BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

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

A yellow and red logo

AI-generated content may be incorrect.

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC NGÀNH HỆ THỐNG THÔNG TIN

TRIỂN KHAI GIẢI PHÁP KẾT HỢP ORACLE RAC VÀ DATA GUARD CHO VIỆC ĐẢM BẢO TÍNH SẴN SÀNG CAO VÀ KHẢ NĂNG PHỤC HỒI SAU THẢM HỌA

CBHD :TS. Nguyễn Thị Hoa Huệ

Sinh viên : Nguyễn Hoàng Dũng

Mã số sinh viên : 2021600222

Hà Nội – Năm 2025

# **MỤC LỤC**

[MỤC LỤC i](#_Toc196299301)

[DANH MỤC CÁC CHỮ VIẾT TẮT iii](#_Toc196299302)

[DANH MỤC BẢNG BIỂU v](#_Toc196299303)

[DANH MỤC CÁC HÌNH VẼ vi](#_Toc196299304)

[MỞ ĐẦU 1](#_Toc196299305)

[1. Lý do chọn đề tài 1](#_Toc196299306)

[2. Mục tiêu của đề tài 2](#_Toc196299307)

[3. Đối tượng và phạm vi nghiên cứu 3](#_Toc196299308)

[4. Kết quả dự kiến đạt được 4](#_Toc196299309)

[5. Bố cục của đề tài 4](#_Toc196299310)

[CHƯƠNG 1: TỔNG QUAN VỀ RỦI RO Dữ Liệu và Giải Pháp Sẵn Sàng Cao, Phục Hồi ĐẦY ĐỦ. 5](#_Toc196299311)

[1.1. Tầm quan trọng của dữ liệu và CSDL 5](#_Toc196299312)

[1.2. Thực trạng của CSDL Standalone trước rủi ro 6](#_Toc196299313)

[1.2.1. Vấn đề gặp phải của CSDL Standalone 7](#_Toc196299314)

[1.2.2. Thách thức của CSDL Standalone rủi ro 9](#_Toc196299315)

[1.2.3. Giá trị mang lại của giải pháp kết hợp Oracle RAC và DataGuard 10](#_Toc196299316)

[1.2.4. Đề xuất giải pháp 10](#_Toc196299317)

[1.3. Tổng kết chương 1 12](#_Toc196299318)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 14](#_Toc196299319)

[2.1. Tìm hiểu về Oracle RAC 14](#_Toc196299320)

[2.1.1. Database Cluster 14](#_Toc196299321)

[2.1.2. Khái niệm, kiến trúc của Oracle RAC 17](#_Toc196299322)

[2.1.3. Mô hình mạng trong Oracle RAC 20](#_Toc196299323)

[2.2. Tìm hiểu về Data Guard 22](#_Toc196299324)

[2.2.1. Sao lưu với công cụ Recovery Manager 22](#_Toc196299325)

[2.2.2. Khái niệm, kiến trúc của Data Guard 23](#_Toc196299326)

[2.2.3. Loại hình đồng bộ 25](#_Toc196299327)

[2.2.4. Cơ chế tương tác giữa các thành phần 29](#_Toc196299328)

[2.2.5. Oracle Data Guard Broker 33](#_Toc196299329)

[2.3. Tổng kết chương 2 36](#_Toc196299330)

[CHƯƠNG 3: TRIỂN KHAI MÔ HÌNH VÀ ĐÁNH GIÁ HIỆU QUẢ 37](#_Toc196299331)

[3.1. Thông tin cấu hình máy chủ ảo cài đặt mô hình 37](#_Toc196299332)

[3.1.1. Oracle RAC 37](#_Toc196299333)

[3.1.2. Data Guard 38](#_Toc196299334)

[3.2. Triển khai giải pháp Oracle RAC 39](#_Toc196299335)

[3.2.1. Kiến trúc tổng quan 39](#_Toc196299336)

[3.2.2. Cấu hình mạng. 40](#_Toc196299337)

[3.2.3. Chuẩn bị môi trường 42](#_Toc196299338)

[3.2.4. Format đĩa, cấu hình ASM 46](#_Toc196299339)

[3.2.5. Cài đặt Oracle Grid Infrastructure 49](#_Toc196299340)

[3.2.6. Cài đặt Database Instance 51](#_Toc196299341)

[3.3. Triển khai Data Guard kết hợp với Oracle RAC 54](#_Toc196299342)

[3.3.1. Kiến trúc tổng quan 54](#_Toc196299343)

[3.3.2. Môi trường Oracle Net và định danh CSDL 54](#_Toc196299344)

[3.3.3. Cấu hình tham số chung cho hệ thống chính 57](#_Toc196299345)

[3.3.4. Tạo hệ thống dự phòng dựa trên RMAN DUPLICATE 59](#_Toc196299346)

[3.3.5. Cấu hình môi trường Data Guard 60](#_Toc196299347)

[3.3.6. Xác định chế độ bảo vệ trong Data Guard 62](#_Toc196299348)

[3.4. Tổng kết chương 3 63](#_Toc196299349)

[KẾT LUẬN 64](#_Toc196299350)

[Kết quả đóng góp 64](#_Toc196299351)

[Hạn chế 68](#_Toc196299352)

[Hướng phát triển 69](#_Toc196299353)

[TÀI LIỆU THAM KHẢO 70](#_Toc196299354)

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| Chữ viết tắt | Chữ đầy đủ | Diễn giải |
| ARL | Archived Redo Log | Tệp tin được lưu cất của dữ liệu đồng bộ |
| ASM | Automatic Storage Management | Tính năng quản lý đĩa, vùng nhớ tự động thay vì File Systems trong Oracle |
| CSDL | Cơ sở dữ liệu | Hệ thống lưu trữ dữ liệu. Trong này là CSDL dạng quan hệ |
| DBWR | Database Writer | Tiến trình ghi dữ liệu từ bộ nhớ xuống đĩa |
| DGB | Data Guard Broker | Thành phần quản lý môi trường Data Guard |
| DGMGRL | Data Guard Command Line  Interface | Công cụ giao diện dòng lệnh thao tác quản trị môi trường Data Guard |
| DML | Data Manipulation | Ngôn ngữ thao tác dữ liệu  trên CSDL như INSERT, UPDATE |
| DMON | Data Guard Monitor | Tiến trình trong mô hình Broker thực hiện theo dõi tình trạng môi trường Data Guard |
| FAL | Fetch Archived Log | Tính năng giúp CSDL dự phòng chủ động trong việc xử lý trễ/thiếu dữ liệu đồng bộ |
| LGWR | Log Writer | Tiến trình ghi dữ liệu đồng bộ từ bộ nhớ xuống đĩa |
| LNS | LogWriter Network Server | Tên chung cho các tiến trình thực hiện vận chuyển dữ liệu đồng bộ |
| RMAN | Recovery Manager | Công cụ sao lưu và phục hồi của Oracle |

|  |  |  |
| --- | --- | --- |
| LSP | Logical Standby Process | Tiến trình thực hiện áp dụng thay đổi của CSDL dự phòng dạng lô-gic |
| MAA | Maximum Availability Architecture | Khung kiến trúc giải pháp sẵn sàng cao của Oracle |
| MRP | Managed Recovery Process | Tiến trình thực hiện áp dụng thay đổi của CSDL dự phòng dạng vật lý |
| MTTR | Mean Time To Recovery | Chỉ số thời gian trung bình phục hồi giữa các lần gặp sự cố |
| RPO | Recovery Point Object | Chỉ số mục tiêu về lượng dữ liệu phục hồi |
| RTO | Recovery Time Object | Chỉ số mục tiêu về thời gian phục hồi |
| ODG | Oracle Data Guard | Giải pháp khôi phục sau thảm họa của Oracle |
| OLTP | Online Transaction Processing | Hệ thống dùng để chuyên xử lý giao dịch |
| OLAP | Online Analytic Processing | Hệ thống dùng để chuyên truy vấn, phân tích |
| PMON | Process Monitor | Một trong sáu tiến trình quan trọng giúp Instance hoạt động của CSDL Oracle |
| RAC | Real Application Cluster | Kiến trúc Oracle Database, trong đó một CSDL có thể được sử dụng bởi nhiều Instance (bản thể, multi-  Instance) |
| RFS | Remote File Server | Tiến trình thực hiện nhận dữ liệu đồng bộ của CSDL dự phòng |

# DANH MỤC BẢNG BIỂU

Bảng 1.1: Bốn kiến trúc trong giải pháp Oracle MAA ……………............. 13

Bảng 2.1: Cấu hình đối số phương thức truyền/xác nhận.............................. 28

Bảng 2.2: Tham số cấu hình cho cơ chế Fetch Archive Log………………..34

Bảng 2.3: So sánh việc sử dụng Broker vào hệ thống...................................... 36

Bảng 3.1: Cấu hình máy chủ ảo cài đặt Oracle RAC............................................ 38

Bảng 3.2: Cấu hình máy chủ ảo cài đặt Oracle Data Guard................................. 39

# DANH MỤC CÁC HÌNH VẼ

Hình 1.1: Minh họa khái niệm chỉ số RPO và RTO....................................... 8

Hình 1.2: Giải pháp backup dữ liệu bằng RMAN.......................................... 9

Hình 1.3: Mô hình áp dụng tính sẵn sàng cao.............................................. 10

Hình 1.4: Khung tham chiếu giải pháp Oracle MAA................................... 11

Hình 2.1: Mô hình Cluster………….………………………………........... 15

Hình 2.2: Mô hình Shared-Nothing.............................................................. 16

Hình 2.3: Mô hình Shared-Disk.................................................................... 17

Hình 2.4: Oracle Database với kiến trúc Oracle RAC.................................. 19

Hình 2.5 Kiến trúc truyền thống và kiến trúc ASM……………………….. 20

Hình 2.6: Phương pháp Round Robin........................................................... 22

Hình 2.7: Mô hình mạng trong Oracle RAC................................................. 23

Hình 2.8: Kết hợp RMAN, Oracle Secure Backup và sao lưu bằng lệnh hệ thống.............................................................................................................. 24

Hình 2.9: Kiến trúc tổng quan giải pháp Oracle Data Guard......................... 25

Hình 2.10: Tính năng Far Sync trong giải pháp Oracle Data Guard............... 29

Hình 2.11: Luồng hoạt động Data Guard với chế độ Ưu tiên bảo vệ............. 31

Hình 2.12: Luồng hoạt động Data Guard với chế độ Ưu tiên hiệu năng........ 32

Hình 2.13: Minh họa cơ chế xử lý thiếu trong việc truyền thông tin.............. 33

Hình 2.14: Kiến trúc Data Guard với tính năng Data Guard Broker ............. 35

Hình 2.15: Mối quan hệ giữa các thành phần trong mô hình Broker.............. 36

Hình 3.1: Kiến trúc tổng quan thực nghiệm giải pháp Oracle RAC ………...40

Hình 3.2: Kiến trúc tổng quan thực nghiệm giải pháp Data Guard kết hợp Oracle RAC.................................................................................................... 54

Hình 3.3: Minh họa phân cấp tên trong hệ thống CSDL………………......... 55

Hình 23: Thông tin các tài nguyên trong Oracle RAC................................... 65

Hình 24: Các tiến trình thuộc hai CSDL trong Data Guard........................... 66

Hình 25: Thông tin được cung cấp bởi Broker............................................... 67

LỜI CẢM ƠN

Để hoàn thành được đồ án tốt nghiệp này, em xin bày tỏ lòng biết ơn sâu sắc đến tất cả những thầy cô đã giúp đỡ và ủng hộ em trong suốt quá trình học tập và nghiên cứu.

Trước tiên, em xin gửi lời cảm ơn chân thành đến Trường Đại học Công Nghiệp Hà Nội, nơi đã cung cấp cho em một môi trường học tập tuyệt vời và những kiến thức quý báu trong suốt bốn năm qua. Em đặc biệt biết ơn cô Nguyễn Thị Hoa Huệ, người đã tận tình hướng dẫn, chia sẻ kinh nghiệm và hỗ trợ em trong quá trình thực hiện đồ án tốt nghiệp. Những lời khuyên, sự chỉ dẫn và kiến thức của cô đã giúp em vượt qua nhiều khó khăn và hoàn thành đồ án này.

Bên cạnh đó, em xin cảm ơn gia đình - nguồn động viên lớn lao, luôn bên cạnh, ủng hộ và khuyến khích em vượt qua mọi thử thách trong học tập. Em cũng không quên gửi lời cảm ơn đến tất cả bạn bè, những người đã đồng hành cùng em trong suốt quãng thời gian học tập tại trường, chia sẻ kiến thức và kinh nghiệm quý báu để giúp em hoàn thành tốt đồ án.

Cuối cùng, em xin chân thành cảm ơn tất cả những thầy cô và bạn bè đã đóng góp ý kiến, hỗ trợ em trong suốt quá trình làm đồ án. Những đóng góp của mọi người là động lực để em tiếp tục phấn đấu, học hỏi và phát triển trên con đường sự nghiệp sau này.

Em xin kính chúc quý Thầy Cô luôn dồi dào sức khỏe, hạnh phúc và thành công trong sự nghiệp giáo dục.

Em xin chân thành cảm ơn!

Sinh Viên

Nguyễn Hoàng Dũng

# MỞ ĐẦU

## Lý do chọn đề tài

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, việc đảm bảo tính sẵn sàng cao và khả năng phục hồi sau thảm họa cho các hệ thống CSDL ngày càng trở thành một yếu tố mang tính sống còn đối với các tổ chức và doanh nghiệp. Những sự cố mất dữ liệu hoặc gián đoạn dịch vụ không chỉ gây thiệt hại về mặt tài chính mà còn làm suy giảm uy tín và niềm tin từ phía khách hàng, đặc biệt đối với các ngành có yêu cầu khắt khe như tài chính, ngân hàng, thương mại điện tử và viễn thông.

Oracle RAC (Real Application Clusters) và Oracle Data Guard là hai giải pháp hàng đầu trong việc đảm bảo tính sẵn sàng cao và khả năng phục hồi sau thảm họa cho hệ thống CSDL. Việc kết hợp hai giải pháp này không chỉ mang lại độ tin cậy tuyệt đối cho các ứng dụng quan trọng mà còn tối ưu hóa hiệu suất và bảo vệ dữ liệu trong mọi tình huống khẩn cấp. Đây chính là nền tảng công nghệ cốt lõi giúp doanh nghiệp duy trì tính liên tục trong hoạt động và đảm bảo khả năng phục hồi nhanh chóng khi xảy ra sự cố.

Bên cạnh những lợi ích rõ rệt về mặt kỹ thuật, việc lựa chọn đề tài này còn xuất phát từ mong muốn tìm hiểu sâu hơn về cách thức triển khai thực tế của các mô hình đảm bảo tính sẵn sàng cao trong môi trường doanh nghiệp. Thay vì chỉ tập trung vào lý thuyết, đề tài hướng đến việc phân tích chi tiết quá trình kết hợp Oracle RAC và Data Guard, từ khâu thiết kế kiến trúc, cấu hình hệ thống cho đến việc kiểm thử khả năng chịu lỗi và phục hồi sau thảm họa. Đây sẽ là cơ sở quan trọng để xây dựng những quy trình triển khai bài bản, phù hợp với nhu cầu vận hành của các tổ chức có quy mô và yêu cầu khác nhau.

Ngoài ra, đề tài này còn mang ý nghĩa quan trọng trong việc nghiên cứu và áp dụng các công nghệ tiên tiến trong lĩnh vực quản trị cơ sở dữ liệu. Nó không chỉ cung cấp giải pháp thực tiễn cho các doanh nghiệp mà còn góp phần nâng cao kiến thức chuyên môn về triển khai và vận hành các hệ thống có tính sẵn sàng cao. Qua việc nghiên cứu và thực nghiệm, đề tài sẽ đóng góp vào việc xây dựng quy trình chuẩn, giúp các doanh nghiệp chủ động hơn trong việc phòng ngừa và ứng phó với các tình huống khẩn cấp.

Hơn nữa, việc triển khai giải pháp kết hợp Oracle RAC và Data Guard không chỉ mang lại lợi ích về mặt kỹ thuật mà còn giúp doanh nghiệp giảm thiểu chi phí do gián đoạn dịch vụ, bảo vệ tài sản dữ liệu và đảm bảo hoạt động kinh doanh không bị gián đoạn. Đây là một yêu cầu không thể thiếu trong thời đại kinh tế số hóa, khi mà sự ổn định và an toàn của hệ thống cơ sở dữ liệu trở thành yếu tố cốt lõi cho sự phát triển bền vững.

Tóm lại, với tầm quan trọng trong việc đảm bảo tính sẵn sàng cao và khả năng phục hồi sau thảm họa, đề tài “Triển khai giải pháp kết hợp Oracle RAC và Data Guard cho việc đảm bảo tính sẵn sàng cao và khả năng phục hồi sau thảm họa” không chỉ mang ý nghĩa khoa học mà còn có tính thực tiễn cao, đáp ứng nhu cầu cấp thiết của doanh nghiệp trong thời đại công nghệ hiện nay.

## Mục tiêu của đề tài

Đề tài: TRIỂN KHAI GIẢI PHÁP KẾT HỢP ORACLE RAC VÀ DATA GUARD CHO VIỆC ĐẢM BẢO TÍNH SẴN SÀNG CAO VÀ KHẢ NĂNG PHỤC HỒI SAU THẢM HỌA đáp ứng được những mục tiêu:

Nghiên cứu các thành phần và cơ chế hoạt động của Oracle RAC và DataGuard: Tìm hiểu sâu về cấu trúc, chức năng cũng như cách hoạt động của Oracle Real Application Clusters (RAC) và Oracle Data Guard nhằm tạo nền tảng cho hệ thống cơ sở dữ liệu liên tục hoạt động trong mọi tình huống.

Nắm được quá trình kết hợp hai giải pháp: Xác định cách kết hợp Oracle RAC với Oracle Data Guard, trong đó Oracle RAC đảm nhiệm vai trò phân phối tải và cung cấp khả năng chịu lỗi thông qua hoạt động của nhiều nút, còn Oracle Data Guard đảm bảo đồng bộ hóa, sao lưu và khôi phục dữ liệu tự động khi xảy ra sự cố.

Phân tích các kịch bản phục hồi: Mô phỏng các tình huống thảm họa, từ đó xây dựng quy trình chuyển đổi dự phòng (Failover) và chuyển đổi trở lại (Switchback) một cách hiệu quả, đảm bảo hệ thống luôn duy trì hoạt động không gián đoạn.

Thử nghiệm hệ thống mẫu và đánh giá chất lượng: Xây dựng một hệ thống mẫu triển khai kết hợp Oracle RAC và Data Guard trên môi trường thử nghiệm, đánh giá hiệu quả của giải pháp qua các chỉ số về độ trễ, tính nhất quán dữ liệu và thời gian phục hồi, từ đó so sánh với các phương pháp truyền thống và đề xuất các hướng cải tiến.

## Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

* Các thành phần, công nghệ và kỹ thuật liên quan đến Oracle Real Application Clusters (RAC) và Oracle Data Guard, bao gồm cơ chế phân phối tải, cân bằng, đồng bộ hóa dữ liệu và các quy trình chuyển đổi dự phòng (Failover) cũng như chuyển đổi trở lại (Switchback).

Phạm vi nghiên cứu:

* Phạm vi hệ thống: Đề tài giới hạn trong việc triển khai và thử nghiệm các giải pháp kết hợp Oracle RAC và Oracle Data Guard trên một hệ thống cơ sở dữ liệu thử nghiệm. Hệ thống này sẽ được cấu hình để đáp ứng các yêu cầu về tính sẵn sàng cao và khả năng phục hồi sau thảm họa.
* Phạm vi kỹ thuật:
* Tập trung vào việc cài đặt, cấu hình và tích hợp Oracle RAC với Oracle Data Guard nhằm tạo ra một môi trường cơ sở dữ liệu liên tục hoạt động, đảm bảo đồng bộ hóa dữ liệu giữa các node và các instance dự phòng.
* Thực hiện các kịch bản kiểm thử nhằm đánh giá hiệu quả của giải pháp, bao gồm kiểm tra thời gian chuyển đổi dự phòng, thời gian phục hồi (Recovery Time Objective - RTO) và điểm phục hồi (Recovery Point Objective - RPO) của hệ thống.
* Phạm vi ứng dụng:
* Đề tài tập trung vào việc xây dựng và đánh giá một ứng dụng mẫu nhằm minh họa khả năng đảm bảo tính sẵn sàng cao và phục hồi sau thảm họa của hệ thống.
* Ngoài ra, nghiên cứu cũng xem xét việc so sánh hiệu quả của giải pháp kết hợp này với các phương pháp truyền thống, từ đó đưa ra các khuyến nghị và hướng phát triển cho các ứng dụng thực tiễn trong môi trường doanh nghiệp.

## Kết quả dự kiến đạt được

Nghiên cứu lý thuyết:

* Nghiên cứu nguyên lý hoạt động, cơ chế và kiến trúc của Oracle RAC Oracle Data Guard, từ đó hiểu rõ vai trò của từng thành phần trong việc đảm bảo tính sẵn sàng cao và khả năng phục hồi sau thảm họa.
* Nắm rõ quy trình cấu hình, đồng bộ hóa dữ liệu giữa các node trong RAC và cơ chế chuyển đổi dự phòng (Failover/Switchback) của Data Guard.
* Đánh giá các phương pháp triển khai, so sánh ưu nhược điểm của các mô hình tích hợp RAC và Data Guard so với các giải pháp truyền thống.

Ứng dụng thực tiễn:

* Phát triển thành công một hệ thống thử nghiệm tích hợp Oracle RAC và Data Guard trên môi trường ảo hóa hoặc hạ tầng thực tế, đảm bảo hệ thống cơ sở dữ liệu hoạt động liên tục và không gián đoạn.
* Viết kịch bản mô phỏng thảm họa (ví dụ: mất kết nối, lỗi phần cứng) để kiểm tra quá trình chuyển đổi dự phòng và khôi phục hệ thống, từ đó xác nhận hiệu quả của giải pháp.

## Bố cục của đề tài

Chương 1: Tổng quan về nhu cầu bảo vệ dữ liệu và giải pháp sẵn sàng cao và phục hồi nhanh

Chương 2: Cơ sở lý thuyết

Chương 3: Triển khai mô hình và đánh giá hiệu quả

# TỔNG QUAN VẤN ĐỀ và Giải Pháp Sẵn Sàng Cao, Phục Hồi ĐẦY ĐỦ.

## Tầm quan trọng của dữ liệu và CSDL

Trong thế giới kinh doanh hiện đại, chúng ta đều hiểu rằng cơ sở dữ liệu không chỉ đơn thuần là nơi lưu trữ thông tin, mà chính là tài sản cốt lõi của một doanh nghiệp. Đối với các tập đoàn lớn, dữ liệu không chỉ là những con số hay thông tin rời rạc mà là “mỏ dầu”, là bằng chứng của mọi quyết định chiến lược và hoạt động kinh doanh. Chính dữ liệu đã và đang tạo nên sức mạnh cạnh tranh, giúp doanh nghiệp nhận diện xu hướng thị trường, dự báo nhu cầu và tối ưu hóa hoạt động nội bộ. Do đó, việc xây dựng một hệ thống CSDL mạnh mẽ, ổn định và có khả năng phục hồi khi gặp sự cố trở thành yêu cầu không thể thiếu đối với các doanh nghiệp quy mô lớn.

Trong môi trường doanh nghiệp, nơi mà mọi giao dịch, mọi hoạt động sản xuất, tài chính và quản trị đều phụ thuộc vào việc truy xuất thông tin nhanh chóng và chính xác, bất kỳ gián đoạn nào trong hệ thống CSDL cũng có thể gây ra hậu quả nghiêm trọng. Một sự cố mất mát dữ liệu hoặc thời gian gián đoạn kéo dài không chỉ ảnh hưởng đến doanh thu mà còn làm tổn hại đến uy tín và niềm tin của khách hàng, đối tác kinh doanh. Điều này càng trở nên đáng lo ngại trong bối cảnh các doanh nghiệp ngày càng phụ thuộc vào công nghệ số để điều hành và quản lý hoạt động hàng ngày.

Để đối phó với những thách thức trên, các doanh nghiệp lớn không chỉ cần một cơ sở dữ liệu hiệu quả mà còn phải đảm bảo rằng hệ thống này luôn hoạt động trơn tru, an toàn và có khả năng phục hồi đầy đủ sau bất kỳ sự cố nào – từ lỗi phần cứng, sự cố mạng cho đến các thảm họa tự nhiên. Chính vì lý do đó, đầu tư vào các giải pháp đảm bảo tính sẵn sàng cao và phục hồi sau thảm họa cho cơ sở dữ liệu trở thành một ưu tiên hàng đầu.

Oracle, với kinh nghiệm lâu năm và uy tín trên thị trường, đã cho ra đời hai công nghệ tiên tiến nhằm bảo vệ, thay thế cho các hệ thống CSDL Standalone – là một cơ sở dữ liệu được cài đặt và vận hành trên một máy chủ độc lập mà không có bất kỳ cơ chế phân tán hay chia sẻ tài nguyên nào giữa các máy chủ khác. Đó chính là Oracle Real Application Clusters (RAC) [1] và Oracle Data Guard [2].

## Thực trạng của CSDL Standalone trước rủi ro

Trong thời đại số, sự liên tục của hoạt động doanh nghiệp không chỉ là một lợi thế cạnh tranh mà còn là yếu tố sống còn. Khi mỗi giây gián đoạn đều có thể dẫn đến tổn thất tài chính và mất mát uy tín, việc đảm bảo hệ thống luôn sẵn sàng vận hành, vượt qua mọi sự cố và khối lượng giao dịch tăng đột biến là cấp thiết. Ngoài ra, một hệ thống được thiết kế với khả năng cân bằng tải tối ưu không chỉ giúp phân chia công việc đồng đều giữa các máy chủ mà còn tạo ra một lớp phòng vệ vững chắc trước những rủi ro đột ngột. Nhưng hiện nay, nhiều hệ thống chưa đáp ứng được tính sẵn sàng, hoạt động liên tục cũng như khả năng cân bằng tải.

Song song đó, giải pháp sao lưu và khôi phục dữ liệu cũng rất quan trọng. Nó phải được thực hiện nhanh chóng và nhất quán đảm bảo rằng, trong tình huống khẩn cấp, doanh nghiệp có thể ngay lập tức phục hồi thông tin và tiếp tục hoạt động mà không để xảy ra mất mát nghiêm trọng. Công nghệ thông tin ngày càng phát triển, dữ liệu được tạo ra không ngừng, một số công cụ sao lưu không đảm bảo về thời gian khôi phục và tính nhất quán của dữ liệu trong bản sao lưu.

Hai khái niệm “tính sẵn sàng cao” và “khả năng phục hồi sau thảm họa” đóng vai trò then chốt trong chiến lược bảo vệ hệ thống. “Tính sẵn sàng cao” đề cập đến khả năng của hệ thống hoạt động liên tục mà không bị gián đoạn, ngay cả khi gặp sự cố hoặc lượng tải tăng đột biến, nhờ đó đảm bảo cung cấp dịch vụ ổn định cho người dùng. Trong khi đó, “khả năng phục hồi sau thảm họa” thể hiện năng lực nhanh chóng khôi phục hoạt động của hệ thống sau những sự cố nghiêm trọng, giúp đảm bảo dữ liệu được phục hồi đầy đủ và chính xác, từ đó giảm thiểu thiệt hại về tài chính cũng như uy tín doanh nghiệp.

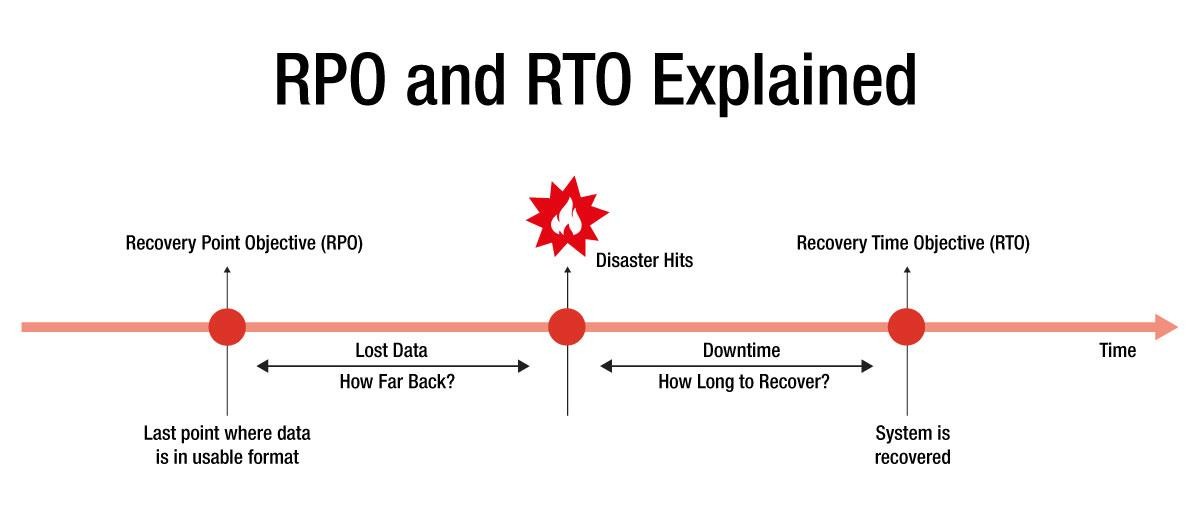
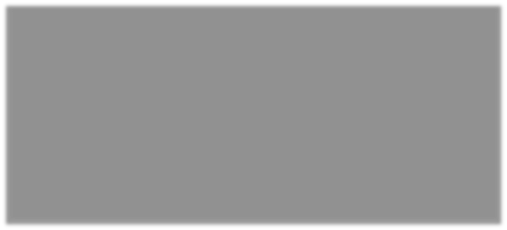
Do đó, doanh nghiệp cần ứng dụng các giải pháp khác để khắc phục những hạn chế trên, đảm bảo khả năng vận hành ổn định, an toàn.

### Vấn đề gặp phải của CSDL Standalone

#### RTO, RPO

Trong quy trình sao lưu và khôi phục dữ liệu, Recovery Time Objective và Recovery Point Objective là hai chỉ số quan trọng, giúp doanh nghiệp xác định được mức độ khôi phục dữ liệu và thời gian khôi phục dữ liệu khi hệ thống gặp sự cố.

* Recovery Time Objective (RTO): khoảng thời gian hệ thống được khôi phục, kể từ thời điểm xảy ra sự cố. Thể hiện tốc độ khôi phục dữ liệu.
* Recovery Point Objective (RPO): khoảng thời gian tối đa hệ thống chấp nhận mất dữ liệu, kể từ thời điểm sao lưu cuối cùng cho tới lúc xảy ra sự cố. Thể hiện tần suất trong việc sao lưu dữ liệu.

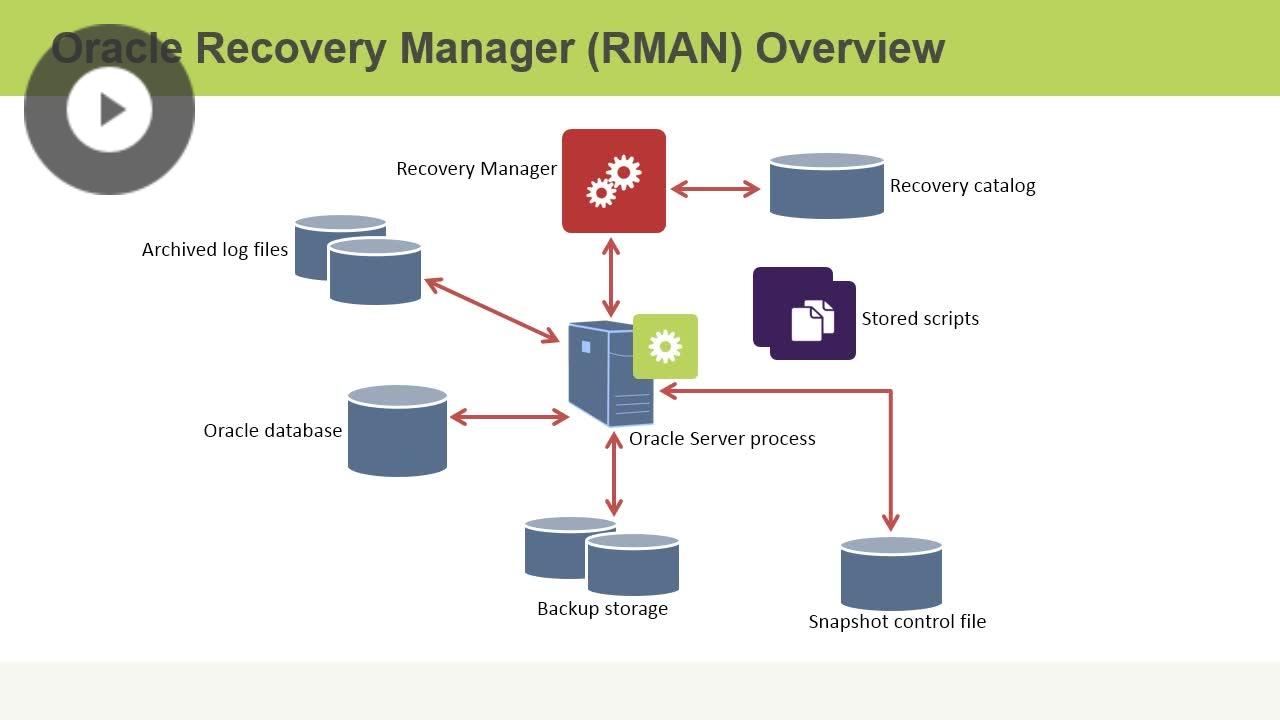


Hình 1.1: Minh họa khái niệm chỉ số RPO và RTO

Hai chỉ số RTO và RPO có mối liên hệ mật thiết với nhau, RPO càng nhỏ thì RTO càng nhỏ (dữ liệu được sao lưu liên tục, dẫn đến thời gian khôi phục nhanh), nhưng chi phí càng cao do yêu cầu hiệu năng và thiết bị lưu trữ lớn, đòi hỏi người quản trị viên cần có kế hoạch trong việc giám sát và thực hiện sao lưu liên tục.

#### Sao lưu, khôi phục

Oracle Database được sử dụng làm hệ quản trị cơ sở dữ liệu cho nhiều hệ thống lõi trong các doanh nghiệp vừa và lớn khác nhau. Để sao lưu dữ liệu, Oracle cung cấp công cụ Recovery Manager (RMAN), thiết lập sao lưu theo lịch cho CSDL.



Hình 1.2: Giải pháp backup dữ liệu bằng RMAN

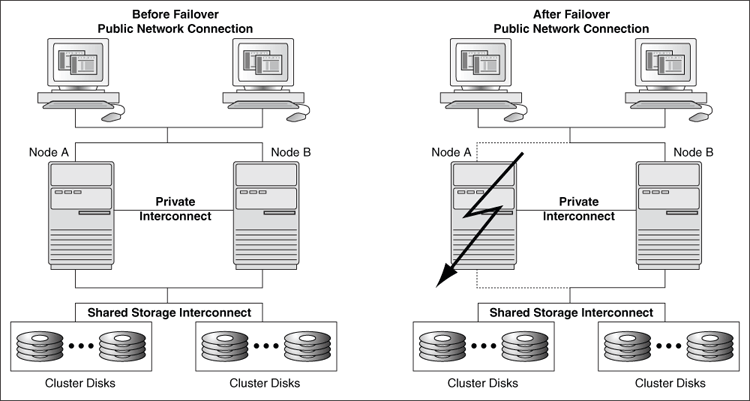
Với công cụ RMAN, đội ngũ quản trị có thể sao lưu từ toàn bộ cho đến một phần CSDL. Tuy nhiên, thời gian khôi phục của RMAN lên tới hàng giờ cho đến hàng ngày, chưa đáp ứng được việc có giá trị RTO nhỏ. Yu & cộng sự (2011) cho rằng, công cụ RMAN có nhược điểm như:

* Hạn chế về băng thông: RMAN gặp hạn chế khi trong quá trình sao lưu lượng dữ liệu lớn tới thiết bị lưu trữ không cùng địa điểm với CSDL. Quá trình sao lưu cần rất nhiều tài nguyên cũng như độ tin cậy của hệ thống mạng, gây rủi ro về mặt hiệu suất và việc gián đoạn trong quá trình sao lưu.
* Hạn chế về thời gian khôi phục: Trong trường hợp toàn bộ hệ thống trong cùng một trung tâm dữ liệu gặp sự cố, việc khôi phục dữ liệu từ bản sao lưu bằng công cụ RMAN tiêu tốn về mặt thời gian, và lượng dữ liệu mất mát là rất lớn.

#### Tính sẵn sàng cao

Ngoài ra, nhiều doanh nghiệp cũng đang chạy Oracle Database nhưng không thiết kế để đảm bảo tính sẵn sàng cao và cân bằng tải. Doanh nghiệp thường phải vận hành trên một hoặc vài máy chủ đơn lẻ. Điều này cũng dẫn đến một số hạn chế khá nghiêm trọng như:

* Nguy cơ gián đoạn: Nếu một máy chủ gặp sự cố hoặc cần bảo trì, toàn bộ hệ thống có thể bị ảnh hưởng, kéo theo thời gian gián đoạn kéo dài và tác động tiêu cực đến hoạt động kinh doanh.
* Phân phối tải hạn chế: Việc xử lý khối lượng giao dịch lớn sẽ tập trung vào một số máy chủ, dễ gây quá tải và giảm hiệu suất xử lý. Điều này không chỉ làm chậm tốc độ phản hồi mà còn tăng khả năng xảy ra lỗi.
* Khả năng phục hồi kém: Thiếu cơ chế tự động chuyển giao nhiệm vụ giữa các máy chủ khiến thời gian khôi phục hệ thống (RTO) và mức mất dữ liệu tối đa (RPO) khó đáp ứng được các yêu cầu khắt khe. Điều này đồng nghĩa với việc khi gặp sự cố, việc khôi phục toàn bộ hệ thống sẽ mất nhiều thời gian và không đảm bảo độ tin cậy cần thiết.



Hình 1.3: Mô hình áp dụng tính sẵn sàng cao

### Thách thức của CSDL Standalone rủi ro

Khi hệ thống gặp vấn đề, như sụp đổ hoặc gián đoạn, gây ảnh hưởng đến hoạt động kinh doanh hàng ngày như mất mát dữ liệu, không thể truy cập thông tin quan trọng để đưa ra quyết định kịp thời, gián đoạn trong dịch vụ có thể dẫn đến mất lòng tin từ phía khách hàng, gây tổn hại về uy tín và doanh số của công ty.

Việc phục hồi hệ thống có thể đòi hỏi nhiều thời gian và chi phí, và nếu không có kế hoạch hồi phục hiệu quả có thể làm gia tăng rủi ro, ảnh hưởng đến sức mạnh cạnh tranh của công ty trong ngành.

Hơn hết, trong một môi trường kinh doanh đầy cạnh tranh như ngành tài chính hoặc ngân hàng, việc duy trì sự ổn định của hệ thống là một yếu tố quan trọng để bảo vệ uy tín và sự phát triển của doanh nghiệp.

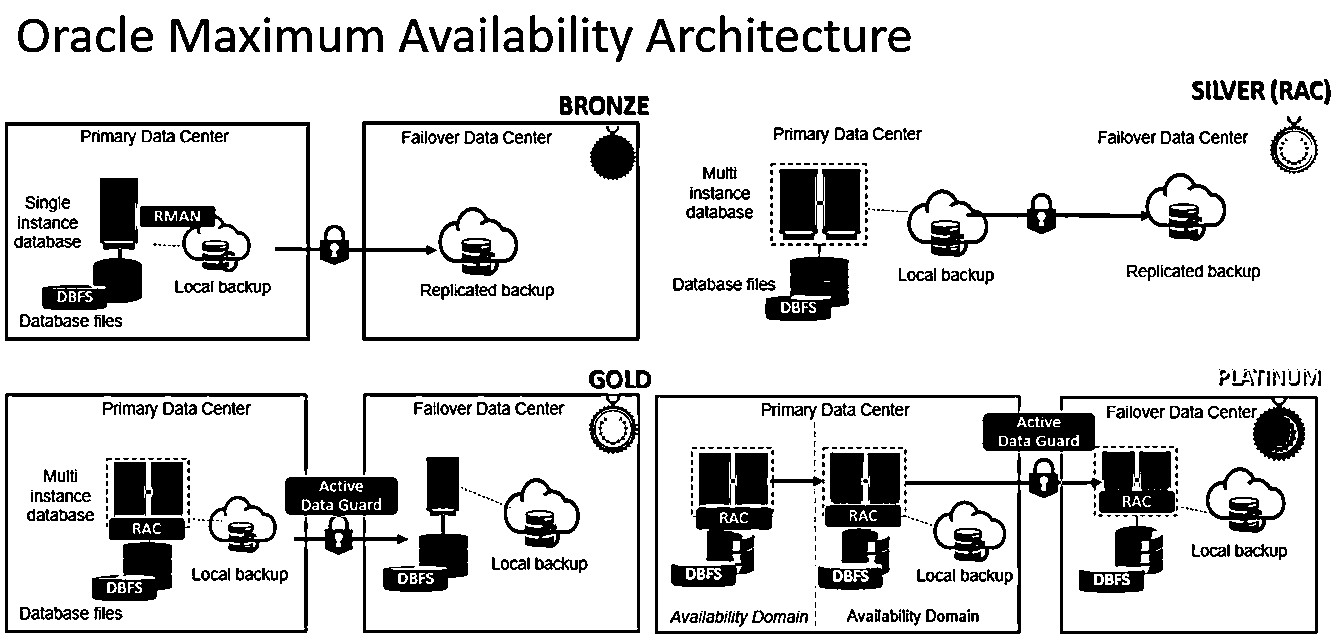
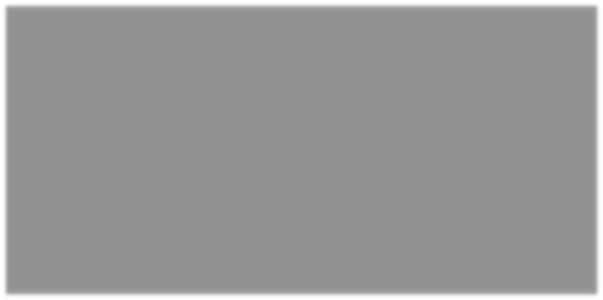
### Giá trị mang lại của giải pháp kết hợp Oracle RAC và DataGuard

Để giải quyết thách thức vấn đề tần suất sao lưu và thời gian khôi phục cơ sở dữ liệu, cần triển khai giải pháp hiệu quả có khả năng sao lưu liên tục, khôi phục dữ liệu tự động, tăng khả năng chịu tải, giảm thiểu thời gian gián đoạn và mất mát dữ liệu. Lợi ích mang lại có thể kể đến như:

* Khôi phục dữ liệu kịp thời: khôi phục dữ liệu nhanh chóng sau sự cố, giảm thời gian gián đoạn.
* Tiết kiệm chi phí và thời gian: giảm chi phí và thời gian phục hồi sau sự cố, loại bỏ việc tiêu tốn nguồn lực nhân sự để thực hiện các công việc sao lưu và khôi phục thủ công. Điều này giúp tổ chức tối ưu hóa hiệu suất và tài nguyên, tập trung vào các hoạt động kinh doanh chính.
* Dữ liệu được sao lưu liên tục: dữ liệu được sao lưu với tần suất cao, tự động và sát với thời gian thực của CSDL, đảm bảo tính toàn vẹn của dữ liệu.
* Tăng cường tính sẵn sàng và khả năng chịu lỗi: cho phép nhiều máy chủ cùng hoạt động trên một cơ sở dữ liệu duy nhất, đảm bảo rằng nếu một node gặp sự cố, các node khác vẫn tiếp tục vận hành, đảm bảo tính sẵn sàng.
* Linh hoạt trong bảo trì và nâng cấp: Việc bảo trì hoặc nâng cấp một node có thể được thực hiện mà không ảnh hưởng đến toàn bộ hệ thống, do các node khác vẫn tiếp tục hoạt động bình thường.
* Tối ưu hiệu suất và phân tải công việc: phân phối khối lượng công việc giữa các node, tăng cường hiệu suất xử lý và đảm bảo hệ thống hoạt động mượt mà ngay cả khi tải công việc tăng cao

### Đề xuất giải pháp

Giải pháp đảm bảo tính liên tục của Oracle - *Oracle Maximum Availability Architecture (MAA)[3]* được chia ra làm bốn mô hình kiến trúc, phân chia theo hạng với mức độ bảo vệ tăng dần từ trái sang phải: Bronze, Silver, Gold và Plantinum. Mức độ bảo vệ được thể hiện qua các tính năng như bảo vệ dữ liệu, tính sẵn sàng cao, khôi phục sau thảm họa.



*Hình 1.4: Khung tham chiếu giải pháp Oracle MAA*

Hai chỉ số RTO và RPO được cải thiện thông qua hạng bậc, lượng dữ liệu khôi phục và thời gian khôi phục được đánh giá thông qua ba trường hợp chính: lỗi máy chủ (Server), lỗi thiết bị lưu trữ (Storage) và lỗi toàn bộ trung tâm dữ liệu (Data Center).

|  |  |  |
| --- | --- | --- |
| Hạng | Công nghệ | RTO/RPO  (Server, Storage, Data Center) |
| Bronze | CSDL dạng Single sdffs Instance, công nghệ sdsdf sao lưu bằng RMAN | RPO: thấp nhất là 0 (phút), hh nhiều nhất là x (tiếng)  RTO: thấp nhất là x (phút), nhiều hhhh nhất là x (ngày) |
| Silver | Bronze + CSDL dạng Mh Multi – Instance với vfd công nghệ Real  Application Cluster fddf (RAC) | RPO: thấp nhất là 0 (phút), nhiều u nhất là x (tiếng)  RTO: thấp nhất là x (phút), nhiều u nhất là x (ngày) |
| Gold | Silver + CSDL dự uuu phòng với công nghệ uuu Oracle Data Guard | RPO: thấp nhất là 0 (phút), nhiều u nhất là x (phút)  RTO: thấp nhất là x (phút), nhiều n nhất là x (phút) |
| Plantinum | Gold + một số tính uuuu năng khác như Golden u Gate, Sharding | RPO: thấp nhất là 0, nhiều nhất là u x (phút)  RTO: thấp nhất là x (phút), nhiều unhất là x (phút) |

*Bảng 1.1: Bốn kiến trúc trong giải pháp Oracle MAA*

Vì vậy, để giải quyết vấn đề đã nêu và sau thời gian tìm hiểu thì kiến trúc Gold là kiến trúc được đề xuất để sử dụng. Ứng dụng được tính năng Oracle RAC và Data Guard, hướng tới việc giảm chỉ số RTO và RPO.

Giải pháp Oracle RAC có các ưu điểm như sau:

* Tính sẵn sàng cao.
* Khả năng cân bằng tải.
* Quản lý bảo trì và nâng cấp không bị gián đoạn.

Giải pháp Data Guard có các ưu điểm như sau:

* Sao lưu dữ liệu theo thời gian thực.
* Khôi phục kịp thời.
* Quản lý tự động.
* Giảm tải cho hệ thống chính.

Thông qua các ưu điểm nổi trội của Oracle RAC và Data Guard đem lại, đây là giải pháp phù hợp các CSDL quan trọng hiện nay. Việc triển khai sẽ giúp hướng tới mục tiêu giảm chỉ số RTO và RPO, đảm bảo tính liên tục trong kinh doanh của doanh nghiệp

## Tổng kết chương 1

Chương đầu tiên này đã đi sâu vào việc nhận thức về tầm quan trọng của dữ liệu và cơ sở dữ liệu trong bối cảnh công nghệ số hiện nay, nơi mà việc duy trì tính liên tục của hoạt động doanh nghiệp trở nên sống còn. Các khái niệm cốt lõi như tính sẵn sàng cao, khả năng phục hồi sau thảm họa, cũng như tầm quan trọng của sao lưu và khôi phục dữ liệu đã được làm rõ, nhằm chỉ ra những yêu cầu cấp thiết của hệ thống hiện đại.

Tiếp đó, chương đã giới thiệu giải pháp kết hợp giữa Oracle RAC và Data Guard như một hướng đi toàn diện để giải quyết các thách thức về hiệu suất, an toàn và khả năng phục hồi sau thảm họa. Giải pháp này không chỉ đảm bảo tính sẵn sàng cao thông qua việc phân phối tải và duy trì hoạt động liên tục ngay cả khi có sự cố, mà còn cung cấp cơ chế sao lưu và khôi phục dữ liệu nhanh chóng, giúp doanh nghiệp giảm thiểu tối đa thời gian gián đoạn và nguy cơ mất mát dữ liệu.

Như vậy, chương 1 đã thiết lập nền tảng lý thuyết và thực tiễn, mở đường cho những nội dung tiếp theo đó là đi sâu hơn về kiến trúc, thành phần của giải pháp Oracle RAC kết hợp Data Guard.

# CƠ SỞ LÝ THUYẾT

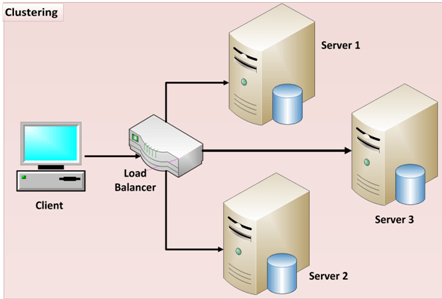
## Tìm hiểu về Oracle RAC

### Database Cluster

#### Khái niệm Clustering và Server Cluster

Clustering là quá trình phân nhóm các thiết bị (nodes) trong mạng thành các nhóm con (clusters) dựa trên một số tiêu chí nhất định. Mỗi cluster được xác định bởi một node chủ đạo (cluster head) và các node khác trong cluster giao tiếp trực tiếp với cluster head thay vì trực tiếp với các node khác trong mạng.

Server Cluster là một hệ thống gồm nhiều máy chủ (Server) hoạt động cùng nhau và chia sẻ tài nguyên để tăng tính sẵn sàng, độ tin cậy và hiệu suất của hệ thống. Trong một Server Cluster, các máy chủ (các node) được liên kết với nhau và hoạt động như một hệ thống đơn lẻ, với mục đích cung cấp dịch vụ cho người dùng hoặc ứng dụng mà không bị gián đoạn.



*Hình 2.1: Mô hình Cluster*

Có nhiều loại Server Cluster được sử dụng trong thực tế, tùy thuộc vào mục đích và yêu cầu cụ thể. Database Cluster là một loại mô hình Cluster trong Server Cluster. Loại Cluster này được sử dụng để tăng cường tính sẵn sàng và hiệu suất của các hệ thống cơ sở dữ liệu bằng cách sử dụng nhiều Server để lưu trữ và truy xuất dữ liệu. Database Cluster được sử dụng rộng rãi trong các doanh nghiệp lớn, các ứng dụng web, điện toán đám mây và các hệ thống lưu trữ dữ liệu.

#### Các mô hình Database Cluster

Shared-Nothing Architecture:

A diagram of a computer network

AI-generated content may be incorrect.

*Hình 2.2: Mô hình Shared-Nothing*

Đây là kiến trúc đã xuất hiện từ lâu nhưng gần đây đã trở lại, có trong các công nghệ lưu trữ dữ liệu. Đặc biệt như Data Warehousing và Big Data. Ở kiến trúc này, dữ liệu được phân chia trên một tập hợp các máy. Mỗi máy sẽ có quyền truy cập và chịu trách nhiệm hoàn toàn đối với phần dữ liệu mà nó lưu trữ và không chia sẻ với máy khác. Cần phải phân chia dữ liệu trên các máy hợp lý để đảm bảo hiệu suất.

Kiến trúc này giúp tăng khả năng mở rộng lên rất cao bằng cách thêm nhiều node mà không gây ra xung đột về truy cập tài nguyên, mỗi node hoạt động độc lập với CPU, RAM, Disk riêng biệt. Ngoài ra còn có thể tận dụng tài nguyên trên các node để giúp tăng hiệu suất đọc-ghi.

Với kiến trúc Shared-Nothing, có thể dễ dàng triển khai cho kiến trúc Distributed Database hoặc Microservice,…. Giúp giảm chi phí hạ tầng vì có thể sử dụng nhiều server nhỏ thay vì một máy chủ lớn làm tốn nhiều chi phí. Ngoài ra còn phù hợp với các hệ thống OLTP như mạng xã hội, game online toàn cầu,... Nơi mà có nhiều giao dịch nhỏ, diễn ra liên tục vì nó đảm bảo được hiệu suất khi số lượng người dùng hoặc số lượng giao dịch tăng lên nhiều.

Shared-Disk Architecture:

A diagram of a computer network

AI-generated content may be incorrect.

*Hình 2.3: Mô hình Shared-Disk*

Ngược lại với kiến trúc Shared-Nothing, kiến trúc Shared-Disk cho phép tất cả dữ liệu có thể được truy cập từ tất cả các node trong cụm. Đặc biệt với việc khi cần truy cập dữ liệu một cách đồng thời từ nhiều phía thì kiến trúc này có thể đáp ứng tốt.

Khi nhiều nút cùng ghi vào một bản ghi, hệ thống phải sử dụng lock trên đĩa hoặc thông báo ý định lock đến các nút khác để đảm bảo tính nhất quán. Ngoài ra còn phải quản lý các lock. Việc này có thể làm giảm hiệu suất tổng thể khi số lượng nút tăng lên, mức độ cạnh tranh tài nguyên tăng. Vậy nên kiến trúc này sẽ bị hạn chế khả năng mở rộng. Thường sẽ chỉ có 2 hoặc 3 node được triển khai trong kiến trúc này.

Kiến trúc này sẽ phù hợp với các hệ thống OLAP hay các hệ thống cần truy cập dữu liệu đồng thời từ nhiều node vì có thể cải thiện hiệu suất việc truy vấn dữ liệu. Không cần yêu cầu truyền dữ liệu từ các máy nếu truy vấn đó cần dữ liệu từ nhiều node. Vì việc truyền lượng dữ liệu lớn từ các máy sẽ làm giảm hiệu suất tổng thể, đặc biệt là với các truy vấn có câu lệnh JOIN.

Kiến trúc Shared-Disk này cũng xuất hiện rất nhiều trong các giải pháp clustering của Oracle Database như là Oracle RAC (Oracle Real Application Cluster). Nơi mà việc chia sẽ dữ liệu được tối ưu qua các kỹ thuật phân phối khóa và cân bằng tải. Còn việc quản lý Disk được chịu trách nhiệm bởi công cụ đặc biệt đó là ASM (Automatic Storage Management).

### Khái niệm, kiến trúc của Oracle RAC

Khái niệm:

Trong các giải pháp đảm bảo tính sẵn sàng cao (High Availability – HA) của hệ thống Database, Oracle Real Application Clusters (Oracle RAC) là một công nghệ được đánh giá cao trong việc hỗ trợ nhiều máy chủ cùng truy cập và quản lý một Oracle Database duy nhất. Oracle RAC được xây dựng và tích hợp trên Oracle Database, gồm nhiều mô-đun chức năng như quản lý, giám sát và phân phối tải để đảm bảo hiệu suất và tính ổn định của hệ thống.

Oracle RAC duy trì sự ổn định này bằng cách cho phép nhiều máy chủ chia sẻ cùng một tập dữ liệu, hoạt động đồng bộ với nhau như một hệ thống thống nhất. Các máy chủ trong cụm có thể đặt tại cùng một trung tâm dữ liệu để tối ưu hóa hiệu suất và giảm thiểu độ trễ trong quá trình xử lý. Khi một node trong cụm gặp lỗi, Oracle RAC sẽ tự động phân phối lại tải công việc (failover) sang các nút còn lại, đảm bảo quá trình vận hành không bị gián đoạn.

Kiến trúc:

Trong kiến trúc Oracle RAC gồm nhiều node, mỗi node là 1 instance của Oracle Database chạy song song, cùng truy cập và xử lý một cơ sở dữ liệu dùng chung. Các instance này liên kết và giao tiếp qua môi trường mạng nội bộ do Oracle cung cấp, đảm bảo tính nhất quán dữ liệu và khả năng phục hồi cao trong trường hợp một hoặc nhiều node gặp sự cố. Các thành phần chính trong kiến trúc RAC được thiết kế để hỗ trợ đồng bộ hóa dữ liệu theo cơ chế Cache Fusion, quản lý tài nguyên và thực hiện cân bằng tải giữa các node.

Trong kiến trúc này, khi client (App, Web,..) yêu cầu kết nối thông qua SCAN IP được cấu hình sẵn. SCAN Listener sẽ chuyển yêu cầu đến Local listener (Virtual IP) sẽ đại diện cho một node và chọn node có ít tải nhất. Tiếp theo yêu cầu sẽ được chuyển xuốnng cho instance của node đó để xử lý yêu cầu. Khi dữ liệu được thay đổi, sẽ lưu trên memory dưới dạng block và chưa cập nhật luôn xuống disk. Lúc đó cơ chế Cache Fusion sẽ hoạt động để cho phép block đó được truyền trực tiếp qua các instance khác thông qua đường interconnect.

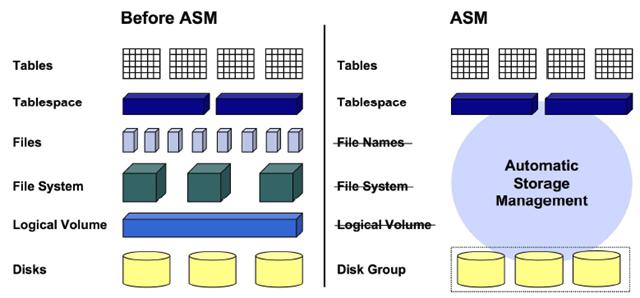
A diagram of a network

AI-generated content may be incorrect.

*Hình 2.4: Oracle Database với kiến trúc Oracle RAC*

Trong kiến trúc Oracle RAC phải sử dụng 2 card mạng public và private để chuyển yêu cầu từ client về máy chủ. Disk cũng phải cấu hình sharing để có thể đọc-ghi trên các node. Ngoài ra, Oracle RAC gồm các tiến trình và thành phần như:

* Oracle Grid Infrastructure: đây là phần mềm quản lý cụm máy chủ (cluster) được thiết kế đặc biệt cho Oracle RAC. Là nền tảng của Oracle RAC giúp cho phép các node giao tiếp với nhau tạo thành 1 máy chủ logic duy nhất. Nó hỗ trợ tối đa 64 nodes trong 1 cụm. Nó giám sát các tài nguyên như database instance, listener, các service và trạng thái của các node. Oracle Grid Infrastructure gồm các thành phần:
* OPROCd (Process Monitor Daemon): giúp giám sát và cung cấp chức năng I/O fencing để bảo tính toàn vẹn của cụm. Nếu process này gặp sự cố, node đó sẽ kích hoạt cơ chế fencing và được coi là nguy cơ cao. Node đó sẽ được restart để bảo vệ dữ liệu.
* CRSd (Cluster Ready Services Daemon): chịu trách nhiệm quản lý các tài nguyên, bao gồm stop, start các dịch vụ và xử lý failover cho các tài nguyên ứng dụng. CRSd còn quản lý OCR và lưu trữ trạng thái hiện tại của cụm, yêu cầu cần có IP public, private, VIP để hoạt động.
* OCSSd (Oracle Cluster Synchronization Service Daemon): cung cấp dịch vụ đồng bộ hóa giữa các node, truy cập thông tin các node và quản lý chức năng khóa (locking). Nếu process này gặp sự cố, node sẽ bị khởi động lại nhằm tránh tình trạng “split-brain”.
* Cluster Interconnect: được sử dụng để đồng bộ tài nguyên của cụm RAC và chuyển giao dữ liệu giữa các instance. Interconnect cần phải là mạng riêng, có độ sẵn sàng cao, tốc độ nhanh và độ trễ thấp (ít nhất 1GB, với việc sử dụng đa đường truyền như bonding trên Linux hay IPMP trên Solaris).
* ASM: là thành phần quản lý lưu trữ tự động của Oracle RAC, giúp quản lý việc sử dụng ổ đĩa thành dạng logic hiệu quả mà không cần cấu hình hệ thống tập tin truyền thống. ASM đã giúp giải quyết được vấn đề “nghẽn cổ chai” mà cơ sở dữ liệu thường gặp. Với ASM, các ổ đĩa vật lý được gom lại thành một “storage pool” mà từ đó dữ liệu được phân chia (striping) và phản chiếu (mirroring) tự động, giúp giảm tình trạng “nghẽn cổ chai” đồng thời đảm bảo tính sẵn sàng cho dữ liệu. Một điểm mạnh của ASM là khả năng thực hiện tái cân bằng (rebalance) trên các ổ đĩa khi có sự thay đổi về cấu trúc hoặc khi thêm mới ổ đĩa, mà không gây gián đoạn dịch vụ. Ngoài ra, mỗi node trong cluster RAC có thể chạy một ASM instance riêng biệt, giúp các node cùng truy cập và chia sẻ dữ liệu trên các disk chung một cách linh hoạt và ổn định.



*Hình 2.5: Kiến trúc truyền thống và kiến trúc ASM*

* Global Cache Services (GCS): là một thành phần thuộc Oracle Kernel Component, quản lý và duy trì các lock (khóa) dữ liệu trong bộ nhớ cache giữa các instance của Oracle RAC. GCS đảm bảo chỉ một instance có thể ghi cập nhật dữ liệu tại một thời điểm trong khi các instance khác chỉ được phép đọc nhờ cơ chế như “cache fusion”. Dữ liệu được chuyển thông qua mạng nội bộ (cluster interconnect) thay vì phải đọc từ đĩa, giúp giảm thiểu độ trễ và tối ưu hóa hiệu suất. Cơ chế này đảm bảo tính nhất quán dữ liệu trên tất cả các node và ngăn ngừa các xung đột khi nhiều instance cùng thao tác trên cùng một dữ liệu.

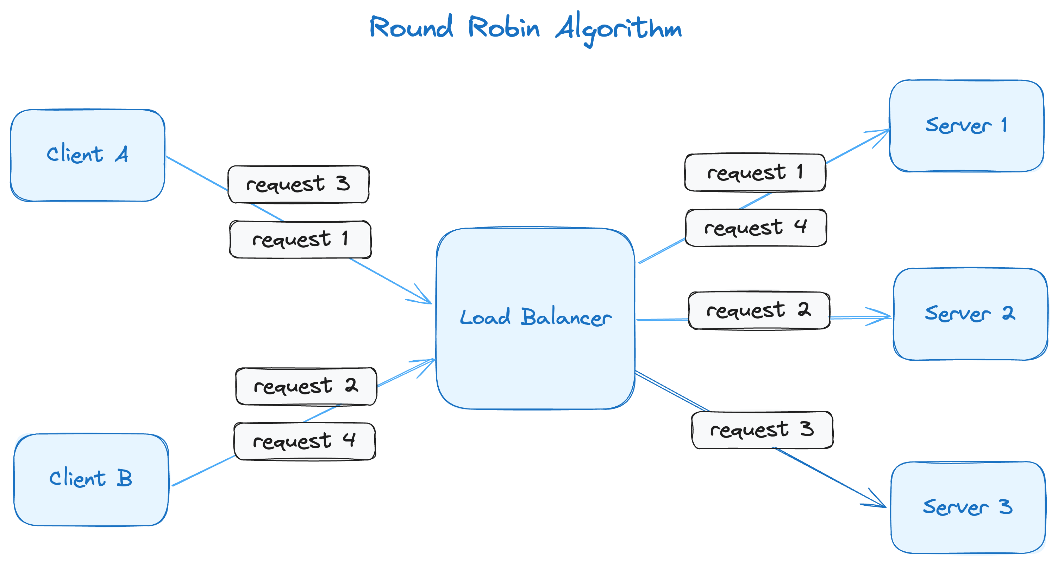
### Mô hình mạng trong Oracle RAC

Mỗi node yêu cầu một số tài nguyên mạng nhất định. Nó phải có quyền truy cập public network cho khách hàng và private network cho interconnect của cụm. Ít nhất cần có 2 card mạng.

* Public Network: Dùng để truy cập công cộng, bao gồm SCAN Listener và Virtual IP.
* Private Network: Được sử dụng cho interconnect của cụm, bao gồm Private IP.

#### SCAN IP (Single Client Access Name)

Là một tên đơn được gán cho cụm, DNS sẽ giải quyết tên này thành 3 địa chỉ IP theo kiểu[*round-robin*](https://docs.oracle.com/en-us/iaas/Content/Balance/Reference/lbpolicies.htm#Policies__RoundRobin) *(xoay vòng)*. IP này có thể di động trên các node trong cụm. Tên SCAN được cấu hình trong DNS, ba SCAN IP sẽ cùng gán cho một tên SCAN. Khi có yêu cầu từ client gửi đến, tên SCAN sẽ phân giải thành 1 trong 3 IP đã được cấu hình sử dụng cho SCAN Listener. SCAN IP giúp đơn giản hóa cấu hình kết nối khi có sự thay đổi về số lượng node trong cụm.



*Hình 2.6: Phương pháp Round Robin*

#### Virtual IP

Sau khi đi qua SCAN Listener sẽ chuyển tiếp yêu cầu đến Local Listener trên node có ít tải nhất. Local Listener sử dụng Virtual IP làm địa chỉ. Địa chỉ IP này chính là IP ảo gán cho mỗi node trong cụm RAC, đảm bảo khả năng failover. Nếu một node gặp lỗi, Virtual IP của node đó sẽ tự động chuyển sang node khác, giúp duy trì các kết nối client một cách liên tục mà không bị gián đoạn.

#### Private IP

* Địa chỉ IP này dùng cho giao tiếp nội bộ giữa các node trong RAC (mạng riêng/cluster interconnect). Nó không được công khai ra bên ngoài mà chỉ phục vụ cho các chức năng quan trọng như heartbeat, cache fusion và các giao tiếp quản lý giữa các node để đảm bảo hoạt động ổn định của cụm. Băng thông của Private Network được khuyến nghị nên từ 1 Gbps trở lên.

A diagram of a cluster

AI-generated content may be incorrect.

*Hình 2.7: Mô hình mạng trong Oracle RAC*

## Tìm hiểu về Data Guard

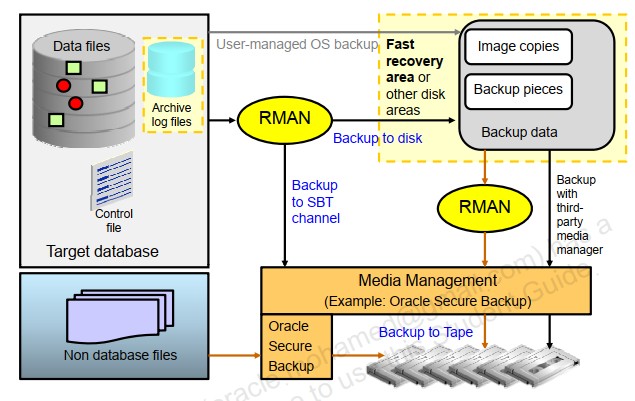
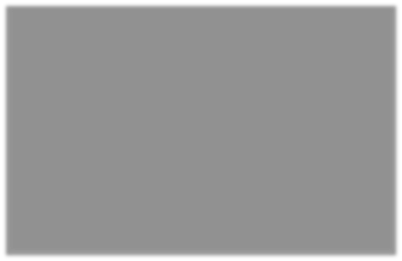
### Sao lưu với công cụ Recovery Manager

Recovery Manager (RMAN) là một công cụ quản lý sao lưu và khôi phục dữ liệu của Oracle, được thiết kế để quản lý và bảo vệ dữ liệu trong các cơ sở dữ liệu Oracle.

Mục đích cơ bản của RMAN là cung cấp các tính năng và công cụ cho việc thực hiện các tác vụ sao lưu và khôi phục dữ liệu một cách linh hoạt, hiệu quả và an toàn.

Đặc điểm nổi bật chính của RMAN là ngoài việc sao lưu khi CSDL đang tắt (Offline/Consistent/Cold), có thể sao lưu ngay cả lúc CSDL đang hoạt động (Online/Inconsistent/Hot). Ngoài ra, RMAN cung cấp tính năng sao lưu một phần, gồm chỉ những thay đổi kể từ bản sao lưu trước đó. Tương tự đối với việc khôi phục, công cụ cho phép khôi phục một phần hoặc toàn phần.

Các thành phần mà RMAN sao lưu gồm: Data Files, Control Files, Archived Redo Log, Parameters File. Đối với mức độ block – đơn vị nhỏ nhất trong kiến trúc lưu trữ vật lý, khi sao lưu, RMAN tự động kiểm tra những block rỗng và sẽ bỏ qua block này. Công cụ được tích hợp lên giao diện quản trị là Oracle Enterprise Manager và cũng có thể sử dụng bởi lệnh SQL. Ngoài ra, RMAN được tích hợp thêm thành phần Oracle Secure Backup để mã hóa và sao lưu ra ổ đĩa dạng băng từ (tape) hoặc sao lưu lên đám mây một cách bảo mật, an toàn.



*Hình 2.8: Kết hợp RMAN, Oracle Secure Backup và sao lưu bằng lệnh hệ thống*

### Khái niệm, kiến trúc của Data Guard

Khái niệm

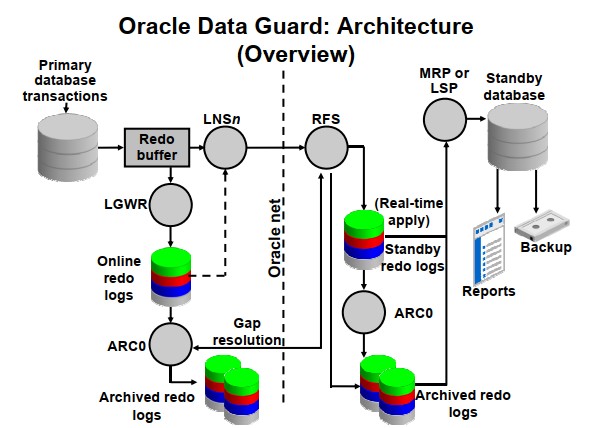
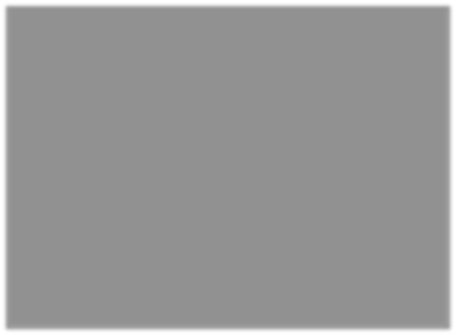
Trong các giải pháp phục hồi thảm họa sau sự cố, Data Guard (ODG) là một công nghệ được đánh giá cao trong việc đảm bảo tính sẵn sàng và liên tục của CSDL Oracle. ODG được xây dựng và tích hợp trên CSDL Oracle, gồm nhiều mô-đun chức năng như quản lý, giám sát, duy trì một hoặc nhiều CSDL dự phòng nhằm mục đích bảo vệ quy trình vận hành của doanh nghiệp khỏi sự cố.

Fuller (2014) cho rằng, CSDL dự phòng duy trì sự ổn định này với vai trò là bản sao và đồng bộ với CSDL chính. CSDL dự phòng có thể đặt cách xa trung tâm dữ liệu của CSDL chính để cải thiện mức an toàn, phòng trường hợp sự cố xảy ra cùng một địa điểm. Khi CSDL chính gặp lỗi, ODG sẽ chuyển đổi vận hành (switch role) từ CSDL chính – đang bị ảnh hưởng sang CSDL dự phòng, CSDL dự phòng sẽ đảm nhận vai trò của CSDL chính.

Kiến trúc

Trong kiến trúc của giải pháp ODG gồm 01 CSDL chính và CSDL dự phòng. Các CSDL liên kết và giao tiếp với nhau thông qua Oracle Network Service - môi trường mạng của Oracle, các CSDL này có thể đặt cách xa nhau về mặt địa lý. Có nhiều cách triển khai Oracle Data Guard, các CSDL có thể trong cùng một máy chủ và trung tâm dữ liệu, khác máy chủ hoặc khác trung tâm dữ liệu.

Trong giải pháp ODG, CSDL chính gửi những thông tin thay đổi được lưu dưới dạng vector có trong Online Redo Logs (ORLs) hoặc Redo Buffer Cache (RBC), giữ cho CSDL dự phòng luôn được đồng bộ hóa thông qua việc nhận và áp dụng.



Hình 2.9: Kiến trúc tổng quan giải pháp Oracle Data Guard

Trong kiến trúc giải pháp ODG sử dụng các tiến trình sao lưu và chuyển đổi vai trò. Môi trường ODG gồm cả tiến trình chung và tiến trình chỉ xuất hiện khi sử dụng giải pháp ODG. Các tiến trình và thành phần của giải pháp ODG gồm:

* Standby Redo Logs: thành phần lưu trữ thông tin thay đổi của CSDL dự phòng.
* Log Writer Network Server (LNSs): tiến trình nhận thông tin thay đổi và chuyển cho CSDL dự phòng.
* Remote File Server (RFS): tiến trình nhận thông tin thay đổi từ LNSs.
* Managed Recovery Process/Logical Standby Process (MRP/LSP): tiến trình áp dụng các thay đổi từ Standby Redo Log trên CSDL dự phòng. Tiến trình MRP cho CSDL dạng Physical và LSP sử dụng cho CSDL dạng Logical.

### Loại hình đồng bộ

Giải pháp ODG cung cấp nhiều loại hình đồng bộ, mỗi loại hình có đặc điểm khác nhau, linh hoạt và phù hợp với nhu cầu của doanh nghiệp. Có thể phân loại các loại hình theo hai góc độ: về loại hình CSDL dự phòng và về mức độ bảo vệ trong cơ chế truyền/đồng bộ hóa thông tin thay đổi giữa các CSDL.

Phân loại theo loại hình CSDL dự phòng

* CSDL dự phòng vật lý (Physical):
* Có cấu trúc File Systems và dữ liệu giống với CSDL chính.
* Được đồng bộ hóa với CSDL chính thông qua việc áp dụng dữ liệu thay đổi (redo data) từ CSDL chính.
* Cho phép thực hiện đồng thời tác vụ trả kết quả truy vấn cũng như việc nhận và áp dụng dữ liệu thay đổi vào CSDL. Chỉ mở CSDL ở chế độ chỉ đọc (read only).
* CSDL dự phòng lô-gic (Logical):
* Chỉ giống ở cấu trúc lô-gic, bộ nhớ vật lý có thể sử dụng các tính năng khác như Oracle Automatic Storage Management (ASM) để quản lý tập tin khác với File Systems theo mặc định. Ngoài ra, không có view, index giống với CSDL chính.
* Được đồng bộ hóa với CSDL chính thông qua việc nhận, chuyển hóa redo data thành SQL và thực thi trên CSDL dự phòng để áp dụng thay đổi. Điều này được thực hiện nhờ công cụ phân tích tệp tin lưu trữ logs là LogMiner.
* Cho phép thực hiện đồng thời các tác vụ trả kết quả truy vấn, áp dụng dữ liệu thay đổi vào CSDL, đặc biệt hơn là cho phép chỉnh sửa đối với các bảng, đối tượng không nằm trong vùng được áp dụng thay đổi. Mở CSDL ở chế độ đọc/ghi (Read/Write) - CSDL dự phòng Snapshot:
* Được chuyển từ CSDL dự phòng dạng vật lý.
* Cho phép thực hiện đọc/ghi trên toàn bộ cơ sở dữ liệu với mục đích kiểm thử.
* Sẽ không nhận và áp dụng các thông tin thay đổi.
* Các thay đổi sẽ bị ROLLBACK lại nếu như chuyển về CSDL dự phòng dạng vật lý.

Phân loại theo chế độ bảo vệ

Khi phân loại theo chế độ bảo vệ, cấu hình của các chế độ phụ thuộc vào các đối số được cài đặt (cụ thể là trong tham số *LOG\_ARCHIVE\_DEST\_n*), có 04 đối số chính như sau:

* SYNC: Xác nhận các redo data được gửi sang CSDL dự phòng thành công trước khi giao dịch được đánh dấu là COMMIT, nếu không, hệ thống sẽ dừng hoạt động/tiếp tục tùy thuộc vào chế độ bảo vệ được chọn.
* ASYNC: Không xác nhận việc redo data được nhận bởi CSDL dự phòng, do đó, giao dịch có thể COMMIT ngay lập tức trên CSDL chính.
* AFFIRM: Gửi tín hiệu tín hiệu Acknowledgement (ACK) *sau khi* redo data nhận được đã được ghi vào Standby Redo Logs.
* NOAFFIRM: Khác với AFFIRM ở chỗ sẽ gửi tín hiệu ACK, nhưng *gửi trước* khi được ghi vào Standby Redo Logs.

Ngoài các đối số, CSDL cũng cần phải cấu hình chế độ bảo vệ, đảm bảo CSDL sẽ thực hiện các phương thức bảo vệ dữ liệu khác nhau khi gặp sự cố.:

* Ưu tiên bảo vệ (max. protection):
* Chế độ bảo vệ mà CSDL chính sẽ đảm bảo rằng không có dữ liệu nào bị sót một cách tuyệt đối trong trường hợp CSDL chính gặp sự cố như thảm họa thiên tai, bị lỗi mạng hoặc bị lỗi với CSDL dự phòng.
* CSDL chính sẽ dừng hoạt động khi gặp sự cố khiến cho các tiến trình trong việc truyền tải/đồng bộ hóa thông tin thay đổi không thể ghi vào ít nhất một trong các CSDL dự phòng.
* Cần thiết lập chế độ cho cách truyền redo data với hai đối số: SYNC – đồng bộ hóa, và AFFIRM – xác nhận đã ghi xuống đĩa vật lý tại Standby Redo Logs, với ít nhất một CSDL dự phòng có Standby Redo Logs.
* Ưu tiên tính sẵn sàng (max. availability):
* Chế độ bảo vệ mà CSDL chính sẽ đảm bảo không có dữ liệu nào bị sót nhưng không hoàn toàn tuyệt đối, vì không tác động tới việc vận hành của CSDL chính trong một ràng buộc về thời gian cho trước
* Nếu có sự cố, CSDL chính sẽ hoạt động theo cách thức không đồng bộ (ASYNC, hoạt động không đợi xác nhận CSDL dự phòng đã nhận redo data hay chưa) cho đến khi ít nhất một CSDL dự phòng được đồng bộ về mặt thông tin thay đổi và chuyển về SYNC.
* Cần thiết lập hai đối số: SYNC – đồng bộ hóa và NOAFFIRM (không cần xác nhận đã ghi vào Standby Redo Logs) hoặc AFFIRM cho ít nhất một CSDL dự phòng có chứa Standby Redo Logs.
* Ưu tiên hiệu năng hệ thống (max. performance):
* Đây là chế độ mặc định của giải pháp ODG. Cung cấp việc bảo vệ dữ liệu thấp hơn hai mức còn lại về tính vẹn toàn, nhưng hiệu năng hệ thống của CSDL chính cao hơn.
* Giao dịch được xác nhận COMMIT, đồng thời thông tin thay đổi sẽ được lưu xuống tệp tin lưu trữ thông tin thay đổi ngay lập tức mà không cần quá trình xác nhận ACK từ CSDL dự phòng.
* Thông tin thay đổi (redo data) được truyền tới CSDL dự phòng theo cách không đồng bộ (ASYNC) với những thông tin thay đổi đã được COMMIT.
* Cần cấu hình đối số như sau: ASYNC – không đồng bộ và NOAFFIRM – không xác nhận đã ghi cho CSDL dự phòng đã có Standby Redo Logs. Bảng so sánh về các chế độ bảo vệ dưới đây sẽ có cái nhìn tổng quan và ngắn gọn hơn. Đây là các kết hợp đối số có ý nghĩa, có một số trường hợp kết hợp đối số khác không được sử dụng như ASYNC/AFFIRM:

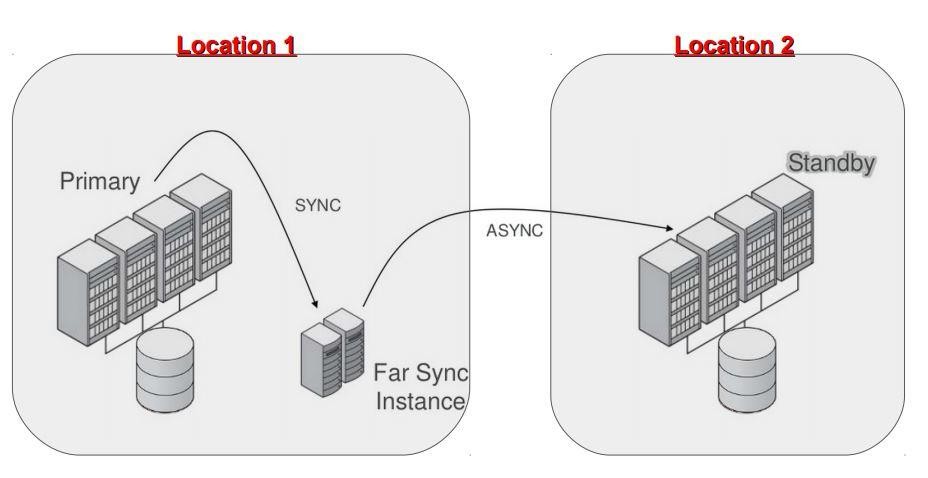
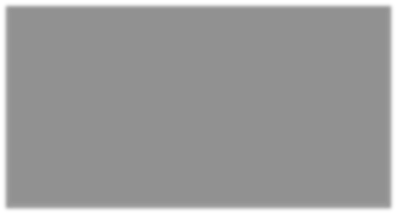
|  |  |  |  |
| --- | --- | --- | --- |
| Chế độ | Rủi ro | Chế độ truyền | Nếu tín hiệu ACK không gửi  (AFFIRM/NOAFFIRM) |
| Ưu tiên bảo vệ | Không mất đồng b b bộ dữ liệu | SYNC | CSDL chính sẽ treo/dừng hoạt động cho đến khi nhận được tín hiệu ACK |
| Ưu tiên tính sẵn sàng | Không mất đồng bộ dữ liệu khi  CSDL chính gặp sự cố hoặc mất khi cả hai gặp sự cố | SYNC/AFFIRM  hoặc  SYNC/NOAFFIRM  (Fast Sync) | CSDL chính sẽ chờ trong một khoảng thời gian được xác định là NetTimeout, nếu hết, sẽ tiếp tục hoạt động |
| Ưu tiên hiệu năng | Mất đồng bộ dữ liệu nếu CSDL chính gặp sự cố | ASYNC | CSDL chính không chờ tín hiệu ACK |

*Bảng 2.1: Cấu hình đối số phương thức truyền/xác nhận*

Đồng bộ trung gian với Far Sync

Tính năng Far Sync xây dựng một Instance làm điểm trung chuyển thông tin thay đổi tới nhiều các CSDL dự phòng khác. Thực tế, Far Sync là một hệ thống có kích thước gọn nhẹ, tiêu thụ ít tài nguyên về lưu trữ cũng như xử lý.

Far Sync giống với CSDL dự phòng thông thường như quản lý các Control File, nhận redo data vào Standby Redo Logs và lưu trữ khi Log Switch xuống Archived Redo Logs. Ngược lại, khác với CSDL dự phòng ở điểm không lưu trữ một số tệp tin không quan trọng như Data Files, không hỗ trợ chuyển đổi vai trò và chỉ thiết lập được tại 02 chế độ bảo vệ: *max. performance* hoặc *max. availability*.



Hình 2.10: Tính năng Far Sync trong giải pháp Oracle Data Guard

Far Sync không bị giới hạn bởi khoảng cách do cách truyền không đồng bộ, không ảnh hưởng đến hiệu năng của CSDL chính. Hơn hết, Far Sync giúp giảm tải CSDL chính trong việc truyền tải redo data đến hàng loạt CSDL dự phòng. Nếu CSDL chính lỗi, các CSDL dự phòng sẽ lấy các redo data cuối cùng từ Far Sync trong việc chuyển đổi vai trò, đảm bảo tính vẹn toàn. Ngoài ra, việc bổ sung các tính năng như đóng gói, mã hóa dữ liệu được Far Sync đảm nhận nhằm giảm tải cho CSDL chính.

### Cơ chế tương tác giữa các thành phần

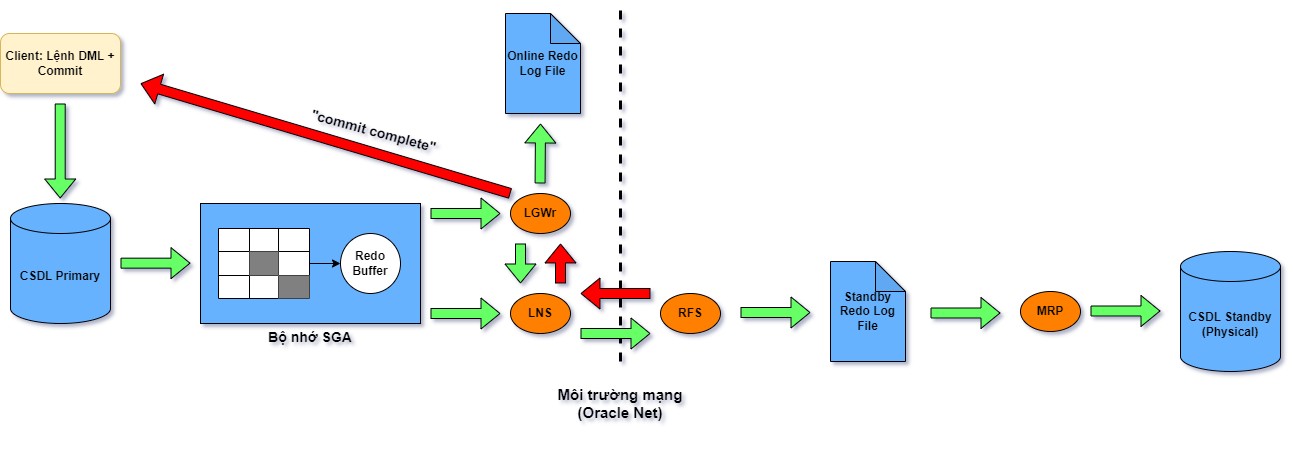
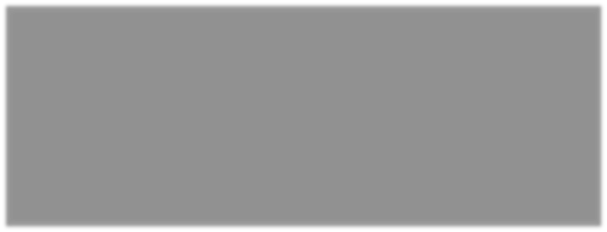
Các thành phần giao tiếp với nhau trong kiến trúc giải pháp Oracle Data Guard thông qua 03 cơ chế chính:

* Cơ chế vận chuyển thông tin thay đổi (Redo Transport Services): gồm các tiến trình ARCn, LGWr, LNSs, RFS phục vụ việc truyền redo data.
* Cơ chế áp dụng thông tin thay đổi (Log Apply Services): gồm các tiến trình MRP, LSP phục vụ việc áp dụng các redo data vào CSDL dự phòng.
* Cơ chế quản lý/chuyển đổi vai trò (Role Management Services): gồm tiến trình DMON và DGR thực hiện việc giám sát và thay đổi vai trò của các CSDL.

Hai chế độ bảo vệ *Ưu tiên hiệu năng* và *Ưu tiên bảo vệ* có những đặc điểm nổi trội khác nhau, đặc biệt hơn so với *Ưu tiên về tính sẵn sàng*. Vì vậy, để hiểu được rõ được sự khác biệt và cơ chế hoạt động của luồng dữ liệu, hình vẽ cùng miêu tả các pha sẽ minh họa cơ chế hoạt động hai chế độ, hoạt động trong loại CSDL dự phòng vật lý.

Với các hình vẽ minh họa, các mũi tên có tông màu nhạt thể hiện luồng truyền redo data từ CSDL chính, tông màu đậm thể hiện luồng trả về tín hiệu ACK từ CSDL dự phòng.

Chế độ *Ưu tiên bảo vệ*



*Hình 2.11: Luồng hoạt động Data Guard với chế độ Ưu tiên bảo vệ*

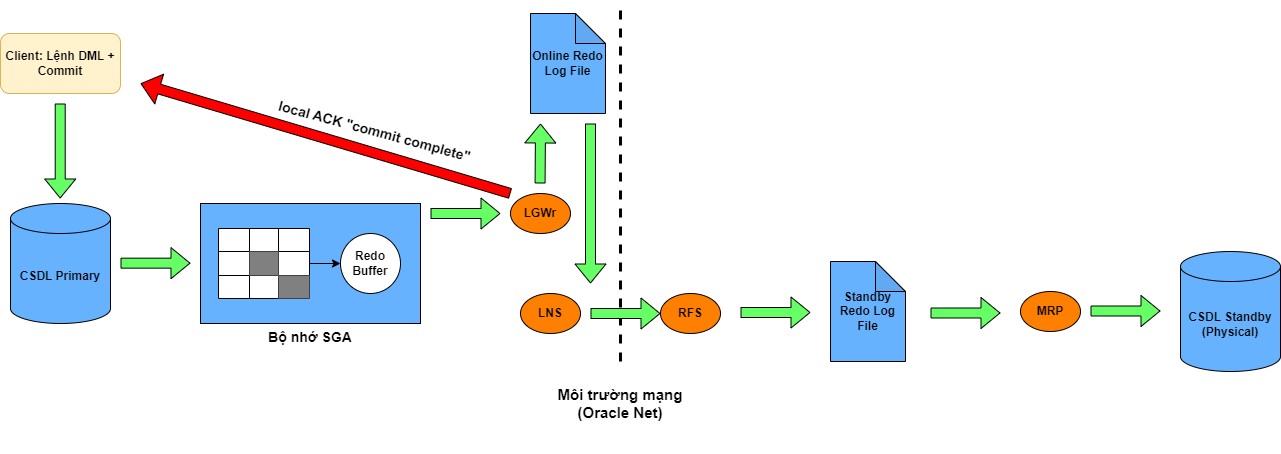
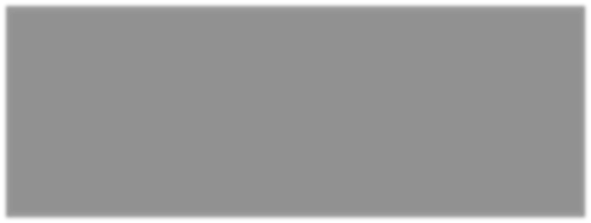
Các pha của cơ chế *Ưu tiên bảo vệ* được thể hiện tuần tự với các bước như sau, khi cấu hình SYNC/AFFIRM:

1. Người dùng tạo các giao dịch thông qua các lệnh Data Manipulation Language (DML). Thông tin thay đổi được lưu vào Redo Buffer Cache.
2. Thông tin thay đổi sẽ được tiến trình LGWR đưa và lưu xuống Online Redo Log sau khi COMMIT, nhưng hệ thống chưa báo “commit complete” ngay.
3. Tiến trình LNS sẽ nhận những thông tin thay đổi trong lúc LGWR xuất ra. Những thông tin thay đổi này sẽ được chuyển cho tiến trình RFS thuộc CSDL dự phòng, nhằm thực hiện sao lưu.
4. Sau khi nhận được thông tin thay đổi, tiến trình RFS sẽ ghi những thông tin thay đổi xuống Standby Redo Log Files.
5. Nếu CSDL dự phòng sử dụng tính năng Real-time Apply, thì ngay lập tức, các thông tin thay đổi này sẽ được áp dụng vào dữ liệu lưu trữ vật lý tại CSDL dự phòng với tiến trình khôi phục MRP.
6. Với SYNC/AFFIRM, sau khi dữ liệu đã được áp dụng thành công, tiến trình RFS sẽ phản hồi lại cho tiến trình LNS tín hiệu ACK. Lúc này, hệ thống mới phản hồi người dùng “commit complete”. Tại đây, nếu gặp sự cố về môi trường mạng, khiến cho RFS không thể gửi cho LNS, CSDL chính sẽ đợi tới khi nào nhận được thông tin dẫn tới hệ thống treo

Ngoài ra, khi xảy ra Log Switch trên CSDL chính, sẽ kích hoạt một Trigger giúp CSDL dự phòng cũng thực hiện Log Switch đối với Standby Redo Log Files nhằm đảm bảo tính toàn vẹn.

Đối với RFS, tiến trình này sẽ gửi trực tiếp redo data xuống Archive Log File nếu: Không có Standby Redo Logs (1), Standby Redo Log được cài đặt nhỏ hơn kích thước của Online Redo Logs (2), tất cả Standby Redo Logs đều chưa được lưu trữ (archived) (3) và nếu RFS thực hiện nhận redo data từ tiến trình ARCn trong cơ chế Gap Resolution (4).

Chế độ *Ưu tiên hiệu năng*



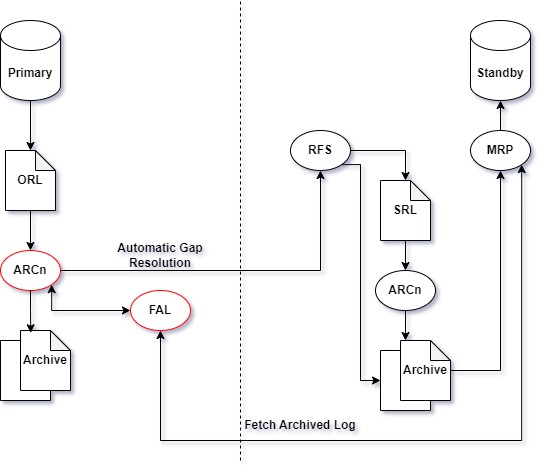
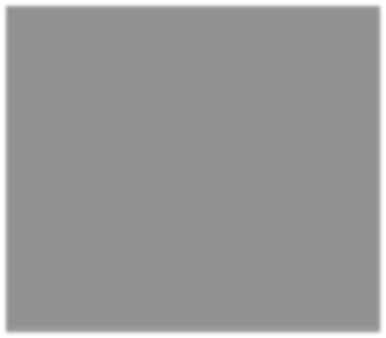
*Hình 2.12: Luồng hoạt động Data Guard với chế độ Ưu tiên hiệu năng*

Trong chế độ ưu tiên hiệu năng, với cơ chế ASYNC/NOAFFIRM, CSDL chính sẽ không yêu cầu nhận tín hiệu xác nhận nào từ CSDL dự phòng trong việc ghi thành công vào SRLs. Các redo data được LNS tiếp nhận và lấy tại Online Redo Logs nhằm tăng hiệu năng của tiến trình LGWr trong việc ghi redo data.

Khi người dùng gõ lệnh DML và yêu cầu COMMIT, hệ thống sẽ ngay lập tức trả lại tín hiệu COMMIT thành công – “commit complete”, do không phải chờ phản hồi từ CSDL dự phòng. Trường hợp này, tín hiệu ACK còn được gọi là “local ACK”, có nghĩa là tín hiệu này sẽ không xuất phát từ tiến trình RFS truyền qua môi trường Oracle Net.

Hiện tượng Archive Redo Gap và cơ chế xử lý

Trong môi trường Oracle Data Guard, tình trạng trễ dữ liệu xảy ra khi kết nối giữa các CSDL gặp sự cố hoặc gói tin gửi bị hỏng. Thuật ngữ để miêu tả tình trạng này là “Archive Redo Gap Sequence”. Oracle cung cấp 02 cơ chế để xử lý tình trạng này là “Automatic Gap Resolution” và “Fetch Archive Log – FAL”.



Hình 2.13: Minh họa cơ chế xử lý thiếu trong việc truyền thông tin

Trong chế độ *Ưu tiên hiệu năng*, do cơ chế ASYNC, các redo data được COMMIT liên tục và không đợi xác nhận từ phía CSDL dự phòng. Khi gặp sự cố mạng, tiến trình LNS của CSDL dự phòng không thể nhận những thông tin thay đổi. Trong một khoảng thời gian như trên sẽ gây ra hiện tượng trễ dữ liệu.

*Với cơ chế Automatic Gap Resolution:*

* Đây là cơ chế chủ động của CSDL chính. Tiến trình ARCn của CSDL chính sẽ liên tục gửi lệnh ping đến cho tiến trình RFS của CSDL dự phòng để xác định trạng thái khi có kết nối.
* Nếu kết nối mạng khôi phục, ARCn sẽ tiến hành ping kèm truy vấn xác định thông tin về Archive Redo Logs mới nhất. Nếu Log Sequence tại CSDL chính lớn hơn Log Sequence tại CSDL dự phòng, RFS sẽ thông báo lại số lượng Archive Redo Logs còn thiếu, từ đây ARCn sẽ gửi cho CSDL dự phòng theo yêu cầu.
* Ngoài ra, tiến trình LNS cũng sẽ hỗ trợ trong việc cập nhật/gửi đi các redo data mới nhất từ Redo Buffer Cache, Online Redo Logs, giúp CSDL dự phòng nhanh chóng khôi phục lại được trạng thái cập nhật gần nhất với CSDL chính, cho đến khi MRP có thể tiếp tục áp dụng với thời gian thực.

*Với cơ chế Fetch Archive Log (FAL):*

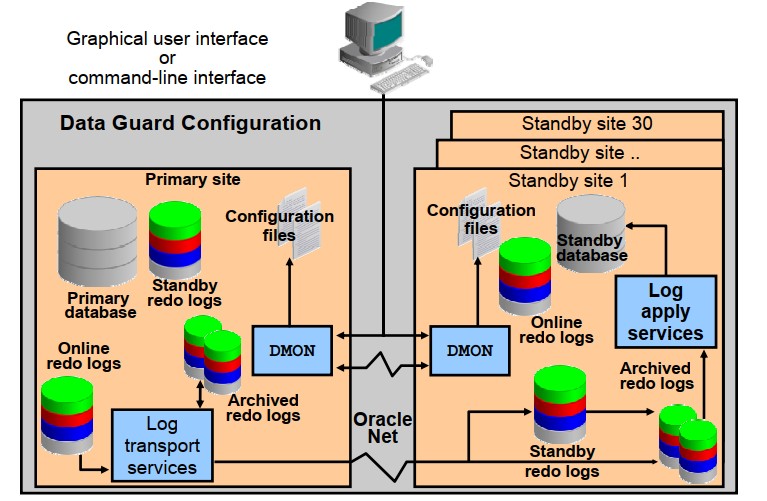
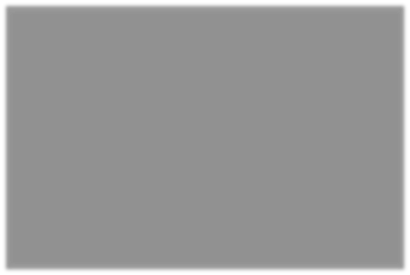
* Đây là cơ chế chủ động của CSDL dự phòng (dạng vật lý), do hành động “fetch” – yêu cầu/lấy thông tin đang bị thiếu được thực hiện một cách chủ động. - Sau khi được tiến trình ARCn gửi và RFS nhận vào, CSDL dự phòng sẽ cập nhật trong Control File của nó về tên và địa điểm của các Archive Log. Khi MRP thấy được những thay đổi mới trong Control File, hệ thống sẽ tiến hành áp dụng những thay đổi này vào CSDL dự phòng. Nhưng khi MRP thấy thông tin về tệp Archive Log từ CSDL chính lỗi/hỏng/thiếu thì nó sẽ sử dụng cơ chế chủ động thông qua tiến trình FAL, yêu cầu gửi lại Archive Log. Cơ chế này cần thiết lập hai tham số chính trên CSDL dự phòng như sau:

|  |  |
| --- | --- |
| Tham số | Miêu tả |
| FAL\_SERVER | Trỏ tới CSDL nhận yêu cầu và xử lý gửi thông tin thiếu |
| FAL\_CLIENT | Trỏ tới CSDL yêu cầu gửi thông tin thiếu |

*Bảng 2.2: Tham số cấu hình cho cơ chế Fetch Archive Log*

### Oracle Data Guard Broker

Theo Fuller (2014), Data Guard Broker là một tính năng được tích hợp trong Oracle Database Server, dùng để quản trị tập trung các CSDL thuộc môi trường Oracle Data Guard. Các thành phần của Broker gồm: trình điều khiển (thuộc client-side), tiến trình DMON và configuration files (thuộc server-side).



*Hình 2.14: Kiến trúc Data Guard với tính năng Data Guard Broker*

Tính năng Data Guard Broker có các ưu điểm sau khi so sánh với việc không sử dụng Data Guard Broker:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | Sử dụng Broker | Không sử dụng Broker |
|  |  |  |
| Quản trị | Quản trị tập trung | Việc quản trị thực hiện riêng lẻ đối với từng CSDL |
| Tạo CSDL dự phòng | Sử dụng OEMCC để đơn  giản hóa và tự động việc tạo | Tạo thủ công (bằng cách sử dụng công cụ RMAN hoặc các công |
|  | CSDL dự phòng (với GUI) (gồm control file, online redo log files, datafiles, và các tệp tin lưu trữ tham số) | cụ khác):   * Sao chép các tệp tin của CSDL chính cho CSDL   dự phòng   * Tạo control file trên CSDL dự phòng * Tạo file tham số trên CSDL dự phòng * Sao chép password file từ CSDL chính sang CSDL dự phòng |
| Cấu hình và quản lý | Cho phép cấu hình và quản lý nhiều CSDL tập trung và quản lý cấu hình kết nối của các CSDL thông qua một tệp tin duy nhất | * Thiết lập Redo Transport Services và Log Apply   Services tại mỗi CSDL   * Quản lý CSDL riêng lẻ |
| Điều khiển | * Tự động thiết lập Redo Transport Services và Log   Apply Services   * Đơn giản hóa việc chuyển đổi vai trò | * Sử dụng SQL để quản lý * Sử dụng nhiều lệnh hệ thống để quản lý các CSDL cho việc chuyển đổi vai trò cũng như điều khiển các tiến trình |
| Theo dõi | * Cho phép theo dõi hiệu năng hệ thống, cấu hình và các tham   số khác   * Cung cấp báo cáo chi tiết về hệ thống | * Chỉ theo dõi cố định vào khoảng thời gian nhất định * Không tập hợp các tham số cần theo dõi cùng một lúc |

Bảng 2.3: So sánh việc sử dụng Broker vào hệ thống

Trong mô hình quản lý của Broker, tập tin cấu hình Configuration chứa thông tin về các CSDL, gồm 01 CSDL chính, tối đa 30 CSDL dự phòng hoặc Far Sync.

A diagram of a server

AI-generated content may be incorrect.

*Hình 2.15: Mối quan hệ giữa các thành phần trong mô hình Broker*

Resource là đơn vị nhỏ nhất được quản lý bởi Broker, thành phần thể hiện một hoặc nhiều (đối với mô hình Real Application Clusters - RAC) Instance của CSDL. Database hoặc Host Server là tập hợp nhiều Resources, hay chính là hệ thống CSDL chính hoặc dự phòng mà Instance chạy trên chính nó.

Tại server-side, các thành phần của DGB gồm tiến trình DMON và tệp tin thông tin cấu hình. DMON là tiến trình nền, chạy ở mỗi Database Host khi Broker khởi động và được quản lý bởi Broker. Tệp tin thông tin cấu hình chứa các cài đặt về thuộc tính, trạng thái của Database Host.

Tiến trình DMON thực hiện quản lý và sao chép các tệp tin thông tin cấu hình cho mỗi Database Host mà Broker quản lý. Các tiến trình DMON ở mỗi CSDL khác nhau giao tiếp thông qua môi trường mạng Oracle Net để quản lý việc chuyển đổi vai trò CSDL cũng như cung cấp các chỉ số liên quan tới hiệu năng hệ thống.

## Tổng kết chương 2

Như vậy, chương 2 đã đi sâu vào khai thác các khái niệm, kiến trúc và thành phần của giải pháp Oracle RAC và Data Guard. Chương trình bày chi tiết các mô hình Cluster, từ kiến trúc Shared-Nothing đến Shared-Disk, đồng thời giải thích vai trò và chức năng của từng thành phần cốt lõi như Oracle Grid Infrastructure, ASM, GCS và các tiến trình như OPROCd, CRSd, OCSSd, EVMd cùng Cluster Interconnect. Phần mô hình mạng Oracle RAC cũng được nêu bật với các khái niệm SCAN IP, Virtual IP và Private IP, làm rõ cách thức kết nối và giao tiếp giữa các node trong cụm. Bên cạnh đó, chương cũng giới thiệu kiến trúc và cơ chế hoạt động của Oracle Data Guard, từ việc sử dụng công cụ RMAN cho sao lưu - khôi phục đến các phương thức đồng bộ dữ liệu giữa CSDL chính và CSDL dự phòng, đảm bảo quá trình chuyển đổi và khôi phục sau sự cố được thực hiện một cách hiệu quả. Những nội dung này đã tạo nền tảng lý thuyết vững chắc cho việc triển khai mô hình và kiểm thử các tình huống thực tế sẽ được làm ở chương tiếp theo.

# TRIỂN KHAI MÔ HÌNH VÀ ĐÁNH GIÁ HIỆU QUẢ

## Thông tin cấu hình máy chủ ảo cài đặt mô hình

### Oracle RAC

Việc xây dựng mô hình này yêu cầu lượng tài nguyên khá lớn về memory, dung lượng đĩa, số CPU và card mạng.

Phần cứng:

* Phần cứng của CSDL chính và CSDL dự phòng có thể khác nhau về số lượng bộ xử lý trung tâm (CPU), kích thước bộ nhớ (Memory), và cấu hình lưu trữ (Storage)
* Cho phép hệ thống xử lý cũng như phiên bản cài đặt Oracle Database Software có kích thước đơn vị biểu diễn thông tin khác nhau (32-bit hoặc 64-bit)
* Yêu cầu phiên bản cài đặt cho Oracle Database từ Enterprise Edition trở lên cho cả hệ thống CSDL chính và CSDL dự phòng. Data Guard không hỗ trợ cho Oracle Database Standard Edition.
* Nếu sử dụng công cụ quản lý bộ nhớ Automatic Storage Management (ASM) hoặc Oracle Managed Files (OMF) thì cần sử dụng giống nhau ở cả hai hệ thống CSDL chính và CSDL dự phòng nếu sử dụng CSDL dạng vật lý. Đối với trường hợp kết hợp các phương thức thì cũng tương tự ở cả hai hệ thống.

|  |  |  |
| --- | --- | --- |
| Máy chủ | Thông tin | Giá trị |
| Node 1 | Oracle CPU | 10 |
| Memory (Gigabyte) | 18 |
| Gateway | 172.16.30.254 |
| Private IP | 172.11.90.67 |
| Public IP | 172.16.30.67 |
| Region | Ha Noi |
| Hostname | lab-oracle-1 |
| Operating System | Oracle Linux 7.9 |
| Local Disk 1 | 100 |
| Shared Disk 1 | 60 |
| Shared Disk 2 | 50 |
| Node 2 | Oracle CPU | 10 |
| Memory (Gigabyte) | 18 |
| Gateway | 172.16.30.254 |
| Private IP | 172.11.90.68 |
| Public IP | 172.16.30.68 |
| Region | Ha Noi |
| Hostname | lab-oracle-1 |
| Operating System | Oracle Linux 7.9 |
| Local Disk 1 | 100 |
|  | Shared Disk 1 | 60 |
|  | Shared Disk 2 | 50 |

###### Bảng 3.1: Cấu hình máy chủ ảo cài đặt Oracle RAC

### Data Guard

Phần cứng:

* Phần cứng của CSDL chính và CSDL dự phòng có thể khác nhau về số lượng bộ xử lý trung tâm (CPU), kích thước bộ nhớ (Memory), và cấu hình lưu trữ (Storage)
* Cho phép hệ thống xử lý cũng như phiên bản cài đặt Oracle Database Software có kích thước đơn vị biểu diễn thông tin khác nhau (32-bit hoặc 64-bit)
* Yêu cầu phiên bản cài đặt cho Oracle Database từ Enterprise Edition trở lên cho cả hệ thống CSDL chính và CSDL dự phòng. Data Guard không hỗ trợ cho Oracle Database Standard Edition.
* Nếu sử dụng công cụ quản lý bộ nhớ Automatic Storage Management (ASM) hoặc Oracle Managed Files (OMF) thì cần sử dụng giống nhau ở cả hai hệ thống CSDL chính và CSDL dự phòng nếu sử dụng CSDL dạng vật lý. Đối với trường hợp kết hợp các phương thức thì cũng tương tự ở cả hai hệ thống.

|  |  |  |
| --- | --- | --- |
| Chính | Oracle CPU | 6 |
| Memory (Gigabyte) | 16 |
| Gateway | 172.16.30.254 |
| Private IP | 10.0.12.202 |
| Public IP | 172.16.30.70 |
| Region | Hai Phong |
| Hostname | DG |
| Operating System | Oracle Linux 7.9 |
| Local Disk 1 | 80 |
| Local Disk 2 | 60 |
| Local Disk 3 | 50 |

###### Bảng 3.2: Cấu hình máy chủ ảo cài đặt Oracle Data Guard

Để triển khai giải pháp Oracle Data Guard, cần cài đặt Oracle Database Software phiên bản Enterprise Edition. Trong phần thực nghiệm, máy chủ chính đã được cài đặt Database theo kiến trúc Real Application Cluster (RAC). Cả hai máy chủ được cài đặt và cấu hình các đường dẫn giống nhau. Các phần tiếp theo sẽ đi sâu về cách triển khai Oracle RAC và Data Guard. Lệnh và các cấu hình đầy đủ sẽ được đính kèm theo phụ lục tại cuối bài.

## Triển khai giải pháp Oracle RAC

### Kiến trúc tổng quan

A diagram of a computer

AI-generated content may be incorrect.

###### Hình 3.1: Kiến trúc tổng quan thực nghiệm giải pháp Oracle RAC

Với mục đích đảm bảo tính sẵn sàng cao, các node trong hệ thống Oracle RAC nên được đặt trong cùng một Data Center và được thiết lập kết nối mạng nội bộ tốc độ cao để đảm bảo khả năng đồng bộ dữ liệu trên memory (sử dụng cơ chế Cache Fusion) giữa các node. Mỗi node đều được cài đặt đầy đủ các thành phần hạ tầng công nghệ thông tin, có liên kết chặt chẽ với nhau thông qua mạng private (interconnect) và sử dụng shared storage là ASM.

### Cấu hình mạng.

Trên mỗi node Oracle RAC, cần cấu hình hai card mạng tách biệt cho mạng công cộng (public) và mạng nội bộ (private interconnect) để đảm bảo hiệu năng và độ tin cậy của cluster. Sau đó, cập nhật file /etc/hosts trên cả hai node với các ánh xạ IP cho public IP, VIP, SCAN và private interconnect.

Thiết lập card mạng:

* Public NIC cho kết nối client, VIP, SCAN.
* Private NIC (interconnect) cho giao tiếp nội bộ giữa các instance

Cập nhật file /etc/hosts:

Trên mỗi máy chủ Oracle Linux, sẽ có 1 file /etc/hosts. Đây là một file ánh xạ tên host (hostname) sang địa chỉ IP. Nó giúp hệ thống định danh các máy chủ (host) trong mạng mà không cần phải truy vấn DNS. Thực hiện mở file /etc/hosts và thêm các thông tin ánh xạ:

# vi /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

#public

172.16.30.67 lab-oracle-01

172.16.30.68 lab-oracle-02

#private

172.11.90.67 lab-oracle-01-priv

172.11.90.68 lab-oracle-02-priv

#vip

172.16.30.83 lab-oracle-01-vip

172.16.30.84 lab-oracle-02-vip

#scan

172.16.30.72 lab-oracle-scan

172.16.30.73 lab-oracle-scan

172.16.30.74 lab-oracle-scan

Trong đó:

2 địa chỉ đâù tiên: giúp máy chủ nhận diện chính bản thân nó khi gọi về localhost hay còn gọi là loopback interface được cấu hình mặc định trong Oracle Linux. 127.0.0.1 dành cho địa chỉ IPv4 và ::1 dành cho địa chỉ IPv6.

Public: Đây là 2 địa chỉ IP công cộng của mỗi node trong cluster, được sử dụng để kết nối từ bên ngoài vào hệ thống. Các dịch vụ như SCAN listener và VIP listener sẽ sử dụng địa chỉ này để phục vụ các kết nối từ client.

Private: Đây là địa chỉ IP riêng (private) được sử dụng cho giao tiếp nội bộ giữa các node trong cluster. Oracle Grid Infrastructure sử dụng địa chỉ này để đồng bộ hóa dữ liệu và quản lý trạng thái của các node trong cluster.

Vip: là địa chỉ IP ảo được gán cho mỗi node trong cluster. Khi một node gặp sự cố, VIP sẽ được chuyển sang node khác để đảm bảo tính liên tục của dịch vụ. Điều này giúp giảm thời gian chờ kết nối lại từ phía client.

Scan: SCAN (Single Client Access Name) cung cấp một điểm truy cập duy nhất cho client kết nối vào Oracle RAC. SCAN cho phép Oracle tự động phân phối kết nối đến các node trong cluster, hỗ trợ cân bằng tải và chuyển đổi dự phòng.

### Chuẩn bị môi trường

* Cài các package

Ở các phiên bản cũ, để chuẩn bị môi trường cho việc cài Oracle Database. Phải làm nhiều việc thủ công và tốn thời gian. Nhận thấy điều đó, Oracle đã phát triển gói cài oracle-database-preinstall cho các phiên bản từ 11g trở lên và chỉ cần cài đặt gói này, nó sẽ tự động thiết lập môi trường chuẩn cho Oracle Database.

Gói cài này sẽ có sẵn trong hệ điều hành đối với Oracle Linux. Đối với các hệ điều hành khác cần tải thủ công trên website của hãng về để cài đặt.

Đối với Oracle Linux, muốn chuẩn bị môi trường cho Oracle Database 19c thì lệnh cài là:

# yum install -y oracle-database-preinstall-19c

Trong đó:

yum: là công cụ quản lý gói dùng để cài đặt, cập nhật và gỡ các phầm mềm trên hệ thống dựa trên RPM của CentOS, RHEL

install: tùy chọn của yum dùng để cài đặt gói

-y: Tùy chọn tự động trả lời yes cho tất cả câu hỏi khi cài đặt

oracle-database-preinstall-19c: là gói phần mềm được cài đặt. Gói này sẽ tạo người dùng oracle, nhóm oinstall, dba. Thiết lập các tham số trong kernel, cài đặt các gói phần mềm liên quan như gcc, libaio,.. để có môi trường chuẩn cho việc cài đặt Oracle Database.

* Tạo biến môi trường

Trong Linux, biến môi trường (environment variables) là những giá trị được lưu trong hệ thống và dùng để cấu hình hoặc định hướng cho các chương trình chạy đúng cách. Đối với Oracle Database, biến môi trường giúp hệ điều hành và các công cụ như SQL\*Plus, RMAN, DBCA… biết được Oracle đang ở đâu, dùng phiên bản nào, và đang kết nối tới CSDL nào.

Biến môi trường được đặt trong file ~/.bash\_profile và mỗi user sẽ có biến môi trường riêng.

Cấu hình cho user grid:

* Node1:

$vi ~/.bash\_profile

export ORACLE\_HOSTNAME=lab-oracle-01

export ORACLE\_BASE=/u01/app/grid

export ORACLE\_HOME=/u01/app/19.0.0/grid

export ORACLE\_SID=+ASM1

export PATH=/usr/sbin:$PATH:$ORACLE\_HOME/bin:$ORACLE\_HOME/OPatch

export LD\_LIBRARY\_PATH=$ORACLE\_HOME/lib:/lib:/usr/lib

export CLASSPATH=$ORACLE\_HOME/jlib:$ORACLE\_HOME/rdbms/jlib

* Node2:

$vi .bash\_profile

export ORACLE\_HOSTNAME=lab-oracle-02

export ORACLE\_BASE=/u01/app/grid

export ORACLE\_HOME=/u01/app/19.0.0/grid

export ORACLE\_SID=+ASM2

export PATH=/usr/sbin:$PATH:$ORACLE\_HOME/bin:$ORACLE\_HOME/OPatch

export LD\_LIBRARY\_PATH=$ORACLE\_HOME/lib:/lib:/usr/lib

export CLASSPATH=$ORACLE\_HOME/jlib:$ORACLE\_HOME/rdbms/jlib

Chạy file .bash\_profile để các biến môi trường cập nhật vào hệ thống

$ . .bash\_profile

Trong đó:

ORACLE\_BASE: thư mục gốc chứa Grid software

ORACLE\_HOME: đường dẫn đến Oracle Grid Infrastructure home

ORACLE\_SID=+ASM1|+ASM2`: tên instance ASM trên từng node

PATH: thêm `grid/bin` và OPatch vào đường dẫn thực thi

LD\_LIBRARY\_PATH`: nạp thư viện của Grid và thư viện hệ thống :

CLASSPATH: bao gồm các thư viện Java của Grid và RDBMS

Cấu hình cho user oracle:

* Node1:

$vi ~/.bash\_profile

export ORACLE\_BASE=/u01/app/oracle

export ORACLE\_HOME=$ORACLE\_BASE/product/19.0.0/dbhome\_1

export GRID\_HOME=/u01/app/19.0.0/grid

export PATH=/usr/sbin:$PATH:$ORACLE\_HOME/bin:$GRID\_HOME/bin:$ORACLE\_HOME/OPatch

export LD\_LIBRARY\_PATH=$ORACLE\_HOME/lib:$GRID\_HOME/lib:/lib:/usr/lib

export CLASSPATH=$ORACLE\_HOME/jlib:$ORACLE\_HOME/rdbms/jlib:$GRID\_HOME/jlib:$GRID\_HOME/rdbms/jlib

export ORACLE\_SID=laboracle1

* Node2:

$vi ~/.bash\_profile

export ORACLE\_BASE=/u01/app/oracle

export ORACLE\_HOME=$ORACLE\_BASE/product/19.0.0/dbhome\_1

export GRID\_HOME=/u01/app/19.0.0/grid

export PATH=/usr/sbin:$PATH:$ORACLE\_HOME/bin:$GRID\_HOME/bin:$ORACLE\_HOME/OPatch

export LD\_LIBRARY\_PATH=$ORACLE\_HOME/lib:$GRID\_HOME/lib:/lib:/usr/lib

export CLASSPATH=$ORACLE\_HOME/jlib:$ORACLE\_HOME/rdbms/jlib:$GRID\_HOME/jlib:$GRID\_HOME/rdbms/jlib

export ORACLE\_SID=laboracle2

Chạy file .bash\_profile để các biến môi trường cập nhật vào hệ thống

$ . .bash\_profile

Trong đó:

ORACLE\_BASE: thư mục gốc chứa Oracle Software

ORACLE\_HOME: đường dẫn Oracle Database home

GRID\_HOME`: đường dẫn Grid Infrastructure home

PATH: thêm `oracle/bin` và OPatch vào đường dẫn thực thi

LD\_LIBRARY\_PATH`: nạp thư viện của Grid và thư viện hệ thống :

CLASSPATH: bao gồm các thư viện Java của Grid và RDBMS

ORACLE\_UNQNAME=laboracle: tên duy nhất của database trên toàn cụm RAC

ORACLE\_SID=laboracle1|laboracle2`: mỗi node sẽ có một SID, là tên của Instance.

* Start chronyd.service để đồng bộ tgian

Oracle RAC yêu cầu thời gian giữa các node trong cụm luôn phải đồng bộ. Vì việc giao tiếp giữa các node, quản lý voting-disk, ASM metadata cần thời gian chính xác và khớp với nhau để hoạt động hiệu quả.

Oracle Linux 7.9 cung cấp một tiến trình nền chronyd thực hiện việc liên tục kiểm tra và điều chỉnh đồng hồ hệ thống. Khởi động và kiểm tra trạng thái process:

# Khởi động service

$ systemctl start chronyd.service

# Enable để service này tự khởi động lại khi server reboot

$ systemctl enable chronyd.service

# Kiểm tra trạng thái của service

$ systemctl status chronyd.service

### Format đĩa, cấu hình ASM

* Tải và giải nén bộ cài GRID

Gói cài này là bộ phần mềm Oracle Grid Infrastructure 19c (19.3.0.0.0) dành cho Linux x86‑64. Bên trong gói này bao gồm các file nhị phân và tài nguyên cần thiết để cài đặt và vận hành cơ sở hạ tầng lưới (Grid Infrastructure) của Oracle, trong đó có các thành phần chính như ASM, ACFS, Oracle Restart, GNS, và các công cụ hỗ trợ như ASMCA, CVU, SRVCTL, CRSCTL,...

Tải gói cài trên trang chính thức của Oracle: [edelivery.oracle.com](https://edelivery.oracle.com)

Giải nén vào GRID\_HOME đã tạo trước đó:

$ unzip V982068-01.zip -d /u01/app/19.0.0/grid

* Cấu hình Oracle ASM Library

Oracle ASMLib (ASM Library) là một bộ công cụ phần mềm được Oracle cung cấp, dùng để hỗ trợ hệ điều hành Linux trong việc quản lý ổ đĩa ASM (Automatic Storage Management). ASM Library có thể tạo, quản lý và duy trì ổ đĩa ASM một cách ổn định. Để cấu hình thông tin cho ASMLib, sử dụng lệnh:

# Lệnh cấu hình ASMLib

$ /usr/sbin/oracleasm configure -i

# Hệ thống hiển thị yêu cầu nhập thông tin. Chọn grid làm user quản trị. Chọn asmadmin làm group quản trị. Chọn y/y để tự khởi động driver và quét ổ đĩa mỗi lần reboot

Default user to own the driver interface []: grid

Default group to own the driver interface []: asmadmin

Start Oracle ASM library driver on boot (y/n) [n]: y

Scan for Oracle ASM disks on boot (y/n) [y]: y

Writing Oracle ASM library driver configuration: done

Sau khi cấu hình xong, kích hoạt ASMLib để sử dụng

$/usr/sbin/oracleasm init

* Định dạng ASM cho ổ đĩa

CVU (Cluster Verification Utility) là công cụ dùng để kiểm tra môi trường hệ thống trước và sau khi cài đặt Oracle Grid Infrastructure hoặc Oracle RAC. Để làm được, CVU cần truy cập các ổ đĩa mà ASM sẽ sử dụng. Tuy nhiên, mặc định Linux không cho phép truy cập cấp thấp như vào đĩa thô (raw device). Do đó cvuqdisk-1.0.10-1.rpm sẽ cung cấp một driver hỗ trợ CVU nhìn thấy và thao tác với thiết bị đĩa thô này.

$ cd $GRID\_HOME/cv/rpm

$ sudo rpm -Uvh cvuqdisk-1.0.10-1.rpm

ASMLib (Oracle ASM Library) chỉ hỗ trợ đánh dấu (stamp) các thiết bị đã được khởi tạo phân vùng, chứ không làm việc trực tiếp với ổ đĩa nguyên (raw) nên cần phải tạo phân vùng cho đĩa:

# Đĩa sda là đĩa riêng cho từng máy chủ, lưu trữ thông tin của node đó. Còn 2 đĩa sdb và sdc là 2 đĩa được chia sẻ trên các node.

# fdisk /dev/sdb

*Command (m for help): n*

*Select (default p): p*

*Partition number (1 -4, default 1): <enter>*

*First sector (63 -62914559, default 63): <enter>*

*Last sector, +sectors or +size{K,M,G} (63 -62914559, default 62914559): <enter>*

*Partition 1 of type Linux and of size 60 GiB is set*

*w -->to save*

# fdisk /dev/sdc

*Command (m for help): n*

*Select (default p): p*

*Partition number (1 -4, default 1): <enter>*

*First sector (63 -51714559, default 63): <enter>*

*Last sector, +sectors or +size{K,M,G} (63 -51714559, default 62914559): <enter>*

*Partition 1 of type Linux and of size 50 GiB is set*

*w -->to save*

Format đĩa thô sang dạng ASM

*#* oracleasm createdisk DATA1 /dev/sdb1

# oracleasm createdisk FRA1 /dev/sdc1

Lệnh này sẽ đánh dấu (mark) ba phân vùng /dev/sdb1, /dev/sdc1 là các ASM disks với tên nhãn tương ứng DATA1, FRA1, để Oracle ASM Library (ASMLib) có thể quản lý và cung cấp chúng cho các ASM diskgroup. Khi thực thi, ASMLib sẽ ghi metadata lên mỗi phân vùng, tạo device node tại /dev/oracleasm/disks/<label>, từ đó cho phép ASM và các công cụ như CVU truy cập và xác minh chúng, cũng như sử dụng trong các diskgroup như +DATA, +FRA

### Cài đặt Oracle Grid Infrastructure

Trước khi cài đặt Orcle Grid Infrastructure, Oracle yêu cầu thiết lập kết nối SSH không mật khẩu để có thể gửi lệnh cài đặt qua lại giữa 2 máy chủ và thiết lập được đường Interconnect. Các bước thiết lập như sau:

# Chạy trên cả 2 nodes bằng user grid

$ ssh-keygen -t rsa

# Chạy trên node 1:

$ cat .ssh/id\_rsa.pub | ssh lab-oracle-01 'cat >> .ssh/authorized\_keys'

# Chạy trên node 2:

$cat .ssh/id\_rsa.pub | ssh lab-oracle-02 'cat >> .ssh/authorized\_keys'

# Chạy trên node 1:

cat .ssh/id\_rsa.pub | ssh lab-oracle-02 'cat >> .ssh/authorized\_keys'

# Chạy trên node 2:

cat .ssh/id\_rsa.pub | ssh lab-oracle-01 'cat >> .ssh/authorized\_keys'

Kiểm tra SSH sau khi thiết lập

Đứng từ user grid ở node 1:

ssh lab-oracle-02 thông sang user grid node 2 mà không cần nhập pass => thành công

Đứng từ grid node 2:

ssh lab-oracle-01 thông sang user grid node 1 không cần nhập pass => thành công

Việc cài đặt Oracle Grid Infrastructure sẽ có 2 cách là silent install và GUI. Với cách silent install thì thông tin cấu hình sẽ nằm trong response file. Chọn cách response file để đơn giản hóa việc cài đặt. Tạo response file:

$ vi $GRID\_HOME/install/response/gridsetup.rsp

oracle.install.responseFileVersion=/oracle/install/rspfmt\_crsinstall\_response\_schema\_v19.0.0

oracle.install.option=CRS\_CONFIG

INVENTORY\_LOCATION=/u01/app/oraInventory

ORACLE\_BASE=/u01/app/oracle

oracle.install.asm.OSDBA=asmdba

oracle.install.asm.OSOPER=asmoper

oracle.install.asm.OSASM=asmadmin

oracle.install.crs.config.gpnp.scanName=lab-oracle-scan

oracle.install.crs.config.gpnp.scanPort=1521

oracle.install.crs.config.clusterName=lab-oracle-cluster

oracle.install.crs.config.autoConfigureClusterNodeVIP=false

oracle.install.crs.config.clusterNodes=lab-oracle-1:lab-oracle-1-vip,lab-oracle-2:lab-oracle-2-vip

oracle.install.crs.config.networkInterfaceList=eth0:172.16.30.0:1,eth1:172.11.90.0:5

oracle.install.crs.config.storageOption=ASM\_STORAGE

oracle.install.asm.SYSASMPassword=oracle

oracle.install.asm.diskGroup.name=DATA

oracle.install.asm.diskGroup.redundancy=EXTERNAL

oracle.install.asm.diskGroup.AUSize=4

oracle.install.asm.diskGroup.disks=/dev/oracleasm/disks/DATA

oracle.install.asm.diskGroup.diskDiscoveryString=/dev/oracleasm/disks/\*

oracle.install.asm.additionalDiskGroup.name=FRA

oracle.install.asm.additionalDiskGroup.redundancy=EXTERNAL

oracle.install.asm.additionalDiskGroup.AUSize=4

oracle.install.asm.additionalDiskGroup.disks=/dev/oracleasm/disks/FRA

oracle.install.crs.rootconfig.executeRootScript=true

Chạy silent install:

$ ./gridSetup.sh -silent -responseFile $GRID\_HOME/install/response/gridsetup.rsp

Sau khi quá trình cài đặt xong, kiểm tra trạng thái các service bằng user grid với lệnh status. Nếu các serviec đang ONLINE tức là đã hoạt động:

$ crsctl status res -t

### Cài đặt Database Instance

Tương tự khi cài đặt Oracle Grid Infrastructure, việc cài đặt Oracle Database cũng yêu cầu thiết lập kết nối SSH không mật khẩu Các bước thiết lập như sau:

# Chạy trên cả 2 nodes bằng user oracle

$ ssh-keygen -t rsa

# Chạy trên node 1:

$ cat .ssh/id\_rsa.pub | ssh lab-oracle-01 'cat >> .ssh/authorized\_keys'

# Chạy trên node 2:

$cat .ssh/id\_rsa.pub | ssh lab-oracle-02 'cat >> .ssh/authorized\_keys'

# Chạy trên node 1:

cat .ssh/id\_rsa.pub | ssh lab-oracle-02 'cat >> .ssh/authorized\_keys'

# Chạy trên node 2:

cat .ssh/id\_rsa.pub | ssh lab-oracle-01 'cat >> .ssh/authorized\_keys'

Kiểm tra SSH sau khi thiết lập

Đứng từ user oracle ở node 1:

ssh lab-oracle-02 thông sang user oracle node 2 mà không cần nhập pass => thành công

Đứng từ oracle node 2:

ssh lab-oracle-01 thông sang user oracle node 1 không cần nhập pass => thành công

Và việc cài đặt Oracle Database cũng giống Oracle Grid Infrastructure sẽ có 2 cách là silent install và GUI. Chọn cách response file để đơn giản hóa việc cài đặt. Tạo response file:

$ vi $ORACLE \_HOME/install/response/db\_install.rsp

oracle.install.responseFileVersion=/oracle/install/rspfmt\_dbinstall\_response\_schema\_v19.0.0

oracle.install.option=INSTALL\_DB

# Hostname của node

ORACLE\_HOSTNAME=lab-oracle-1, lab-oracle-2

# Nhóm OS cho inventory

UNIX\_GROUP\_NAME=oinstall

# Oracle Inventory

INVENTORY\_LOCATION=/u01/app/oraInventory

SELECTED\_LANGUAGES=en

# Đường dẫn ORACLE\_HOME/BASE

ORACLE\_BASE=/u01/app/oracle

ORACLE\_HOME=/u01/app/oracle/product/19.0.0/dbhome\_1

# Edition, nhóm OSDBA/OSOPER/…

oracle.install.db.OSDBA\_GROUP=dba

oracle.install.db.OSBACKUPDBA\_GROUP=dba

oracle.install.db.OSDGDBA\_GROUP=dba

oracle.install.db.OSKMDBA\_GROUP=dba

oracle.install.db.OSRACDBA\_GROUP=dba

# Cập nhật bảo mật

DECLINE\_SECURITY\_UPDATES=true # Không gửi email update

Chạy silent install:

$ ./runInstaller -silent -responseFile $ORACLE\_HOME/install/response/db\_install.rsp

Sau khi quá trình cài đặt thành công, tiếp tục kiểm tra trạng thái các service của cơ sở dữ liệu dưới sự quản lý của Oracle Grid Infrastructure bằng user grid với lệnh status. Nếu các serviec đang ONLINE tức là đã hoạt động:

$ crsctl status res -t

Hoàn thành quá trình cài đặt Oracle RAC, các service và cơ sở dữ liệu đã hoạt động bình thường.

## Triển khai Data Guard kết hợp với Oracle RAC

### Kiến trúc tổng quan

A diagram of a data center

AI-generated content may be incorrect.

###### Hình 3.2: Kiến trúc tổng quan thực nghiệm giải pháp Data Guard kết hợp Oracle RAC

Với mục đích là dự phòng, đảm bảo an toàn dữ liệu. Nên 2 hệ thống chính và phụ sẽ đặt trải dài trên lãnh thổ địa lý đó, cung cấp nền tảng mạng và tài nguyên cho các ứng dụng. Mỗi khu vực chứa các hạ tầng công nghệ thông tin này hoàn toàn độc lập về mặt giao tiếp mạng cũng như về vị trí địa lý với các khu vực khác.

### Môi trường Oracle Net và định danh CSDL

Trong môi trường Data Guard, để phân biệt về loại CSDL (dự phòng hoặc chính), cần sử dụng DB\_UNIQUE\_NAME để hệ thống nhận diện, giám sát các CSDL này, thay vì sử dụng DB\_NAME.

A diagram of a company

AI-generated content may be incorrect.

Hình 3.3: Minh họa phân cấp tên trong hệ thống CSDL

Cả CSDL chính (primary) và dự phòng (standby) đều dùng Listener để giao tiếp qua Oracle Net. Khi primary cần gửi bản ghi (redo data) sang standby, nó sẽ kết nối tới Listener của standby qua TCP (mặc định port 1521). Listener cũng hỗ trợ IPC cho các kết nối nội bộ trên cùng máy chủ.

Trong Data Guard đơn giản hóa chuỗi kết nối bằng cách dùng Local Naming Method. Khai báo alias (định nghĩa host, port, SID) trong tnsnames.ora, rồi khi kết nối chỉ cần gọi CONNECT user/pass@alias thay vì gõ đầy đủ chuỗi.

CONNECT username/password@LABORACLE

# LABORACLE chứa thông tin IP, port, instance cần kết nối

Bước đầu trong quá trình triển khai kiến trúc Data Guard là cấu hình Listener và Local Naming cho mỗi hệ thống CSDL. Tại cả hai máy chủ CSDL, thực hiện việc cấu hình Listener có dạng như sau:

<listener\_name>=

(DESCRIPTION\_LIST =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = lab-oracle-scan)(PORT = 1521))

(ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))

))

SID\_LIST\_<listener\_name>=

(SID\_LIST =

(SID\_DESC = (GLOBAL\_DBNAME = laboracle) (ORACLE\_HOME =

/u01/app/oracle/product/19.0.0/dbhome\_1) (SID\_NAME = laboracle)))

Trong đó:

*<listener\_name>*: tên của Listener

*DESCRIPTION\_LIST*: chứa danh sách mô tả các kết nối đến mà Listener sẽ xử lý. Giao thức TCP dành cho kết nối từ các ứng dụng và giữa các Database với nhau. Còn giao thức ICP (Inter-Process Communication) dành cho các ứng dụng cùng trên máy chủ chứa CSDL có thể kết nối nội bộ với nhau.

*SID\_LIST\_<listener\_name>:* chứa danh sách mô tả các CSDL mà Listener sẽ điều hướng kết nối của người dùng tới CSDL đó.

Cấu hình Local Naming Method cho hai máy chủ chứa CSDL có dạng như sau:

|  |
| --- |
| <alias\_primary\_name> =  (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = lab-oracle-scan)(PORT =1521)) (CONNECT\_DATA = (SERVER = DEDICATED) (SERVICE\_NAME=laboracle))) |

Trong đó:

*<alias\_primary\_name>:* tên bí danh được dùng để mô tả kết nối

*DESCRIPTION*: mô tả kết nối, gồm địa chỉ và dữ liệu kết nối

*CONNECT\_DATA:* chỉ định phương thức kết nối là DEDICATED – mỗi client kết nối vào sẽ có một tiến trình nền hỗ trợ riêng và kết nối vào CSDL với Service\_name là laboracle.

### Cấu hình tham số chung cho hệ thống chính

*Bật chế độ FORCE LOGGING:* Trong chế độ FORCE LOGGING, mọi thay đổi tại CSDL được lưu trong Redo Buffer Cache đều được ghi xuống thiết bị đĩa cứng tại Online Redo Logs theo cơ chế xoay vòng (ghi đè khi hết) bằng tiến trình LGWr, giúp CSDL có thể khôi phục được những thay đổi đã COMMIT sau khi xảy ra sự cố.

*Bật chế độ ARCHIVELOG:* Khi một Online Redo Logs đầy, chế độ ARCHIVELOG sẽ giúp lưu trữ tệp tin này bằng tiến trình ARCn với điều kiện Online Redo Logs thực hiện cơ chế Log Switch để thực hiện chuyển qua tệp tin Online Redo Logs khác. Trạng thái của Online Redo Logs (ORLs) thời điểm này sẽ chuyển từ CURRENT qua ACTIVE cho đến khi được lưu trữ thành công, trở về trạng thái INACTIVE.

# Truy vấn thông tin cấu hình CSDL

SELECT NAME, DB\_UNIQUE\_NAME, OPEN\_MODE, LOG\_MODE, FORCE\_LOGGING FROM V$DATABASE;

# Thực hiện bật Force Logging, ArchiveLog và Flashback

SHUTDOWN IMMEDIATE;

STARTUP MOUNT;

SHOW PARAMETER NAME;

ALTER DATABASE ARCHIVELOG;

ALTER DATABASE FORCE LOGGING;

*Tạo Standby Redo Logs:* Standby Redo Logs (SRLs) dùng khi vai trò của CSDL là dự phòng/phụ. Cần tạo SRLs ở cả hai CSDL chính và phụ trong việc chuyển đổi vai trò. Điều kiện bắt buộc khi tạo SRLs như sau: nhiều hơn 01 groups so với ORLs tại CSDL chính (1), size của SRLs cần lớn hơn hoặc bằng ORLs của CSDL chính (2). Nếu cấu hình sai, tiến trình RFS sẽ ghi vào Archive Redo Log (ARL), mất đi tính năng Real-Time Apply và gây ra hiện tượng trễ.

# Kiểm tra dung lượng ORLs theo Megabyte (200MB mỗi ORLs) select GROUP#,THREAD#,SEQUENCE#,bytes/1024/1024, MEMBERS,STATUS from v$log;

# Tạo Standby Redo Logs phù hợp

ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 GROUP 4 ('+DATA/') SIZE 200M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 GROUP 5 ('+DATA/') SIZE 200M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 GROUP 6 ('+DATA/') SIZE 200M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 GROUP 7 ('+DATA/') SIZE 200M;

# Kiểm tra lại các loại Logs hiện tại

SELECT TYPE, MEMBER FROM V$LOGFILE ORDER BY GROUP#;

*Thiết lập vị trí lưu trữ Redo Log cục bộ:* Trong môi trường Data Guard, Redo Transport Services được cài đặt, điều khiển bằng tham số LOG\_ARCHIVE\_DEST\_n. Tham số này cho phép redo data vừa được gửi đồng bộ sang Standby Database, vừa được lưu trữ xuống đĩa. Cụ thể:

LOG\_ARCHIVE\_DEST\_N: [1 | 2 | 3 | … | 31] =

‘LOCATION = path\_name | SERVICE = service\_name

SYNC | ASYNC

AFFIRM | NOAFFIRM

VALID\_FOR = (redo\_log\_type, database\_role) DB\_UNIQUE\_NAME = db\_unique\_name’

LOG\_ARCHIVE\_DEST\_N là thông tin xác định Redo Transport Services sẽ chuyển redo data xuống cục bộ hay đi sang Standby Redo Logs. n luôn phải đặt là 1 khi lưu redo cục bộ, LOCATION sẽ được đặt giá trị là một đường dẫn của máy chủ cài đặt CSDL chính.

Cách truyền phụ thuộc vào kiểu truyền SYNC/ASYNC và AFFIRM /NOAFFIRM, mặc định khi thiết lập AFFIRM thì sẽ thiết lập SYNC. Với tham số VALID\_FOR gồm hai đối số đầu vào, khi CSDL có vai trò là *database\_role* thì sẽ lưu trữ *redo\_log\_type* xuống hoặc gửi redo\_log\_type đi cho CSDL dự phòng. Trong cài đặt cục bộ, thông tin cài đặt sẽ như sau:

ALTER SYSTEM SET LOG\_ARCHIVE\_DEST\_1=

'LOCATION=+FRA

VALID\_FOR=(ALL\_LOGFILES,ALL\_ROLES)

DB\_UNIQUE\_NAME=laboracle' scope=spfile;

Với cài đặt này, dù CSDL ở vai trò chính hoặc vai trò phụ, các redo data được lưu trữ trong ORLs hoặc SRLs đều được sao chép và cất giữ theo đường dẫn đã cấu hình tại LOCATION.

Ngoài ra, còn một số tham số phụ trợ khác cho CSDL chính được cấu hình như sau:

# Cấu hình file mật khẩu chỉ sử dụng trong phạm vi máy chủ cài đặt Database

ALTER SYSTEM SET REMOTE\_LOGIN\_PASSWORDFILE =EXCLUSIVE SCOPE=SPFILE;

# Cấu hình quản lý các tệp tin đồng thời của các CSDL

ALTER SYSTEM SET STANDBY\_FILE\_MANAGEMENT =AUTO SCOPE=SPFILE;

### Tạo hệ thống dự phòng dựa trên RMAN DUPLICATE

*Sao chép và gửi file mật khẩu, tham số từ CSDL chính sang CSDL dự phòng:*

Trong kiến trúc Data Guard cần sử dụng chung một mật khẩu cho người dùng quản trị SYS. Cụ thể, việc sao chép từ máy chủ chính sang máy chủ phụ được thực hiện bằng cú pháp sau:

scp [other options] [source username@IP]:/[full file name] [destination username@IP]:/[directory]

Tệp tin mật khẩu có tên là *orapwlaboracle* và tệp tin tham số có tên là *initoracle.ora* đều nằm ở thư mục *$ORACLE\_HOME/dbs*. Lệnh sao chép là:

scp initoracle.ora [oracle@172.16.30.70:$ORACLE\_HOME/dbs](mailto:oracle@172.16.30.70:$ORACLE_HOME/dbs) scp orapworacle oracle@172.16.30.70:$ORACLE\_HOME/dbs

*Dựng CSDL dự phòng bằng RMAN DUPLICATE:*

Công cụ RMAN tạo CSDL dự phòng bằng cách nhân bản các tệp tin được sử dụng ở CSDL chính mà CSDL chính vẫn diễn ra bình thường.

CSDL chính gọi là TARGET, CSDL nhận và khôi phục bản sao chép gọi là AUXILIARY. Các dữ liệu được RMAN sao chép như: các datafiles hệ thống, control files, undo datafiles và tệp tin tham số cấu hình Instance của CSDL (spfile).

rman TARGET sys/oracle@LABORACLE AUXILIARY sys/oracle@STANDBY

DUPLICATE TARGET DATABASE

FOR STANDBY

FROM ACTIVE DATABASE

DORECOVER

SPFILE

SET db\_unique\_name = 'standby' COMMENT 'IS STANDBY'

SET log\_archive\_dest\_1 = 'LOCATION=+FRA

VALID\_FOR=(ALL\_LOGFILES,ALL\_ROLES)

DB\_UNIQUE\_NAME=standby' COMMENT 'IS STANBY' NOFILENAMECHECK

Ý nghĩa của các cài đặt trong RMAN DUPLICATE như sau:

- *FOR STANDBY:* Giữ nguyên DBID của primary để tạo standby cùng môi trường Data Guard (không sinh DBID mới).

* *FROM ACTIVE DATABASE:* Sao chép data files trực tiếp từ primary đang chạy, không cần tắt database.
* *DORECOVER:* Tự động áp dụng các redo logs (archived & online) sau khi nhân bản, nếu không, database standby chỉ có thể mở ở trạng thái MOUNT.
* *SPFILE:* copy và tùy chỉnh tệp tham số (SPFILE) từ primary sang standby.
* *NOFILENAMECHECK:* Bỏ qua kiểm tra trùng tên/đường dẫn file, khi cấu trúc thư mục giữa primary và standby giống nhau.

### Cấu hình môi trường Data Guard

Cấu hình Redo Transport Services đối với CSDL chính:

Cũng có thể nói đây là việc "Thiết lập vị trí lưu Redo Log" nhưng ở phạm vi toàn cục – gửi redo data sang CSDL dự phòng. Sử dụng Local Naming Method cho tham số SERVICE thay vì LOCATION với đường dẫn để lưu xuống như thông thường.

ALTER SYSTEM SET LOG\_ARCHIVE\_DEST\_2=

'SERVICE=STANDBY ASYNC

VALID\_FOR=(ALL\_LOGFILES,PRIMARY\_ROLE)

DB\_UNIQUE\_NAME=standby' SCOPE=SPFILE;

Khai báo CSDL chính, dự phòng trong môi trường Data Guard:

Thông qua tham số LOG\_ARCHIVE\_CONFIG, liệt kê CSDL chính và CSDL dự phòng bằng tham số con DG\_CONFIG.

DG\_CONFIG: xác định danh sách gồm CSDL chính và các CSDL dự phòng được nhận redo data

Cấu hình dành cho hai CSDL thực hiện trong bài sẽ như sau:

ALTER SYSTEM SET LOG\_ARCHIVE\_CONFIG='DG\_CONFIG=(laboracle,standby)';

ALTER SYSTEM SET LOG\_ARCHIVE\_DEST\_STATE\_1=ENABLE;

ALTER SYSTEM SET LOG\_ARCHIVE\_DEST\_STATE\_2=ENABLE;

Cấu hình tiến trình xử lý trễ chủ động FAL cho vai trò CSDL dự phòng:

Cả hai tiến trình con này đều được sử dụng cho CSDL dự phòng trong việc chủ động xử lý thiếu dữ liệu thay vì bị động với tiến trình ARCn của CSDL chính.

# CSDL chính

ALTER SYSTEM SET FAL\_CLIENT='LABORACLE';

ALTER SYSTEM SET FAL\_SERVER='STANDBY';

# CSDL dự phòng

ALTER SYSTEM SET FAL\_CLIENT='STANDBY';

ALTER SYSTEM SET FAL\_SERVER='LABORACLE';

Thiết lập chế độ ưu tiên hiệu năng sau khi đã thiết lập Redo Transport Services:

ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE PERFORMANCE;

Khởi động Redo Log Apply – tiến trình MRP trên CSDL dự phòng:

Khởi động tiến trình MRP cho CSDL dự phòng để CSDL có thể bắt đầu áp dụng các redo data nhận được từ SRLs cũng như Archived Redo Logs.

# Sử dụng MRP với Real-Time Apply

ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURENT LOGFILE DISCONNECT;

# Dừng hoạt động tiến trình Redo Apply

ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;

### Xác định chế độ bảo vệ trong Data Guard

Trong 03 chế độ bảo vệ của Data Guard, cần lựa chọn chế độ để phù hợp với yêu cầu vận hành của doanh nghiệp. Các chế độ có độ ưu tiên khác nhau về hiệu năng, tính sẵn sàng của hệ thống và mức độ mất mát dữ liệu.

*Với chế độ Ưu tiên bảo vệ (max. protection):*

Chế độ này thực hiện cơ chế chỉ xác nhận một giao dịch đã được COMMIT khi và chỉ khi ít nhất một CSDL dự phòng trả lại tín hiệu cho CSDL chính rằng dữ liệu thay đổi đã được ghi vào CSDL dự phòng. Ngược lại, nếu không có bất kỳ tín hiệu nào trở về, CSDL chính sẽ treo và dừng hoạt động để đảm bảo tính toàn vẹn khi giao dịch chưa được COMMIT ở cả hai CSDL.

*Với chế độ Ưu tiên tính sẵn sàng (max. availability):*

Trong chế độ này, CSDL chính sẽ chờ đến thời gian tối đa được cấu hình trong biến NET\_TIMEOUT khi chờ tín hiệu phản hồi lại từ CSDL dự phòng, nếu nhận được tín hiệu, CSDL chính có thể đánh dấu COMMIT và tiếp tục một giao dịch mới. Ngược lại, CSDL chính sẽ hoạt động như chế độ Ưu tiên hiệu năng và liên tục cập nhật trạng thái của CSDL dự phòng.

*Với chế độ Ưu tiên hiệu năng (max. performance):*

Chế độ này COMMIT ngay khi có tín hiệu của người dùng, ghi redo data vào Online Redo Log. Hệ thống Data Guard sẽ truyền dữ liệu thay đổi song song tới: Standby Redo Log của CSDL dự phòng, trực tiếp từ Online Redo Logs (đối với đường truyền tốt) (1), tới Archive Redo Log của CSDL chính (2) theo cơ chế không đồng bộ với giao dịch được COMMIT, hạn chế việc mất mát dữ liệu khi CSDL chính xảy ra sự cố.

Chế độ này phù hợp khi không đặt nặng vấn đề về mất đồng bộ dữ liệu và yêu cầu hiệu năng hệ thống chính cần hoạt động với hiệu năng cao.

## Tổng kết chương 3

Chương 3 đã trình bày chi tiết quá trình triển khai mô hình Oracle RAC kết hợp với giải pháp Data Guard trên hệ thống máy chủ ảo. Nội dung chương bao gồm các bước cấu hình, cài đặt và đánh giá hiệu quả hoạt động của mô hình.

Cụ thể, chương tập trung vào cấu hình máy chủ và cài đặt hai mô hình Oracle RAC và Data Guard, tiếp theo là quá trình triển khai Oracle RAC từ kiến trúc tổng quan, cấu hình mạng, chuẩn bị môi trường cho đến cài đặt Grid Infrastructure và Database Instance. Cuối cùng, chương mô tả cách tích hợp Data Guard với Oracle RAC, bao gồm thiết lập Oracle Net, đồng bộ dữ liệu, cấu hình RMAN, kiểm tra hoạt động và xác định chính sách bảo vệ dữ liệu. Qua đó, chương giúp đánh giá được khả năng đảm bảo tính sẵn sàng cao và an toàn dữ liệu trong hệ thống Oracle.

# KẾT LUẬN

## Kết quả đóng góp

Sau khi thực nghiệm triển khai giải pháp kết hợp Oracle RAC và Data Guard dựa trên CSDL dự phòng dạng vật lý, đề tài đạt được một số kết quả đóng góp như sau:

*Thứ nhất*, các tài nguyên (resource) của Oracle RAC đều đang hoạt động:

* Cột đầu tiên: đây là tên các serviec đang chạy dưới sự quản lý của Oracle Grid Infrastructure như ora.DATA.dg, ora.FRA.dg, ora.LISTENER.SCAN, ora.cvu, ora.laboracle.db, ora.scan, ora-vip.
* Cột thứ 2: Tình trạng mục tiêu mà Oracle Grid Infrastructure muốn resource đó đạt được.
* Cột thứ 3: Tình trạng hiện tại của service, phải duy trì trạng thái hiện tại giống trạng thái mục tiêu ở cột 2.
* Cột thứ 4: Tên máy chủ trong cụm mà resource đang chạy trên đó.
* Cột thứ 5: Trạng thái tổng thể của resource trên node đó:
  + STABLE: trạng thái ổn định, không thay đổi gần đây
  + Started,STABLE: resource đang chạy và ở trạng thái ổn định
  + Open: đối với database, có nghĩa là DB đã được mở (open)

A screenshot of a computer screen

AI-generated content may be incorrect.

###### Hình 23: Thông tin các tài nguyên trong Oracle RAC

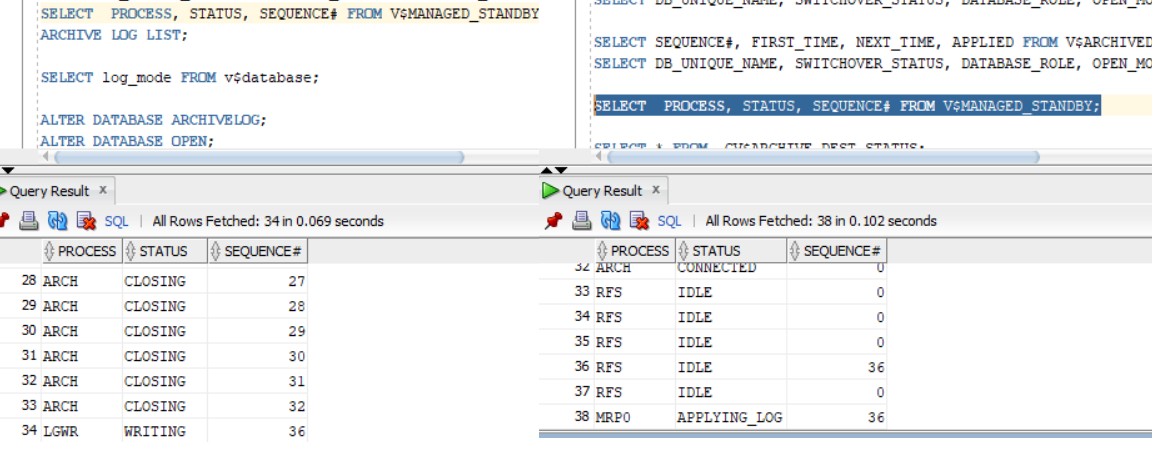
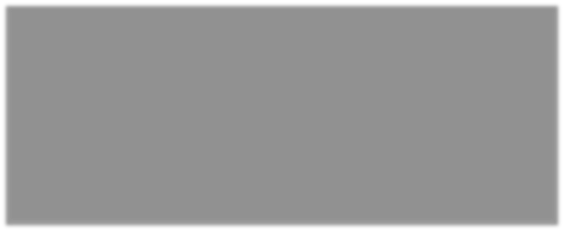
*Thứ hai*, các tiến trình quan trọng thuộc Log Apply Services và Log Transport Services hoạt động ổn định trong việc truyền và áp dụng thông tin thay đổi từ CSDL chính sang CSDL dự phòng theo chế độ Real-Time Apply, gồm cả các tiến trình xử lý trễ dữ liệu tự động:

Bảng v$Managed\_Standby dùng để truy vấn thông tin các tiến trình thuộc môi trường Data Guard:

− Cột “Process” thể hiện các tiến trình của CSDL chính: LGWR – ghi redo data xuống Online Redo Logs, ARCH – lưu trữ Online Redo Logs thành Archive Redo Logs, RFS – nhận và lưu trữ redo data từ CSDL chính vào Standby Redo Logs và MRP – áp dụng các redo data.

− Cột “Status” cho biết trạng thái hoạt động của tiến trình: CLOSING – hoàn thành việc lưu trữ, WRITING – ghi redo data, IDLE – không hoạt động và APPLYING\_LOG – áp dụng thay đổi.

− Cột “SEQUENCE#” hiển thị thông tin thứ tự của Redo Logs đang được sử dụng.



###### Hình 24: Các tiến trình thuộc hai CSDL trong Data Guard

Bảng v$Archive\_Log cho biết thông tin các tệp Archive Redo Log. Để tìm thông tin về các Logs đã áp dụng, sử dụng bảng v$Log\_History. Khi kết hợp hai bảng này với nhau, thông qua hiệu số của giá trị Log Sequence lớn nhất thuộc Archive Redo

Log đã nhận và Log đã áp dụng, người quản trị sẽ biết được độ trễ hiện tại trong quá trình đồng bộ.

*Thứ ba,* hoàn thành xây dựng các kịch bản kiểm thử khi gặp sự cố bao gồm 3 kịch bản:

* Lỗi đồng bộ dữ liệu giữa Primary và Standby.

Chi tiết: <https://docs.google.com/document/d/1JkFnDq_NJf8nLZFh-ytXC42kG-qJAB-E/edit#heading=h.865kh7gfwq8b>

* Lỗi 1 node trong cụm RAC.

Chi tiết: <https://docs.google.com/document/d/1AthKsvzRB4AE3l79e4EzHwgOaYwNbjAp/edit#heading=h.865kh7gfwq8b>

* Chuyển đổi vai trò khi gặp sự cố không lường trước.

Chi tiết: <https://docs.google.com/document/d/1WVS4GDPs6FmLrkALM8br-MHrKphumpMd/edit#heading=h.865kh7gfwq8b>

## Hạn chế

Hiện tại, hệ thống Standby Database mới chỉ được triển khai theo mô hình single-cluster, tức là chỉ chạy trên một cụm máy chủ duy nhất. Điều này dẫn đến một số hạn chế về khả năng sẵn sàng cao (HA - High Availability) và khả năng mở rộng.

Cụ thể, trong trường hợp xảy ra switchover hoặc failover, Standby Database sẽ chuyển thành Primary Database, tuy nhiên Primary mới chỉ hoạt động trên một node duy nhất thay vì toàn bộ cụm (multi-node cluster) như môi trường Production hiện tại. Điều này không đảm bảo được tính liên tục trong dịch vụ nếu node duy nhất này gặp sự cố, dẫn đến nguy cơ gián đoạn hoạt động.

Thêm vào đó, do không có khả năng phân tán workload hoặc tự động cân bằng tải giữa các node như trong mô hình multi-cluster, hệ thống sau khi switchover sẽ trở nên dễ bị quá tải nếu lưu lượng truy cập tăng cao, đặc biệt trong các khung giờ cao điểm hoặc sự kiện đặc biệt.

## Hướng phát triển

Để khắc phục những hạn chế nêu trên, hướng phát triển trong tương lai là tiến hành triển khai mô hình multi-cluster cho Standby Database.

Cụ thể, Standby Database cũng sẽ được cấu hình trên một cụm RAC (Real Application Cluster) tương tự như môi trường Primary hiện tại, bao gồm nhiều node hoạt động song song. Việc này sẽ giúp:

* Đảm bảo tính sẵn sàng cao (HA) khi switchover/failover xảy ra.
* Cho phép Standby sau khi trở thành Primary có thể tận dụng nhiều node để xử lý, đảm bảo hiệu suất và độ ổn định của hệ thống.
* Tăng khả năng chịu lỗi (fault tolerance) và đảm bảo dịch vụ không bị gián đoạn trong các tình huống khẩn cấp.

Việc triển khai multi-cluster cho standby không chỉ cải thiện đáng kể khả năng dự phòng mà còn giúp hệ thống sẵn sàng hơn cho các hoạt động như DR test, bảo trì định kỳ, hay xử lý các sự cố lớn mà không ảnh hưởng đến hoạt động kinh doanh liên tục của doanh nghiệp.

Bên cạnh việc triển khai mô hình multi-cluster cho Standby Database, một hướng phát triển quan trọng khác là tích hợp hệ thống với Oracle Enterprise Manager Cloud Control (OEM) nhằm nâng cao khả năng giám sát, quản lý và tự động hóa vận hành cơ sở dữ liệu.

* Giám sát tập trung Primary và Standby, cảnh báo sự cố tự động.
* Quản lý switchover/failover thông minh qua giao diện đồ họa.
* Hỗ trợ kiểm thử DR định kỳ, phân tích hiệu suất sau chuyển đổi để tối ưu tài nguyên.

Việc kết hợp mô hình multi-cluster RAC cho Standby Database cùng với công cụ Oracle Enterprise Manager Cloud Control sẽ tạo ra một hệ thống cơ sở dữ liệu có tính sẵn sàng cao, linh hoạt trong quản lý và sẵn sàng cho các tình huống vận hành phức tạp, đảm bảo hoạt động liên tục và hiệu quả cho doanh nghiệp.

# TÀI LIỆU THAM KHẢO

1. Subhash Chandra & Janet Stern (2023). *Real Application Clusters Administration and Deployment Guide.* <https://docs.oracle.com/en/database/oracle/oracle-database/19/racad/real-application-clusters-administration-and-deployment-guide.pdf>
2. Fuller, M. (2014). *Oracle Database 12c: Data Guard Administration*. Joseph Fernandez, Veena Narasimhan.
3. Meeks, J., Carpenter .L & Oracle Corporation . (2006). *Case Study: AmTrust Bank Maximum Availability Architecture – Oracle Database 10g*. Retrieved May 15, 2024, from Oracle.com website: [https://www.oracle.com/us/solutions/amtrustprofile132977.pdf](https://www.oracle.com/us/solutions/amtrustprofile-132977.pdf)
4. K. Keesling, D. & L. Spiller (2014). *Oracle Database 12c: Administration Workshop.* Joseph Fernandez, Veena Narasimhan.
5. Yu, P., Zhou, N. & Sun .H (2011). *The application of Oracle Data Guard in the Logistics Distribution Management Platform*. IEEE. <https://doi.org/10.1109/iccsnt.2011.6182094>
6. Liu Xiu-ju. (2010). *A brief analysis of the disaster recovery backup technology in Oracle database DataGuard*. IEEE. <https://doi.org/10.1109/indusis.2010.5565635>
7. Lääts, M. (2023, July 17). *Cost of Downtime in Manufacturing: Insights & Implications*. Retrieved April 25, 2024, from Evocon website: <https://evocon.com/articles/cost-of-downtime-in-manufacturing-insights-implications/>
8. *Oracle Data Guard Best Practices*. (2019). Retrieved April 24, 2024, from Oracle Help Center website: [https://docs.oracle.com/en/database/oracle/oracledatabase/19/haovw/oracle-data-guard-best-practices.html](https://docs.oracle.com/en/database/oracle/oracle-database/19/haovw/oracle-data-guard-best-practices.html)