb\* oracle@192.168.10.102:/redo02/

$ scp -r /archivelog/\* oracle@192.168.10.102:/archivelog/

* Copy controlfilestb01.ctl từ server Primary sang server Standby vào các thư mục tương ứng.

$ scp -r /u01/app/oracle/oradata/SANERO/controlstandby01.ctl oracle@192.168.10.102:/u01/app/oracle/oradata/SANERO

* Trên server Standby đổi tên controlfilestb01.ctl thành controlfile01.ctl. sau đó copy file, đặt tên là controlfile02.ctl và lưu file đó vào thư mục tương ứng với controlfile02.ctl trên server Primary.
* Copy tnsnames.ora va listener.ora sang server Standby vào các thư mục tương ứng và sửa IPADDRESS.

$ scp –r /u01/app/oracle/product/11.2.0/db\_1/network/admin/tnsnames.ora oracle@192.168.10.102:/u01/app/oracle/product/11.2.0/db\_1/network/admin/

$ scp –r /u01/app/oracle/product/11.2.0/db\_1/network/admin/listener.ora oracle@192.168.10.102:/u01/app/oracle/product/11.2.0/db\_1/network/admin/

* Copy file orapwSANERO trong thu muc $ORACLE\_HOME/dbs/ từ server Primary sang server Standby vào các thư mục tương ứng

$ scp -r /u01/app/oracle/product/11.2.0/db\_1/dbs/orapwSANERO oracle@192.168.10.102:/u01/app/oracle/product/11.2.0/db\_1/dbs/

* Copy các datafile từ server Primary sang server Standby vào các thư mục tương ứng

$ scp -r /u01/app/oracle/oradata/SANERO oracle@192.168.10.102:/u01/app/oracle/oradata/

* Copy pfile initSANERO.ora từ server Primary sang server Standby vào các thư mục tương ứng

$ scp $ORACLE\_HOME/dbs/initSANERO.ora oracle@192.168.10.102:$ORACLE\_HOME/dbs/

1. Thay đổi tham số pfile initSANERO.ora trên server Standy

\*.db\_name='SANERO'

\*.db\_unique\_name='STANDBY'

\*.fal\_client='TO\_STANDBY'

\*.fal\_server='TO\_PRIMARY'

\*.log\_archive\_config='DG\_CONFIG=(PRIMARY,STANDBY)'

\*.LOG\_ARCHIVE\_DEST\_1='LOCATION=/archivelog/ VALID\_FOR=(ALL\_LOGFILES,ALL\_ROLES) DB\_UNIQUE\_NAME=STANDBY'

\*.LOG\_ARCHIVE\_DEST\_2='SERVICE=TO\_PRIMARY LGWR ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=PRIMARY'

\*.log\_archive\_dest\_state\_1='ENABLE'

\*.log\_archive\_dest\_state\_2='ENABLE'

\*.LOG\_ARCHIVE\_FORMAT='DATA\_%t\_%s\_%r.arc'

\*.LOG\_ARCHIVE\_MAX\_PROCESSES=30

\*.STANDBY\_FILE\_MANAGEMENT='AUTO'

\*.service\_names='STANDBY'

\*.db\_create\_online\_log\_dest\_1='/redo01'

\*.db\_create\_online\_log\_dest\_2='/redo02'

1. Start standby database

SQL> shutdown immediate;

SQL> startup nomount pfile='/u01/app/oracle/product/11.2.0/db\_1/dbs/initSANERO.ora';

SQL> create spfile from pfile;

SQL> alter database mount;

* 1. Start Appy Process

Thực hiện rên db Standby:

* Foreground redo apply. Session never returns until cancel.

SQL> alter database recover managed standby database;

* Background redo apply. Control is returned to the session once the apply process is started.

SQL> alter database recover managed standby database disconnect from session;

* Muốn dừng Media Recovery Manager sử dụng câu lệnh sau:

SQL> alter database recover managed standby database cancel;

* Bạn cũng có thể đặt thời gian delay apply archived redo log sau khi đã nhận được từ server Primary:

SQL> alter database recover managed standby database cancel;

SQL> alter database recover managed standby database delay 30 disconnect from session;

* Apply nodelay:

SQL> alter database recover managed standby database cancel;

SQL> alter database recover managed standby database nodelay disconnect from session;

* Sử dụng Real Time Apply:

SQL> alter database recover managed standby database using current logfile;

* 1. Xác minh cơ sở dữ liệu dự phòng đang hoạt động đúng
* Trên db Primary thực hiện các câu query sau:

SQL>select max(sequence#) from v$archived\_log;

SQL>alter system switch logfile;

SQL>archive log list;

SQL>select max(sequence#) from v$archived\_log;

* Trên db Standby thực hiện các câu query sau:

SQL> select DATABASE\_ROLE,OPEN\_MODE from v$database;

SQL>archive log list;

SQL>select max(sequence#) from v$archived\_log;

SQL>select max(sequence#), applied from v$archived\_log group by applied;

* Chuyển Standby về chế độ read only

SQL>alter database recover managed standby database cancel;

SQL>alter database open read only;

SQL>alter database recover managed standby database disconnect from session;

* 1. Active Data Guard

SQL> recover managed standby database cancel;

SQL> alter database open read only; //có thể bỏ qua bước này nếu database đã được mở ở chế độ read only.

SQL> recover managed standby database disconnect using current logfile;

* 1. Kiểm tra lại

Insert vào db primary. Select trên db standby

1. **Thay đổi Protection Mode (thực hiện trên server Primary)**

* Kiểm tra Protection mode hiện tại:

SQL> select protection\_mode from v$database;

* Maximum Availability.

SQL> alter system set log\_archive\_dest\_2='service=TO\_STANDBY affirm sync valid\_for=(online\_logfiles,primary\_role) db\_unique\_name=STANDBY';

SQL> alter database set standby database to maximize availability;

* Maximum Performance.

SQL> alter system set log\_archive\_dest\_2='service=TO\_STANDBY noaffirm async valid\_for=(online\_logfiles,primary\_role) db\_unique\_name=STANDBY';

SQL> alter database set standby database to maximize performance;

* Maximum Protection.

SQL> alter system set log\_archive\_dest\_2='service=TO\_STANDBY affirm sync valid\_for=(online\_logfiles,primary\_role) db\_unique\_name=Standby';

SQL> shutdown immediate;

SQL> startup mount;

SQL> alter database set standby database to maximize protection;

SQL> alter database open;

1. **Switchover**

* Trên db Primary cũ:

SQL> alter database commit to switchover to standby; //thêm with session shutdown nếu switchover\_status là session;

SQL> shutdown immediate;

SQL> startup nomount;

SQL> alter database mount standby database;

SQL> alter system set log\_archive\_dest\_state\_2=defer;

* Trên db Standby cũ

SQL> alter database commit to switchover to primary;

SQL> shutdown immediate;

SQL> startup

* Trên db Standby mới

SQL> alter system set log\_archive\_dest\_state\_2=enable;

SQL> recover managed standby database disconnect from session;

SQL> recover managed standby database cancel;

SQL> alter database open read only;

SQL> recover managed standby database disconnect using log file;

1. **Failover**

* Trên db Standby cũ:

SQL> alter database recover managed standby database finish;

SQL> alter database activate standby database;

* Chuyển đổi db Primary cũ sau khi được khôi phục thành db Standby mới:

+ Kiểm tra thời điểm thực hiện Failover

SQL> select to\_char(standby\_became\_primary\_scn) from v$database;

+ Khôi phục Primary tới thời điểm standby\_became\_primary\_scn đã xác định ở trên. Có thể sử dụng 1 trong 3 cách sau:

C1: Sử dụng flash back

SQL> shutdown immediate;

SQL> startup mount;

SQL> flashback database to scn <standby\_became\_primary\_scn>;

C2: Sử dụng file backup

RMAN> run

{

set until scn <standby\_became\_primary\_scn + 1>;

restore database;

recover database;

}

C3: Chỉ sử dụng khi mới back up db Primary trong một vài giờ trước

SQL> recover database using backup controlfile until change <standby\_became\_primary\_scn + 1>;

+ Chuyển đổi Primary thành Physical Standby

SQL> alter database convert to physical standby;

SQL> shutdown immediate;

SQL> startup mount;

1. **Snapshot Standby Database**
   1. Convert Physical to Snapshot standby database

* Sử dụng câu lệnh sau trên server Primary để chuyển đổi sang snapshot standby database:

SQL> recover managed standby database cancel;

SQL> alter database convert to snapshot standby;

SQL> alter database open

* Có thể sử dụng Snapshot standby database để testing: ví dụ tạo một schema mới trên snapshot nhưng khi chuyển Snapshot trở lại thành Physical standby database thì schema đó sẽ mất

SQL> create user SBE identified by 351;

SQL> grant connect,resource to SBE;

SQL> connect SQB/351

SQL> create table temp (id number);

SQL> insert into temp values (1);

SQL> select \* from temp;

* 1. Convert Snapshot to Physical standby database
* Phải đảm bảo Protection Mode của server Primary không là Maximum Protection

SQL> shutdown immediate;

SQL> startup mount;

SQL> alter database convert to physical standby;

SQL> shutdown immediate;

SQL> startup nomount;

SQL> alter database mount standby database;

SQL> alter database open read only;

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

1. **Logical Standby Database**

* B1: Tạo physical standby database
* B2: Chuyển phýsical standby database sang logical standby database:

+ Dừng apply redo log từ primary db trên server standby

SQL> recover managed standby database cancel;

+ Tạo thư mục chứa archive redo log được sinh ra trên cả 2 server

$ mkdir -p /archivelog/online

$ mkdir -p /archivelog/standby

+ Thay đổi tham số trên server chính

SQL> alter system set log\_archive\_dest\_1='location=/archivelog/online/ valid\_for=(online\_logfiles,all\_roles) db\_unique\_name=primary' scope=both;

SQL> alter system set log\_archive\_dest\_2='service=to\_standby async valid\_for=(online\_logfiles,primary\_role) db\_unique\_name=standby' scope=both;

SQL> alter system set log\_archive\_dest\_3='location=/archivelog/standby/ valid\_for=(standby\_logfiles,standby\_role) db\_unique\_name=primary' scope=both;

SQL> alter system set log\_archive\_dest\_state\_1=enable scope=both;

SQL> alter system set log\_archive\_dest\_state\_2=enable scope=both;

SQL> alter system set log\_archive\_dest\_state\_3=enable scope=both;

SQL> execute dbms\_logstdby.build;

+ Trên server dự phòng thực hiện lệnh sau để chuyển đổi sang logical standby

SQL> alter database recover to logical standby SANERODG {db\_name};

+ Khởi động lại server và thay đổi các tham số

SQL> shu immediate;

SQL> startup mount;

SQL> alter system set LOG\_ARCHIVE\_DEST\_1='LOCATION=/archivelog/online/ VALID\_FOR=(ONLINE\_LOGFILES,ALL\_ROLES) DB\_UNIQUE\_NAME=STANDBY' scope=both;

SQL> alter system set LOG\_ARCHIVE\_DEST\_2='SERVICE=TO\_PRIMARY ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=PRIMARY';

SQL> alter system set LOG\_ARCHIVE\_DEST\_3='LOCATION=/archivelog/standby/ VALID\_FOR=(STANDBY\_LOGFILES,STANDBY\_ROLE) DB\_UNIQUE\_NAME=STANDBY' scope=both;

SQL> alter system set LOG\_ARCHIVE\_DEST\_STATE\_1=ENABLE scope=both;

SQL> alter system set LOG\_ARCHIVE\_DEST\_STATE\_2=ENABLE scope=both;

SQL> alter system set LOG\_ARCHIVE\_DEST\_STATE\_3=ENABLE scope=both;

+ Mở db để đọc ghi dữ liệu

SQL> ALTER DATABASE OPEN RESETLOGS;

+ Bắt đầu apply các redo log nhận được từ db chính

SQL> alter database start logical standby apply immediate;

+ Xác nhận cơ sở dữ liệu dự phòng đang hoạt động

SQL> select max(sequence#) from dba\_logstdby\_log;

* Quản lý thông tin archive log file

+ Trên server primary:

SQL> select thread#, sequence# from v$log where status='current';

SQL> alter system archive log current;

+ Trên server dự phòng

SQL> select file\_name from dba\_logstdby\_log where thread#=1 and sequence#= (select sequence# from dba\_logstdby\_log);

SQL> select applied\_scn, newest\_scn from dba\_logstdby\_progress;

+ Theo dõi hoạt động của sql apply trên dự phòng

SQL> select sid, serial#, spid,type, high\_scn from v$logstdby\_process;

SQL> select applied\_scn, latest\_scn from v$logstdby\_progress;

# **Các tài liệu tham khảo**

1. <https://docs.oracle.com/cd/E11882_01/server.112/e41134/toc.htm>
2. <https://vnoracle.wordpress.com/2016/07/28/oracle-data-guard-architecture-p1/>
3. <https://oracle-base.com/articles/11g/data-guard-setup-11gr2>
4. <http://www.oracledistilled.com/oracle-database/switchover-to-a-physical-standby-using-the-data-guard-command-line-interface/>
5. Nhiều nguồn khác trên Internet..