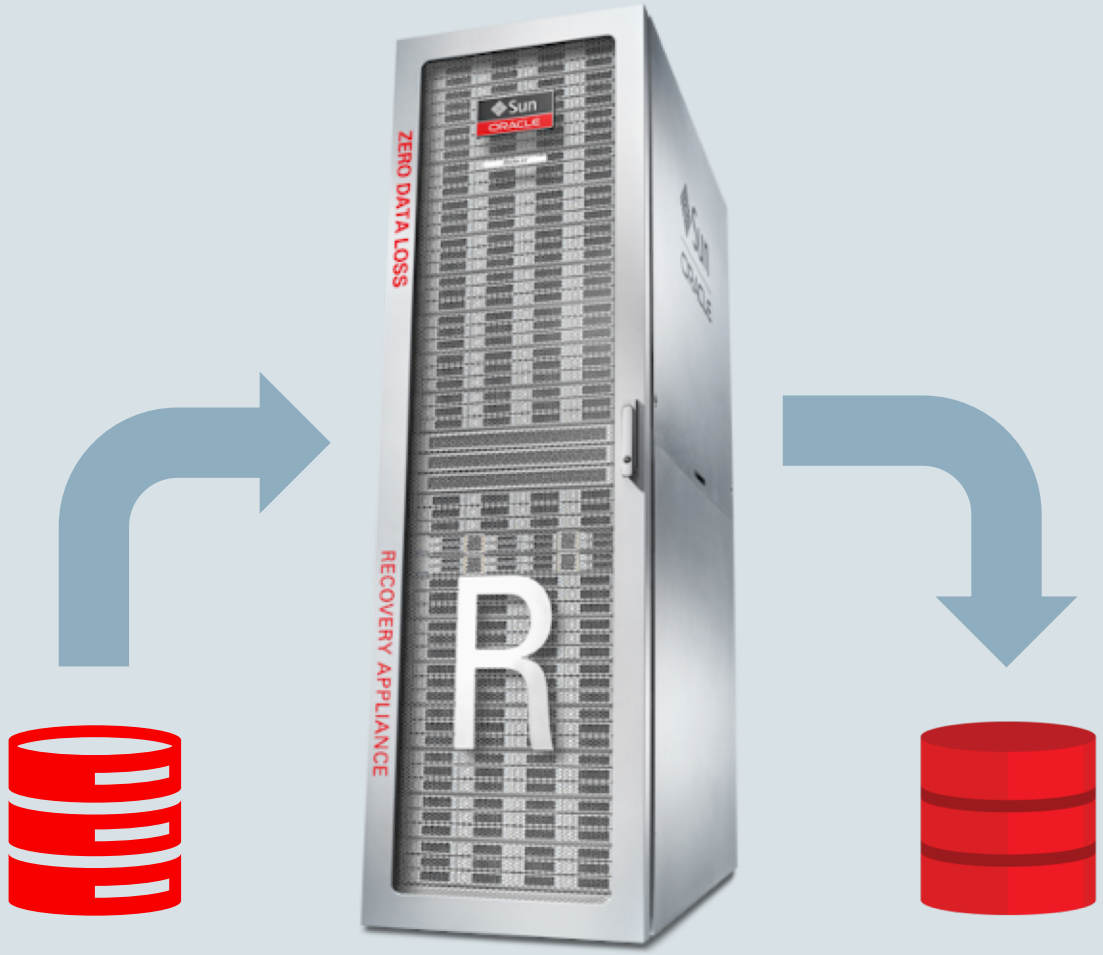

ORACLE®

Database Platform Migration Using ZDLRA

Oracle Server Technology
High Availability Systems Development
Maximum Availability Architecture
September 2020



Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Program Agenda

1. Solution Overview
2. ZDLRA Backup and Restore Strategy
3. Database Migration Process using dbmigusera.pl
4. Customer Case Study

Solution Overview

- Database migration is the process of moving databases to a new platform, including Exadata and ExaCC
- Zero Data Loss Recovery Appliance (ZDLRA) provides a simple, reliable, and fast migration solution with minimal downtime

This presentation outlines the steps to execute:

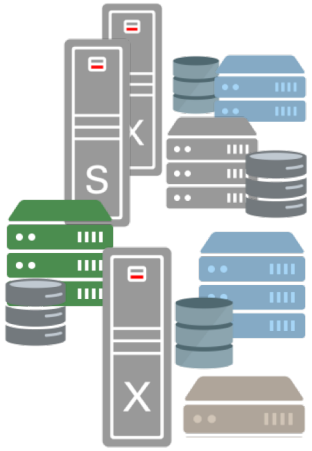
- Cross-endian platform DB migration using ZDLRA
- Same-endian platform DB migration – even simpler and faster with ZDLRA

Program Agenda

1. Solution Overview
2. ZDLRA Backup and Restore Strategy
3. Database Migration Process using ZDLRA
4. Overview – Real Life Scenario

Zero Data Loss Recovery Appliance Overview

Protected Databases



Delta Push

- DBs access and send only changes
  - Minimal impact on production
- Real-time redo transport instantly protects ongoing transactions

Recovery Appliance

Offloads Cloud / Tape Backup

Replicates to Remote Recovery Appliance

Protects all DBs in Data Center

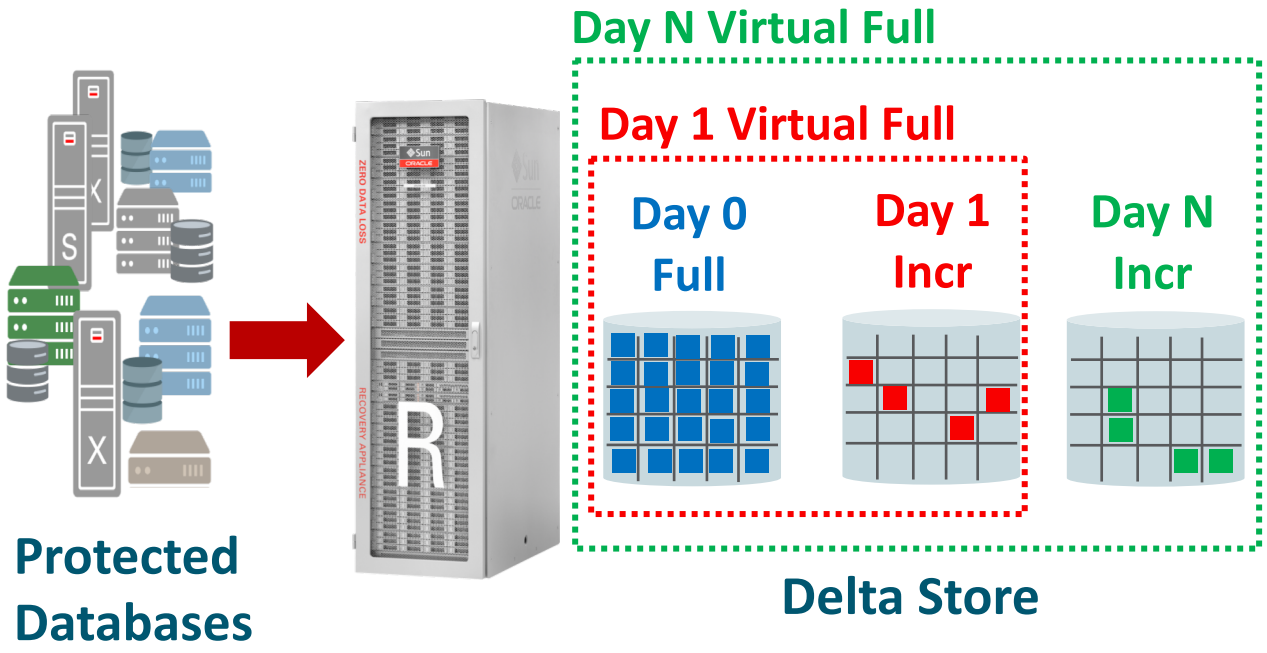
- Petabytes of data
- RMAN-driven backup & restore
- Oracle 10.2-12.2, any platform
- No expensive DB backup agents

Delta Store

- Stores validated, compressed DB changes on disk
- Fast restores to any point-in-time using 'virtual full'
- Built on Exadata scaling and resilience
- Enterprise Manager end-to-end control

Space-Efficient “Virtual” Full Backups

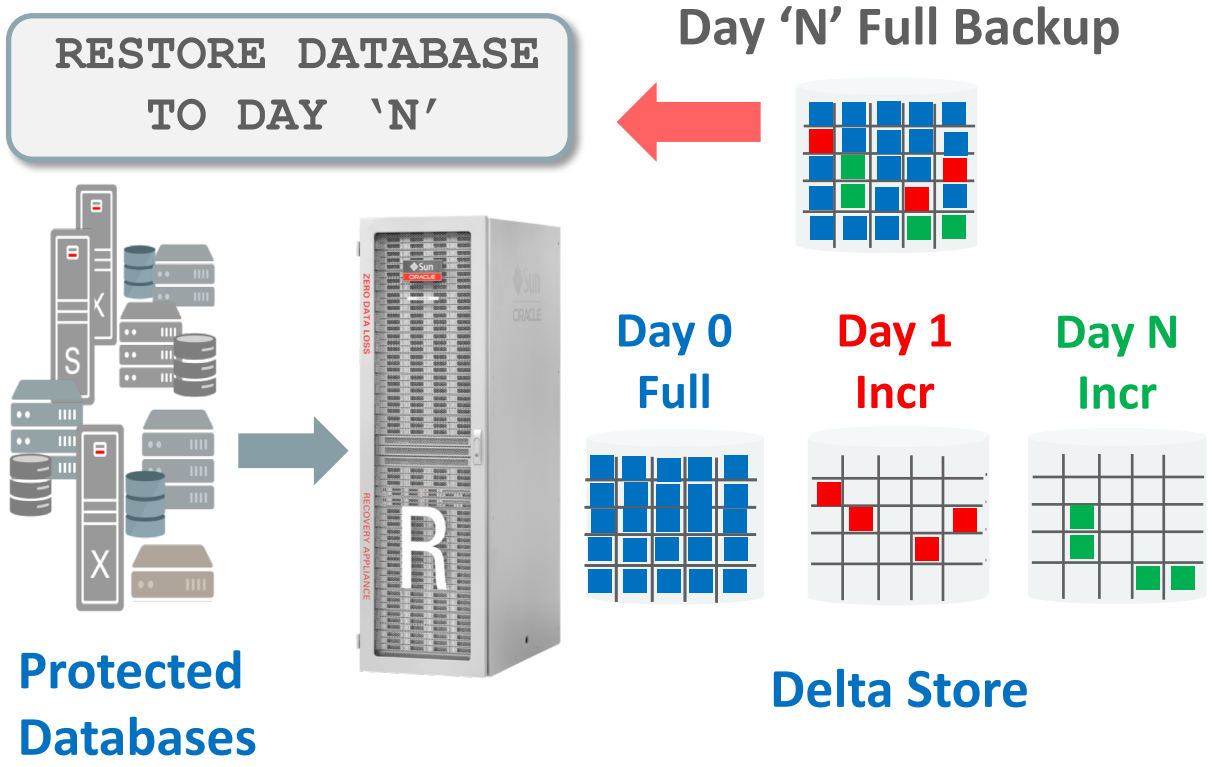
No More Full Backups: Incrementals Forever Architecture



- After one-time full backup, incrementals used to create ***virtual*** full database backups on a daily basis
  - Pointer-based representation of physical full backup as of incremental backup time
  - Virtual backups typically 10x space efficient
  - Enables long backup history to be kept with the smallest possible space consumption
    - **“Time Machine” for database**

Fast Restore to Any Point-in-Time

No Incremental Apply = No Load on Production Server



- Directly restore any virtual full backup
  - All blocks referenced from virtual full are efficiently retrieved
  - Eliminates production server overhead of traditional restore and merge of incrementals
- Inherits scalability and performance via Exadata-based hardware platform

Program Agenda

1. Abstract
2. ZDLRA Backup and Restore Strategy
3. Database Migration Process using ZDLRA
4. Overview – Real Life Scenario

Database Migration Process using ZDLRA

Database migration can be either:

1. Cross-Endian – Big Endian to Small Endian, or vice versa.
2. Across the Same-Endian.

Where did the word endianness came from?
From a passage in *Gulliver's Travels* in which an emperor, after cutting his finger after opening an egg at the large end, commands his subjects to open them at the small end; those who comply are called “Little-Endians”, while those who rebel by opening their eggs at the large end are called “Big-Endians.”

In computer architecture:
•In big-endian format, whenever addressing memory or sending/storing words byte-wise, the most significant byte—the byte containing the most significant bit—is stored first (has the lowest address) or sent first, then the following bytes are stored or sent in decreasing significance order, with the least significant byte—the one containing the least significant bit—stored last (having the highest address) or sent last.

•Little-endian format reverses this order: the sequence addresses/sends/stores the least significant byte first (lowest address) and the most significant byte last (highest address).

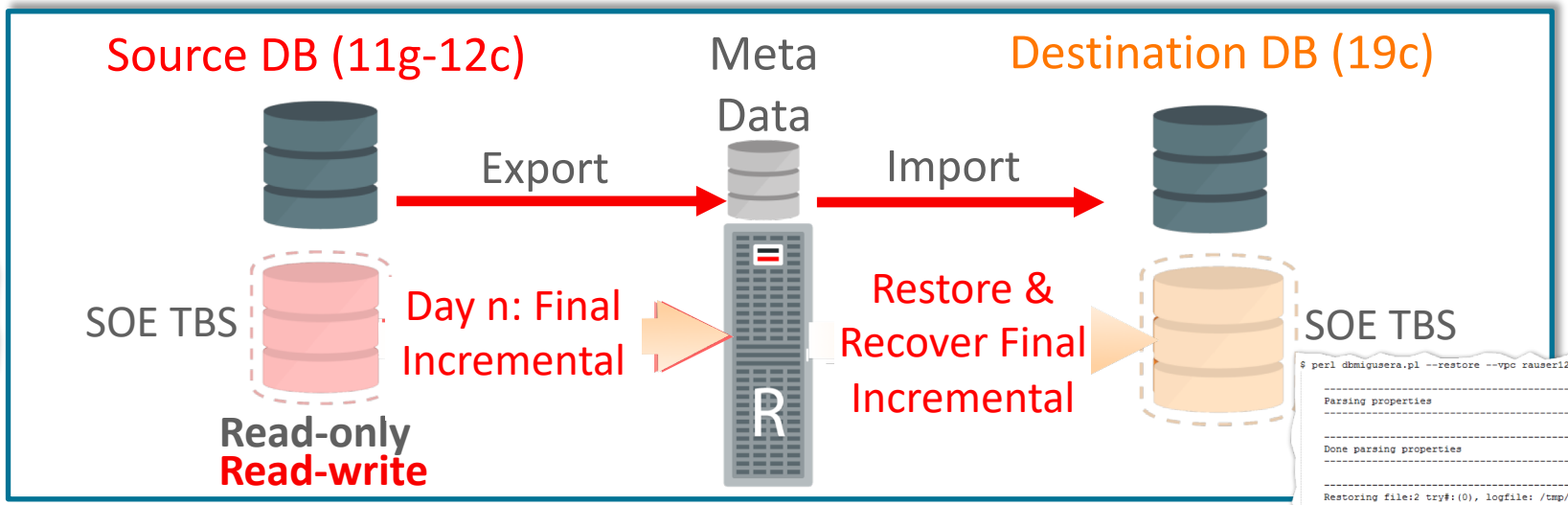
Cross-Endian Database Migration using ZDLRA

1. Performing incremental backups as normal from the source database to ZDLRA
2. Create a new database on the destination platform.
3. On the destination database, restore the latest virtual level 0 backup from the ZDLRA for the source tablespaces.
   - The restore operation will convert and copy the data files to the new destination database storage.
   - The migration procedure will also create an export data pump file to plug in the tablespaces at the destination.
4. Recover the destination data files with incrementals (i.e. data changes) taken at the source as many times as desired to keep the data in sync as close as possible between the two databases
   - Ideally, the recover operation should be run on regular basis (e.g. daily), until it takes less than 2 hours.
5. Select a final migration window to the destination database, ideally where there are minimal source database activities:
   - On the source database:
     1. Put the tablespaces on the source database in read only mode
     2. Take a final incremental level 1 backup to the ZDLRA
     3. Export the transportable tablespaces definition (metadata dump file)
   - On the destination database:
     1. Recover the data files with the final incremental backup and export the transportable tablespaces
     2. Import the tablespace metadata dump file to plug in the tablespaces
     3. Activate destination database and enable services

Database Migration using ZDLRA

- Dramatically Reduce Read-only Downtime – Several Hours+ to Few Hours or Less

```
RUN
{
 BACKUP DEVICE TYPE SBT
  CUMULATIVE INCREMENTAL LEVEL 1
  FILESPERSET 1 SECTION SIZE 64G
  DATABASE
    PLUS ARCHIVELOG NOT BACKED UP
      FILESPERSET 32;
}
```




```
$ perl dbmigusera.pl --restore --vpc rauser12/[redacted]@[redacted]oracle.com:1521/zdlra8:dedicated -L

------------------------------------------------------------------------------
Parsing properties
------------------------------------------------------------------------------

------------------------------------------------------------------------------
Done parsing properties
------------------------------------------------------------------------------

------------------------------------------------------------------------------
Restoring file:2 try#:(0), logfile: /tmp/mignew1//ramigratedb/restore_Aug15_Wed_18_54_32_581//rman_2.log
------------------------------------------------------------------------------

------------------------------------------------------------------------------
Restoring file:5 try#:(0), logfile: /tmp/mignew1//ramigratedb/restore_Aug15_Wed_18_54_32_581//rman_5.log
------------------------------------------------------------------------------

------------------------------------------------------------------------------
```

- Minimal downtime – DB remains read-write throughout process
  - Tablespaces are only placed in read-only prior to final incremental
- Daily incremental backups -> virtual full backups on Recovery Appliance
- Restore latest virtual full backup prior to start of migration window
  - RESTORE FROM PLATFORM XXX FOREIGN DATAFILE YYY

```
$ perl dbmigusera.pl --recover --vpc rauser12/[redacted]@[redacted]oracle.com:1521/zdlra9:dedicated -L

------------------------------------------------------------------------------
Setting up files for recover
------------------------------------------------------------------------------
Taking handle=VBS_1483149084_5521342I       ckp_scn=7570539          incr_scn=7251914     bs_key=5521353
Taking handle=VBS_1483149084_5523174I       ckp_scn=7601575          incr_scn=7570539     bs_key=5523183
.
.
------------------------------------------------------------------------------
Recovering file:19 try#:(0), index:(1), logfile: /tmp/mignew1//ramigratedb/recover_Aug20_Mon_11_24_12_69//rman_19_1.log
------------------------------------------------------------------------------

------------------------------------------------------------------------------
Recovering file:20 try#:(0), index:(1), logfile: /tmp/mignew1//ramigratedb/recover_Aug20_Mon_11_24_12_69//rman_20_1.log
------------------------------------------------------------------------------
```

- Final incremental taken in read-only, then restored & recovered onto destination files
  - RECOVER FROM PLATFORM XXX FOREIGN DATAFILECOPY YYY
  - Data pump metadata export at source and import of metadata dump file at destination

Cross Platform DB Migration using ZDLRA - Benefits

Primary benefits:

1. Significant reduction in downtime (Potentially less than two hours of service downtime)[1]
2. Application service is READ ONLY during most of the *service downtime*
3. New ZDLRA tool (dbmigusera.pl) simplifies cross platform migration by automating steps, especially useful for large databases
4. Migration time is not impacted by the size of the database

Read-only downtime based on:

1. Final incremental backup and recovery step – incremental size should be < 5% of DB size
2. Final tablespace metadata export/import duration (only necessary for cross platform)
3. Number of target tablespaces – import is done serially
4. Application service switchover from source to destination database

Refer to MOS Note: Cross Platform Database Migration using ZDLRA (Doc ID 2460552.1)

[1]**Downtime does not include the user objects to be imported on the destination.**

Same-Endian DB Migration using ZDLRA

- Leverage Data Guard and ZDLRA for minimal downtime migration
  - By using the source database backups available on the ZDLRA, instantiate new database on the destination platform using RMAN DUPLICATE FOR STANDBY.
  - Start MRP to synchronize standby with primary database
  - Verify destination data by opening standby read-only
  - Perform switchover and redirect application clients to new primary database
  - Refer to MOS Note : Creating a Physical Standby Database in an 11.2, 12.1, 12.2 or later environment (Doc ID 2275154.1)[1]
- Advantages:
  - Near-Zero to Zero downtime during switchover to destination database
  - Increased availability during the migration process vs transportable tablespace approach
  - With Active Data Guard, standby database can be used to offload read-only activities from primary (e.g. reporting), until switchover is performed

[1]When using RMAN DUPLICATE, connect to the ZDLRA as CATALOG and configure the RMAN AUXILIARY SBT channels to connect to the ZDLRA.

Program Agenda

1. Abstract
2. ZDLRA Backup and Restore Strategy
3. Database Migration Process using dbmigusera.pl
4. Customer Case Study

Customer Case Study

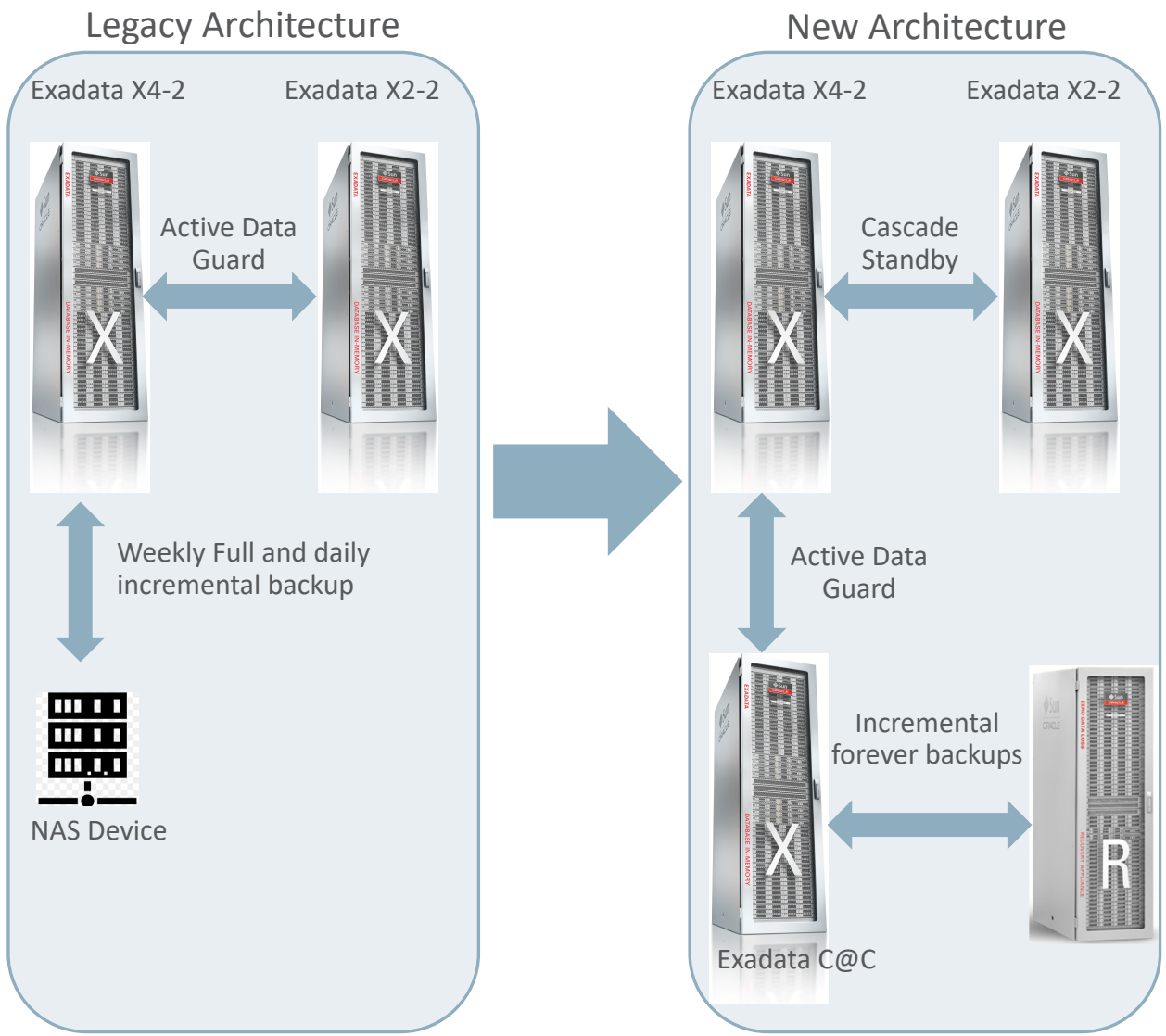
Leading Global Semiconductor Manufacturer

Business Needs

- Accelerate Growth
- Drive Operational Excellence
  - Customer Experience
  - Operational Efficiency
- Grow organizational capabilities – optimize innovation
- Address current & planned business growth objectives

Solution Needs

- Stability
- Zero Preventable Outages
- Focus on Business Ops
- Increase IT agility, self-service and alignment to business drivers



Results Achieved

- Consolidate and standardize
  - Consolidated several DBs to ExaCC + Multi-Tenant option
- Improved the time and cost to build and maintain Analytics platform
- Near zero downtime DB migration to ExaCC using RA + RMAN DUPLICATE.
- Deliver exceptional service to business users
- Eliminated full backups
- Reduced incremental backup time by 2X
- Improved RTO by 4X

ORACLE

Leading European Retailer

Current Architecture

Oracle Databases
on IBM/AIX

Incremental
forever backups

Future Architecture

Exadata C@C

Incremental
forever backups

Business Goals

Enhance the customer shopping experience

- Provide a robust platform for the point-of-sale databases
- Improve overall performance
- Increase time to market
- Data center modernization
  - **Migrating from IBM AIX to Exadata**
- Database consolidation

Integrated Cloud

Applications & Platform Services


ORACLE®