



Oracle ActiveDataGuard 监控及性能分析

王光辉

QQ: 83767582

微信: 15147123801



CONTENTS

01 Active DataGuard 架构

02 Active DataGuard 监控

03 Active DataGuard 性能分析

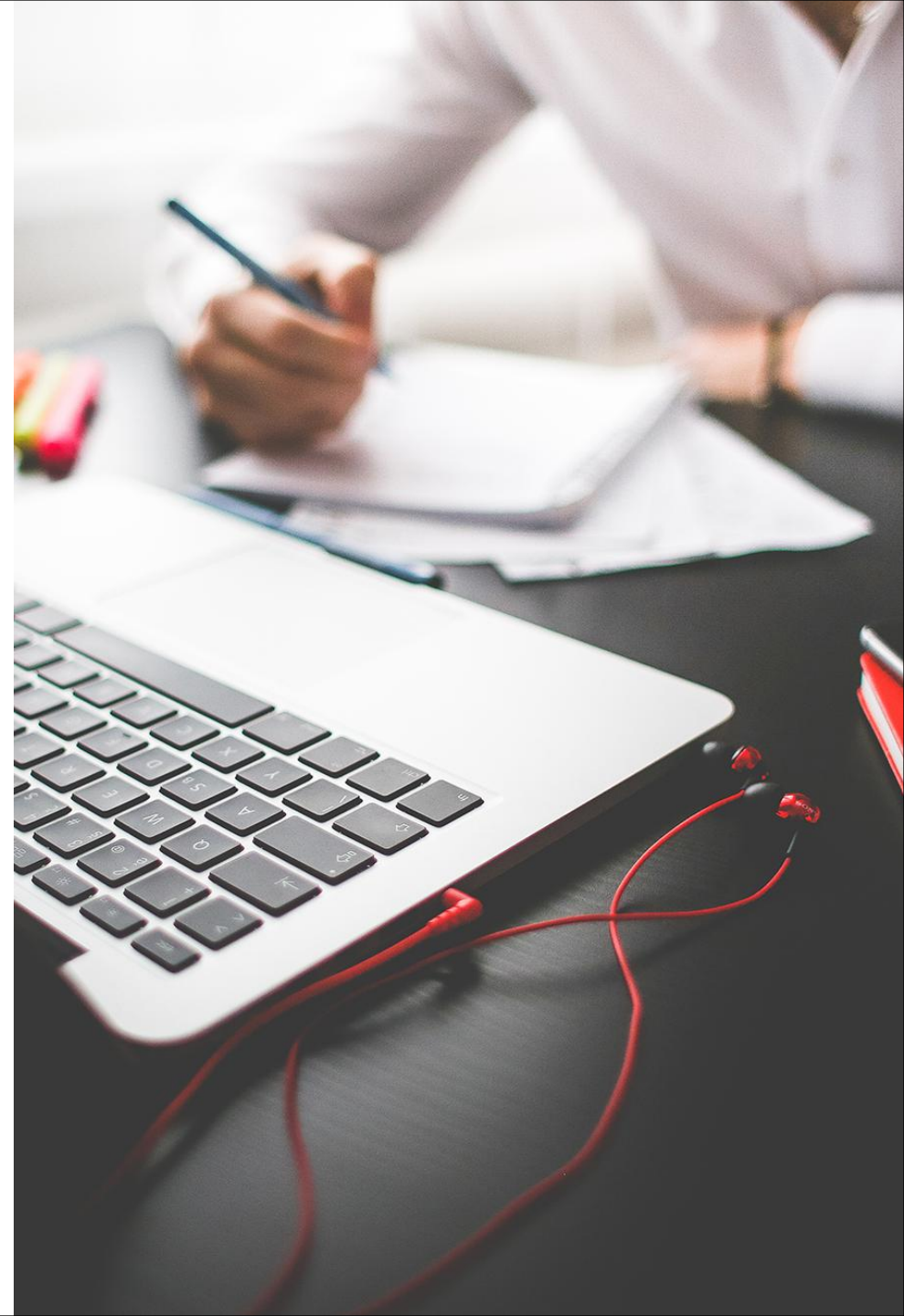
01

Active DataGuard 架构



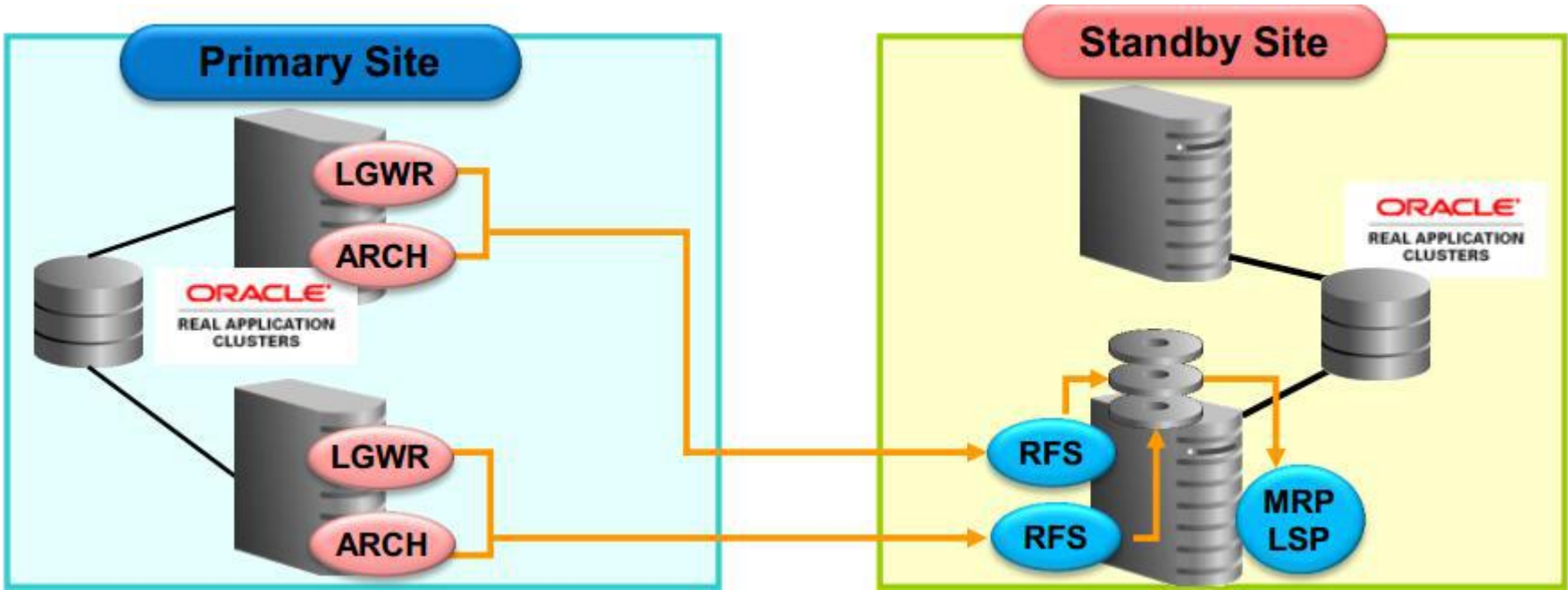
Data Guard 基本架构

- 一个生产库，也称主库
- 两种Standby构筑模式
物理 Standby
逻辑 Standby
- 三种保护模式



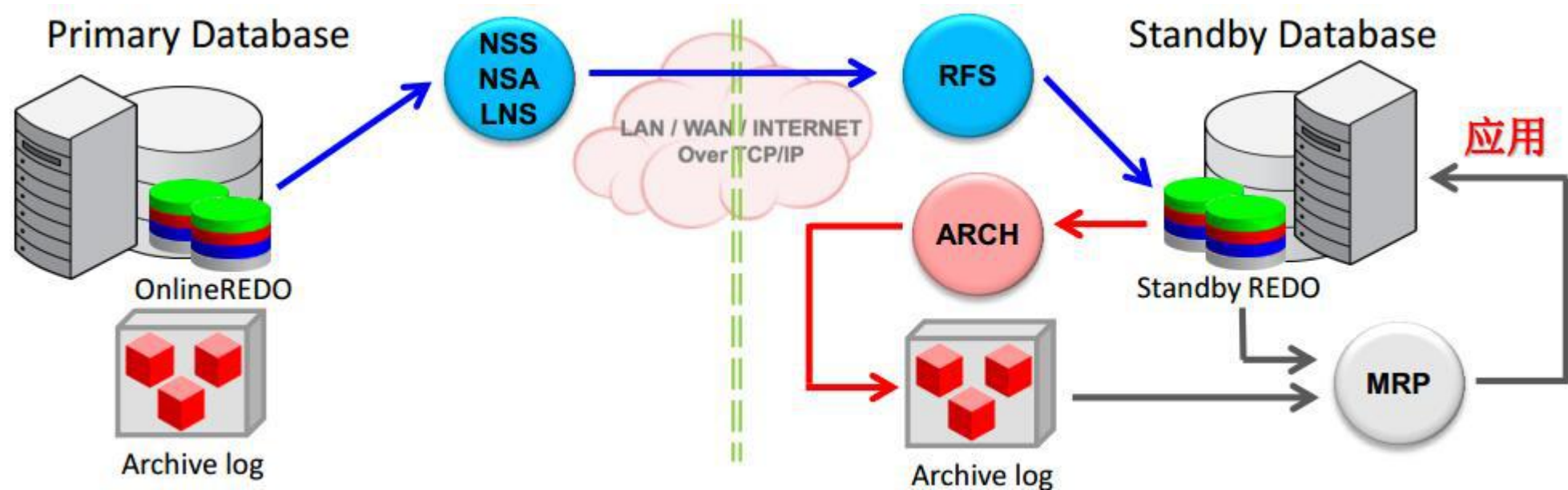
Data Guard 基本架构

- 三种保护模式



保护模式	数据丢失风险	转送	保护模式机制说明
最大保护	零数据丢失	同步(SYNC)	重做记录由日志写入器 (LGWR) 进程从主数据库同步地传输到备用数据库，并且直到确认事务数据在至少一个备用服务器上的磁盘上可用时，才在主数据库上提交事务
最大可用性	有数据丢失可能性	同步/异步	与最大保护模式相同，但如果故障导致主数据库无法写重做流到异地备重做日志时，主数据库不会关闭。替代地，主数据库自动继续以最大可用性模式运行
最大性能	有数据丢失可能性	异步 (ASYNC)	通过允许事务在恢复该事务所需重做数据在写到本地联机重做日志后立即提交而实的。主数据库的重做数据流也写到至少一个备数据库，但是那个重做流相对于创建重做数据的事务是异步写的。（移动现在采用的模式）

Data Guard 基本架构-进程



相关进程 LNS RFS ARCH MRP

Data Guard 基本架构-心跳机制



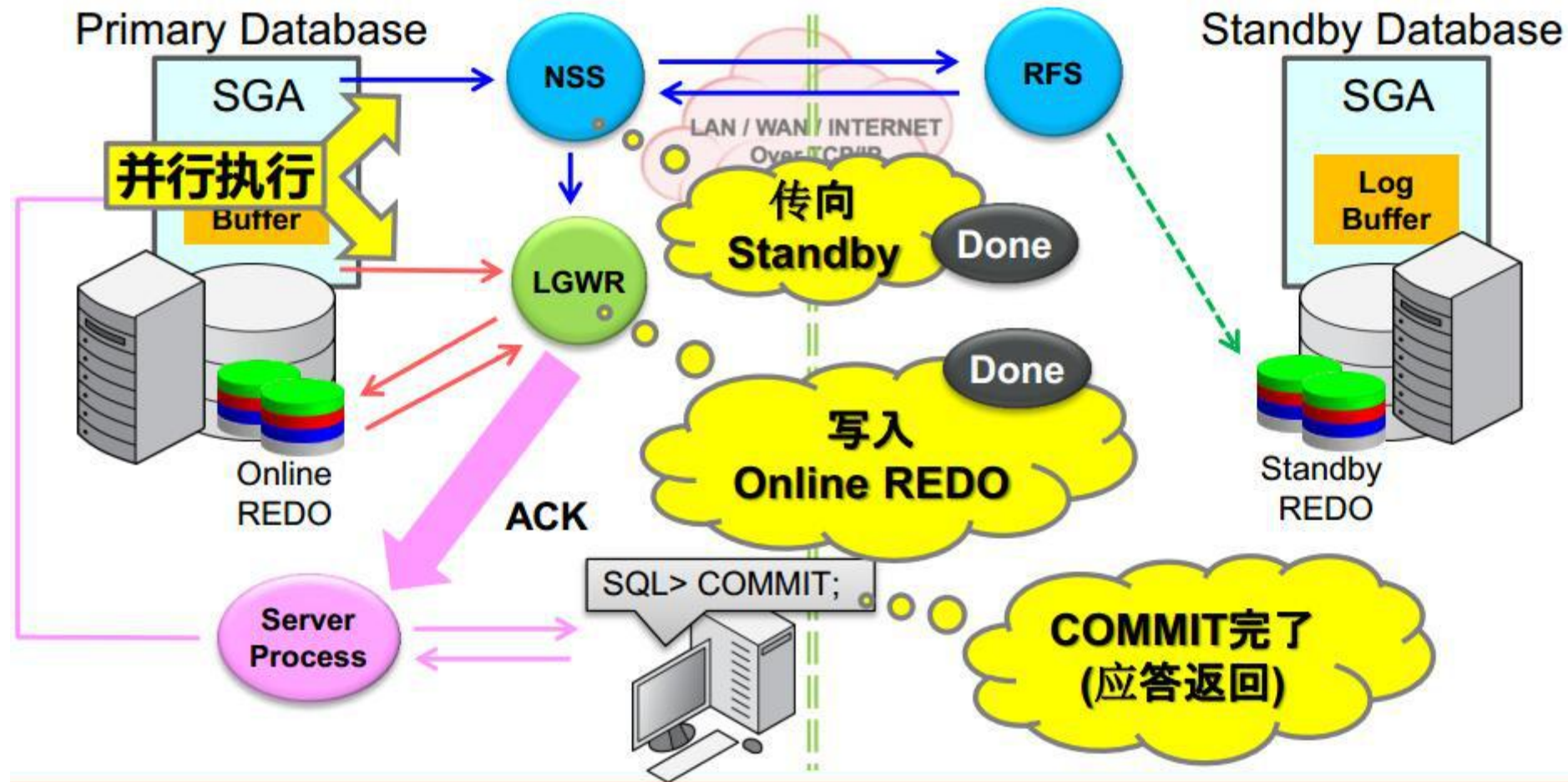
每 60秒 检测是否有未传送的 Archive Log File

如果有未传送的 Archive Log File ，将自动传送过去
执行 Log Switch (11gR2 only)

Log传送发生问题后，经过REOPEN时间，自动执行 Log Switch

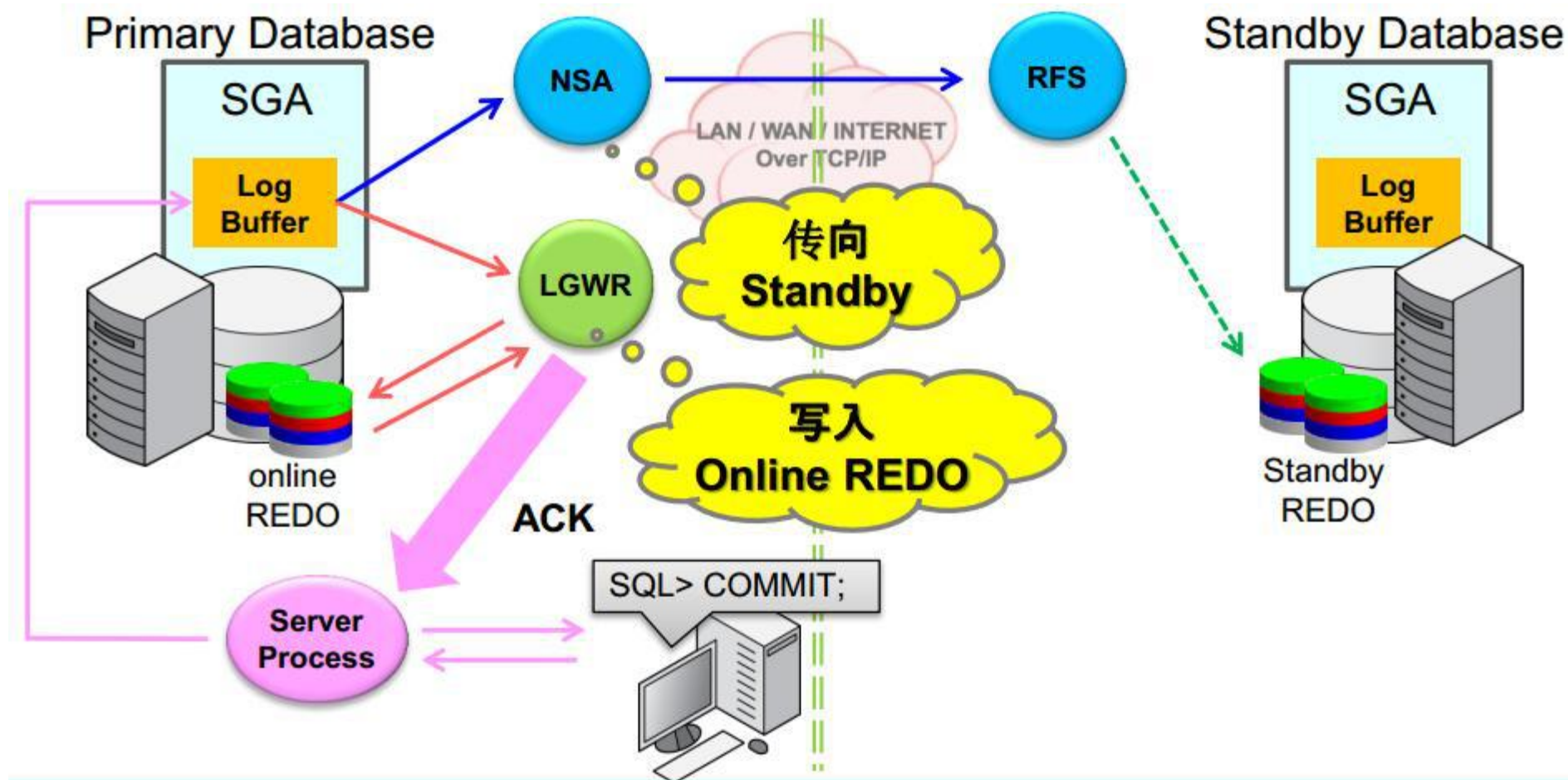
Data Guard 基本架构-日志传输

同步传输



Data Guard 基本架构-日志传输

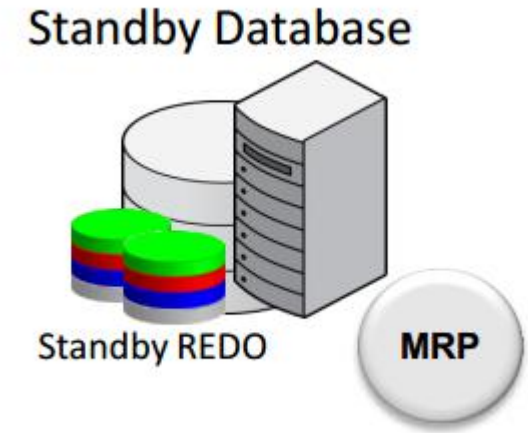
异步传输



Data Guard 基本架构-日志应用

Redo Apply 机制

1. Default 的应用方法
2. Real time 的应用方法（即时应用）
3. Time delay 应用



根据 MRP 的设定处理应用日志

REDO Log / Archive Log File 的接收可以在MOUNT / OPEN 状态下进行
MRP 将接收到的REDO Log / Archive Log File 自动应用

02

Active DataGuard 监控

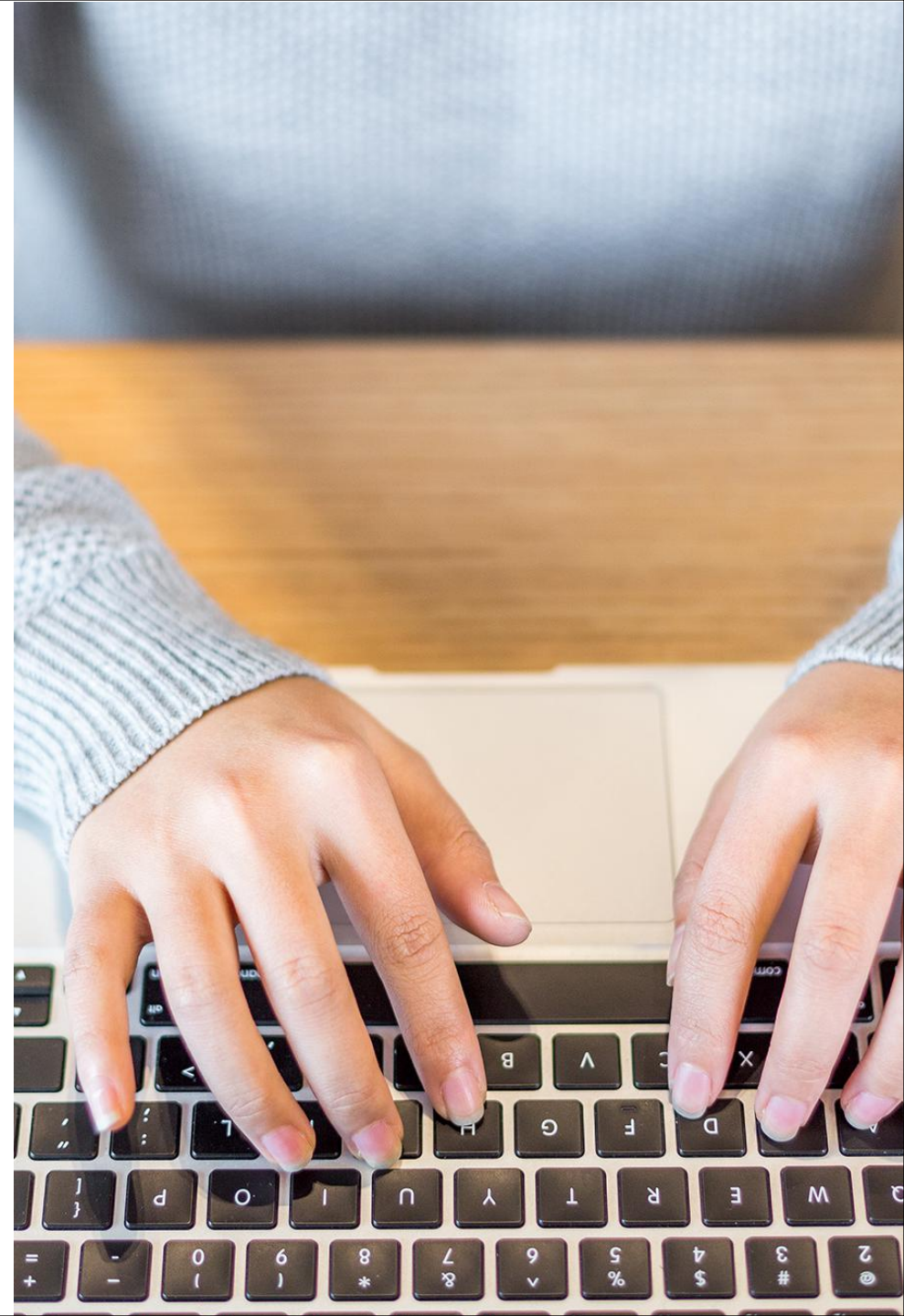


Active DataGuard 监控

主动监控数据库对于保持生成数据库启动和运行至关重要，对于 Active Dataguard 环境，我们更是密切监控数据库拓扑架构中的选择性组件，同时满足基于RPO 和 RT0 的恢复要求：找到潜在停机风险的影响，是尽量延迟生产生产环境运行时间的最好方式。

监控解决方案将数据库警报按严重程度分为两个或者三个类别：

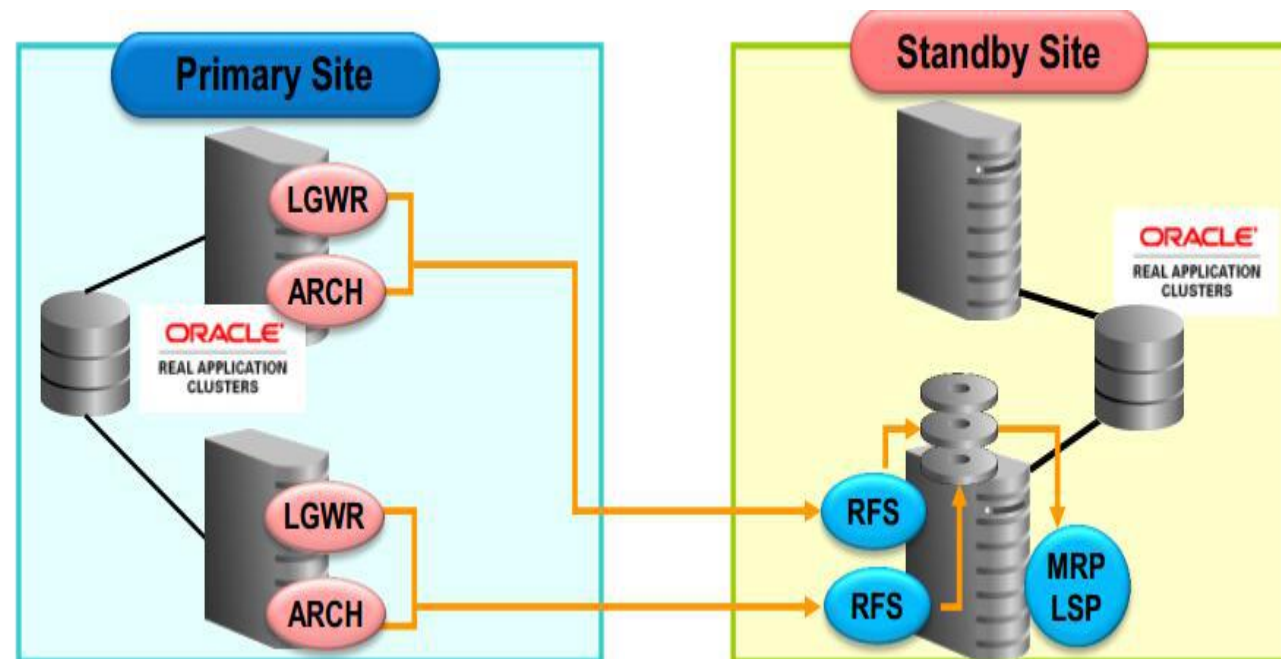
critical , warning 和 monitor



Active DataGuard 监控

通常至少监控生产数据库一下状况：

- 监控日志：
- 归档日志记录统计信息
- 检测归档日志间隔
- 检测重做传输中的延迟
- 监控归档日志的目的地
- 检测应用率和活动率
- 审阅传输和应用滞后
- 确定备用数据库的当前时间
- 管理恢复进程状态



Active DataGuard 监控

Sat Dec 01 03:16:43 2018

LNS: Standby redo logfile selected for thread 1 sequence 51758 for destination LOG_ARCHIVE_DEST_2

Sat Dec 01 03:16:43 2018

RFS[7]: Selected log 18 for thread 1 sequence 51758 dbid 1845160853 branch 923060696

Sat Dec 01 03:16:43 2018

Media Recovery Waiting for thread 1 sequence 51758 (in transit)

Recovery of Online Redo Log: Thread 1 Group 18 Seq 51758 Reading mem 0

Mem# 0: +ZGADAT DG/hlzqa/onlinelog/group_18.701.982034595

Sat Dec 01 04:17:21 2018

Archived Log entry 15083 added for thread 1 sequence 51758 ID 0x71c6fc64 dest 1:

Active DataGuard 监控

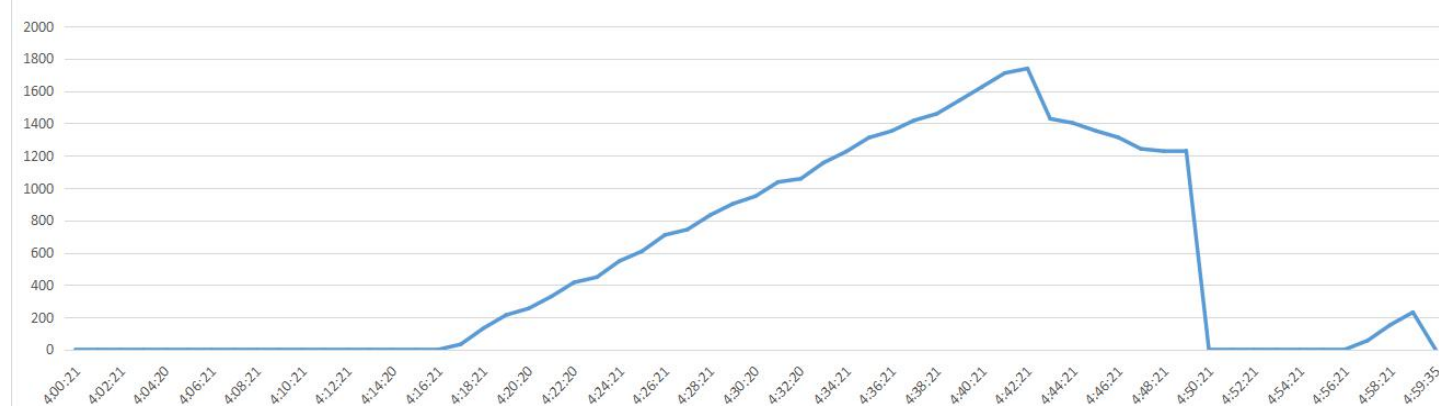
gv\$log_history	---- 历史redo log 会话详细
gv\$archive_gap	---- 关于备用数据库上的归档的信息
gv\$archived_log	---- 主库上归档详细
gv\$asm_disgroup	--- 数据库使用ASM 存储当前使用情况
v\$recovery_process	--- 用于跟踪数据库恢复操作，以确保DG 正常应用，并估计完成正在进行的操作所需的时间。
v\$dataguard_stats	--- 当在备用数据库上查询时，显示有关数据保护指标的信息。在主数据库上查询时不返回任何行。
gv\$database	--- 数据库SCN 查询
v\$managed_standby	---显示与数据保护环境中的物理备用数据库相关的Oracle数据库进程的当前状态信息。
gv\$archive_dest_statu	---显示数据库本地和远程归档 是否正常

Active DataGuard 监控-案例

DBA 节点一 二 每秒日志生产量



dba1 节点一 日志应用延迟时间



03

Active DataGuard 性能 分析

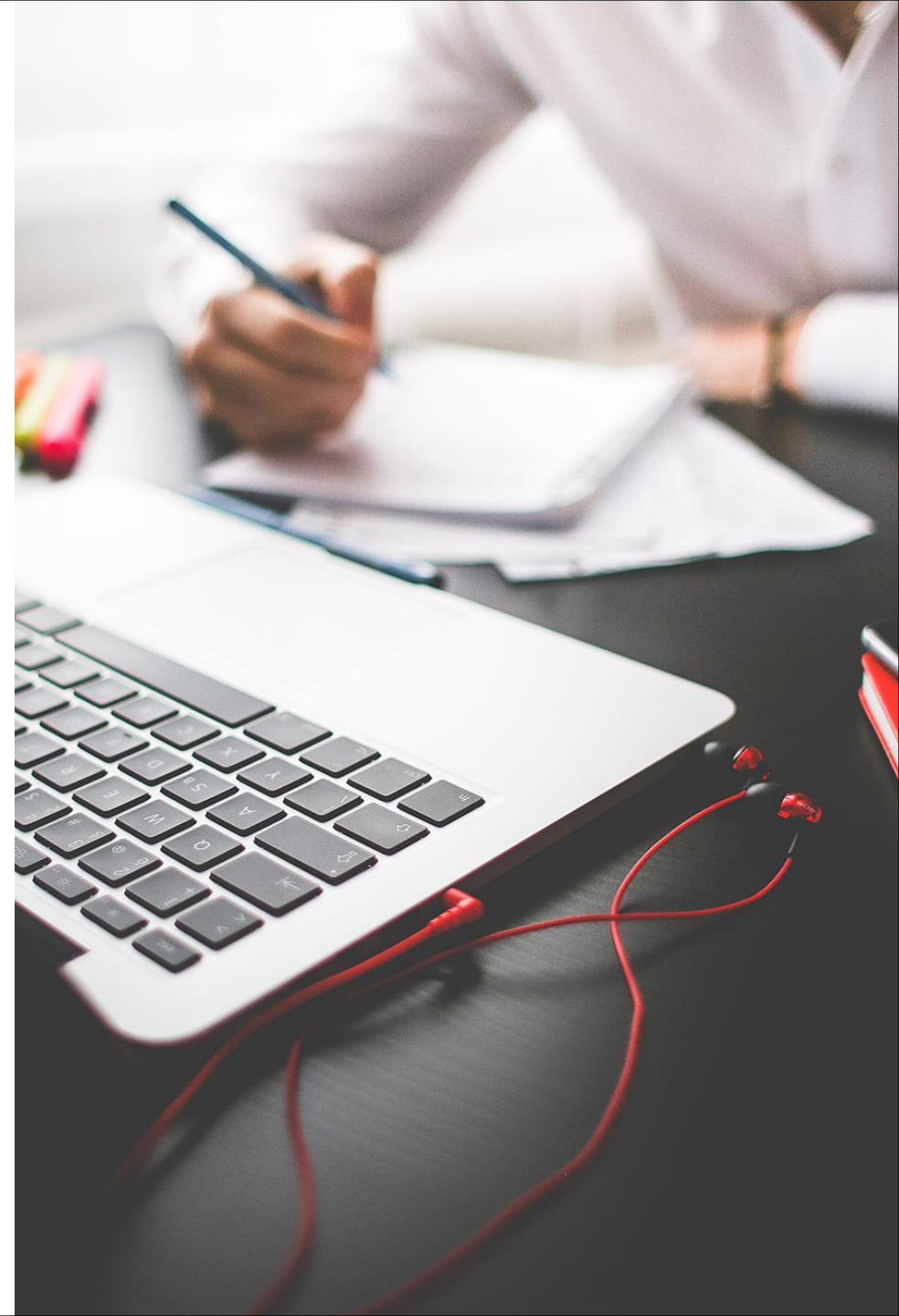


Active DataGuard 性能分析

在优化Oracle DG 数据的性能时，需要考虑两个方面保护配置：

第一种：是传输重做的重做传输服务，
生产数据库生成到本地或远
程备用数据库的数据

第二种：是日志应用服务，它将重做
数据应用到备用数据库



Active DataGuard 性能分析

一旦主数据库和备用数据库之间所需的带宽达到, 我们通过优化配置可以实现的改进。为每种类型的传输执行最优方式:

Primary Database

ARCH Redo Transport:

ARCn processes max_conections

LGWR ASYNC Redo Transport:

I/O bandwidth

LGWR SYNC Redo Transport:

NET_TIMEOUT COMMIT NOWAIT

All Redo Transports:

I/O bandwidth standby redo logs

Active DataGuard 性能分析

网络方面考虑

TCP Socket Buffer Sizes:

RECV_BUF_SIZE SEND_BUF_SIZE SDU

Network Device Queue Sizes:

TXQUEUELENGTH NET_DEV_MAX_BACKLOG

Oracle Net

TCP_NODELAY

STANDBY I/O

oracle ASYNC I/O

Best I/O stack

SRLs

Typically RAID

Testing Stage	Test duration (seconds)	Amount of data transferred	Network throughput achieved (Megabits/sec)	% change
Prior to tuning	60	77.2 MB	10.8 Mbps	N/A
After increasing network socket buffer size to 3*BDP from default of 16K	60	5.11 GB	731.0 Mbps	665% improvement over baseline prior to tuning
After above adjustment and increase of device queue lengths to 1,000 from default of 100	60	6.55 GB	937.0 Mbps	28% improvement

Active DataGuard 性能分析

Wait Events Related to Data Guard

Primary Database :

ARCH wait on ATTACH	----	spawn a new RFS connection.
ARCH wait on SENDREQ	----	write archive logs
ARCH wait on DETACH	----	delete an RFS connection.
LNS wait on ATTACH	----	spawn a new RFS connection.
LNS wait on SENDREQ	----	write archive logs and open and close the remote archived redo logs.
LNS wait on DETACH	----	delete an RFS connection
LGWR wait on LNS	----	receive messages from LNS processes
LNS wait on LGWR	----	receive messages from the log writer (LGWR) process
LGWR-LNS wait on channel	----	waiting to receive messages

Standby Database:

RFS Write	-----	I/O work
RFS random i/o	----	write to a standby redo log
RFS sequential i/o	----	write to an archive log

谢谢！