

WHITE PAPER

Meeting Oracle Recovery SLAs at Enterprise Scale

High-performance Oracle RMAN backup and recovery with FlashBlade

Contents

Executive summary	3
Challenges with Oracle backup and recovery	3
Backup windows	3
Recovery time objectives	4
Solution overview	4
Solution benefits	4
Technology overview	5
Oracle Recovery Manager (RMAN)	5
Pure Storage FlashBlade	6
Solution patterns	7
Pattern A: Consolidated backup repository	7
Pattern B: Distributed backup and recovery with RAC	8
Pattern C: Incremental merge (single instance and RAC)	11
Performance validation	14
Environment	14
RMAN parameter optimization	15
RAC: Distributed backup and recovery	17
Standalone instances: Consolidation and scale	18
Conclusion	18



Executive summary

Enterprise Oracle environments face a dual challenge: backing up growing databases within shrinking backup windows while still ensuring rapid recovery when disaster strikes. Traditional backup infrastructure forces a compromise—either backup frequency is reduced to accommodate large databases, or recovery times stretch into days because the storage layer was optimized for capacity rather than throughput. Software solutions can improve efficiency, but they cannot change the fundamental physics of how fast data moves to and from storage.

Pure Storage® FlashBlade® approaches this problem differently. Rather than optimizing around storage limitations, it eliminates them. With a scale-out architecture purpose-built for high-throughput workloads, FlashBlade enables Oracle Recovery Manager (RMAN) backup and restore rates measured in tens of terabytes per hour. Using Oracle Direct NFS (dNFS) as the data path, RMAN drives FlashBlade at full throughput. In validated testing, an 8TB database was restored in under nine minutes.

This paper provides storage and platform teams with validated architectures, proven performance benchmarks, and example deployment patterns for implementing Oracle RMAN backup and recovery on FlashBlade. Actual performance will vary based on database characteristics, network configuration, and infrastructure. Whether consolidating backup for dozens of standalone databases or protecting a mission-critical Oracle Real Application Cluster (RAC) environment, FlashBlade delivers the performance required to meet enterprise backup and recovery SLAs with margin to spare.

Challenges with Oracle backup and recovery

Protecting enterprise Oracle databases requires solving two related problems: completing backups within shrinking windows and recovering fast enough to meet business SLAs. Both come down to storage throughput.

Backup windows

According to the [2023 IOUG Database Priorities Survey](#), 28% of Oracle environments now manage databases exceeding 25TB, with the 5–25TB range becoming the norm for mission-critical systems. These are no longer edge cases—they are the new baseline.

For a four-hour backup window, the required throughput scales with database size as shown in Table 1.

Database size	Required throughput
25TB	6TB/hour or more
50TB	12TB/hour or more
100TB	25TB/hour or more

TABLE 1 Required throughput based on database size

Mid-range deduplication appliances typically achieve 5–6TB/hour. Even top-tier systems claim only 20–30TB/hour under ideal conditions, and real-world performance often falls short of spec sheet numbers.

The result is a slow compression of options: reduce backup frequency, extend maintenance windows, or accept that full backups are no longer possible within operational constraints.



Recovery time objectives

Backup window pressure is visible and plannable, but recovery failures are not. Traditional backup architectures are optimized for backup throughput at the expense of restore speeds. Deduplication appliances compress data efficiently on the way in but must rehydrate on the way out. Restore throughput on dedupe storage typically runs significantly slower than backup, with mid-size appliances delivering less than 10TB/hour for restore operations.

The result is asymmetric risk. An organization may have perfect backup compliance yet face a 12-hour or 24-hour recovery when an incident occurs. Every database administrator has a story like this. In one documented case, a full Oracle server restore from tape backups took two days to complete. The backup succeeded, but the infrastructure failed.

Solution overview

This paper validates an approach using Pure Storage FlashBlade as the backup target for Oracle RMAN, with Oracle dNFS providing the data path between database hosts and storage.

The architecture is straightforward: RMAN backup and restore operations write directly to FlashBlade over Network File System (NFS), with dNFS enabling parallel data streams from the Oracle kernel. This eliminates the need for backup agents, intermediate appliances, or proprietary data formats. For environments using enterprise backup software, FlashBlade also integrates with solutions like Commvault and Veeam as a high-performance backup target.

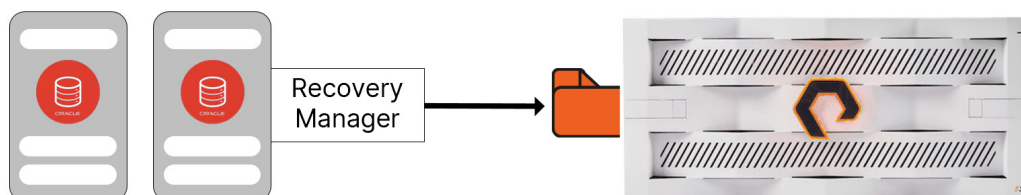


FIGURE 1 Oracle RMAN backup and recovery from NFS on FlashBlade using dNFS

Solution benefits

FlashBlade delivers several key benefits that address the backup and recovery challenges faced by enterprise Oracle environments:

- **Meet aggressive recovery time objectives (RTOs):** Restore very large databases in minutes rather than hours with parallel, high-throughput recovery operations.
- **Protect databases within backup windows:** Achieve consistent backup throughput regardless of database size. Because backup duration scales linearly with size, there is no performance regression as databases grow.
- **Scale independently:** Add capacity and performance as database portfolios grow without impacting existing infrastructure or requiring forklift upgrades.
- **Simplify recovery operations:** Eliminate tape management complexity with fast, direct recovery from NFS storage.
- **Support concurrent operations:** Maintain consistent performance when backing up multiple databases simultaneously across large database portfolios.
- **Protect backup data:** FlashBlade SafeMode™ Snapshots provide immutable recovery points resistant to ransomware and accidental deletion. Replication to a secondary site or cloud extends protection for disaster recovery.

The following sections explain each component, provide configuration guidance, and validate these benefits with tested results.



Technology overview

Oracle Recovery Manager (RMAN)

Oracle Recovery Manager (RMAN) is an Oracle Database-native backup and recovery solution, integrated into the database kernel. RMAN performs block-level backups, reading data directly from datafiles and writing to backup destinations without requiring the database to be taken offline.

RMAN supports parallel operations through channels. Each channel represents a separate server process that can read from the database and write to the backup destination simultaneously (see Figure 2). For high-throughput environments, configuring multiple channels allows RMAN to scale backup and restore performance with available I/O bandwidth.

Key capabilities relevant to this solution:

- Incremental backups capture only changed blocks since the last backup, reducing backup volume and duration for large databases.
- Incremental merge maintains a continuously updated, full backup image by applying incremental changes, eliminating periodic full backup requirements.
- Block-level recovery restores individual corrupt blocks without requiring full datafile recovery.

RMAN writes backups as backup sets (proprietary format) or image copies (direct datafile copies). For FlashBlade, image copies provide the fastest restore path since they require no conversion during recovery.

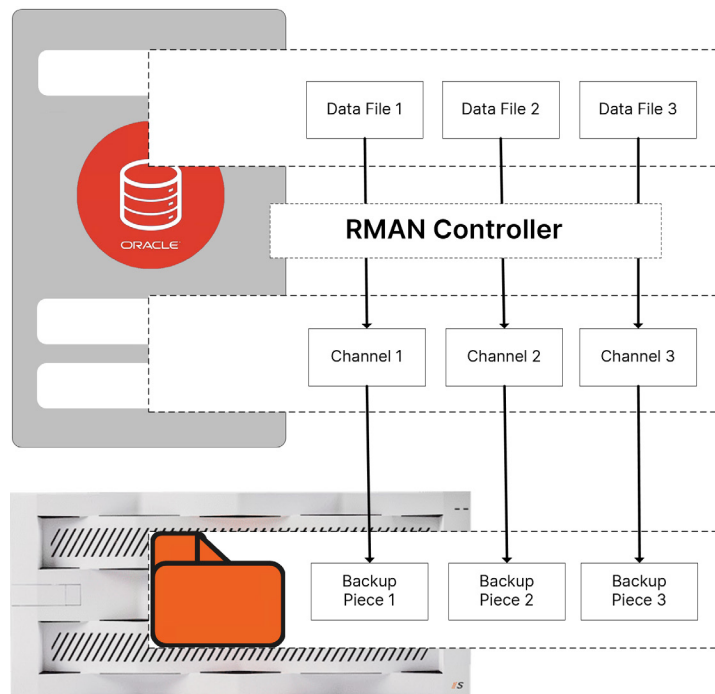


FIGURE 2 Oracle RMAN backup and recovery data flow

Oracle Direct NFS (dNFS)

[Oracle Direct NFS \(dNFS\)](#) is an optimized NFS client built into the Oracle database kernel, available since Oracle 11g. It bypasses the operating system's native NFS client, allowing Oracle to manage NFS connections directly from database processes.

For backup and recovery workloads, dNFS provides several advantages:

- **Parallel I/O:** dNFS opens multiple parallel connections to NFS storage, enabling RMAN channels to drive higher aggregate throughput than a single NFS mount.
- **Connection management:** dNFS automatically balances connections across available network interfaces and recovers from transient failures without interrupting database operations.
- **Reduced overhead:** By eliminating the kernel NFS layer, dNFS reduces CPU overhead and memory copies in the I/O path.

dNFS is configured through the orafstab file, which specifies NFS server addresses, export paths, and network interfaces. In this white paper, dNFS is used with network interface card (NIC) bonding rather than multiple dNFS interfaces or data virtual IP (VIP) addresses. As a result, dNFS load balancing and high availability features across multiple interfaces are not demonstrated.

Pure Storage FlashBlade

FlashBlade is a scale-out file and object storage platform designed for high-throughput workloads. Unlike traditional network-attached storage (NAS) systems that scale capacity and performance together, FlashBlade allows independent scaling of both dimensions.

In addition to NFS-based backups using dNFS, Oracle RMAN backups to S3-compatible object storage on FlashBlade are also supported, providing an alternative target for backup and long-term retention workflows.

Figure 3 shows the FlashBlade product family and key features of each model.

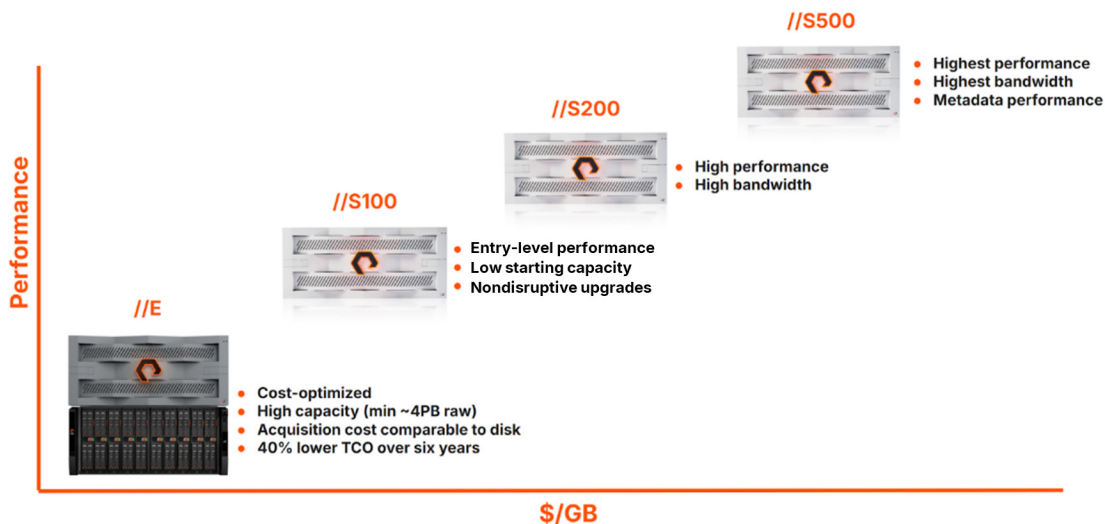


FIGURE 3 Pure Storage FlashBlade product portfolio



Architecture characteristics relevant to Oracle backup and recovery include:

- **Scale-out metadata.** FlashBlade distributes metadata across all blades in the system, eliminating single-controller bottlenecks that limit traditional NAS performance under parallel workloads.
- **Symmetric read/write performance.** FlashBlade delivers consistent throughput for both read and write operations. There is no performance asymmetry between backup (write) and restore (read) workloads.
- **Linear scaling.** Adding blades increases both capacity and throughput proportionally. A FlashBlade system can scale from tens of terabytes to multiple petabytes while maintaining consistent per-terabyte performance.
- **Performance parallelism.** FlashBlade distributes I/O across all blades internally, eliminating single-path bottlenecks common in traditional NAS architectures.

For RMAN backup targets, FlashBlade file systems are provisioned as standard NFS exports, with no special agents or backup integrations required.

Solution patterns

This section presents three deployment patterns for Oracle RMAN with FlashBlade, each addressing different operational requirements. Each pattern is not mutually exclusive but focuses on scenario building for best outcome. The [Oracle RMAN Configuration guide](#) provides deeper steps for how to configure dNFS and other properties if needed.

Pattern A: Consolidated backup repository

Use case: Multiple Oracle databases sharing FlashBlade as a central backup target.

Problem addressed: Managing backup infrastructure for database estates with tens or hundreds of databases creates operational overhead. Using separate backup targets for each database increases storage costs, complicates monitoring, and fragments capacity planning.

Solution: Consolidate backup operations to a single FlashBlade, with each database writing to a dedicated directory or file system (see Figure 4). FlashBlade scale-out architecture handles concurrent backup streams without performance degradation.

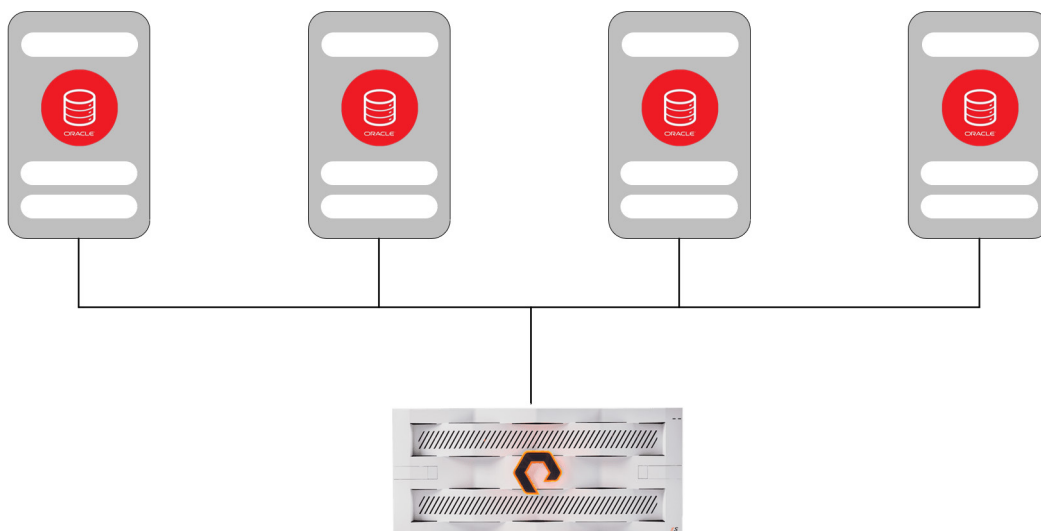


FIGURE 4 Backup and recovery storage consolidation



Configuration considerations

The following configuration considerations help guide how RMAN should be deployed and managed on FlashBlade:

- **File system layout:** Create separate file systems per database for isolation, or use a shared file system with directory-based separation. Separate file systems provide independent quotas and simpler capacity tracking. Shared file systems simplify provisioning and maximize storage efficiency.
- **Scheduling:** Concurrent backups work but compete for network bandwidth. For maximum per-database throughput, stagger backup windows. For operational simplicity, run concurrent backups and let FlashBlade distribute the load.
- **Retention:** Multiple databases with independent user-managed retention policies accumulate storage. FlashBlade thin provisioning and data reduction help, but capacity planning must account for aggregate retention across all databases.

Pattern B: Distributed backup and recovery with RAC

Use case: RAC environments where single-node I/O bandwidth is insufficient for backup or recovery requirements.

Problem addressed: A single database server has finite I/O capacity. Network bandwidth, CPU, and dNFS parallelism all have limits. When backup windows shrink or RTOs tighten, single-node throughput may not be enough. RAC clusters have multiple nodes with independent I/O paths, but by default RMAN runs from one node.

Solution: Distribute RMAN channels across RAC nodes to aggregate I/O bandwidth from the cluster (see Figure 5). Each node contributes its network capacity, CPU, and dNFS connections to the operation.

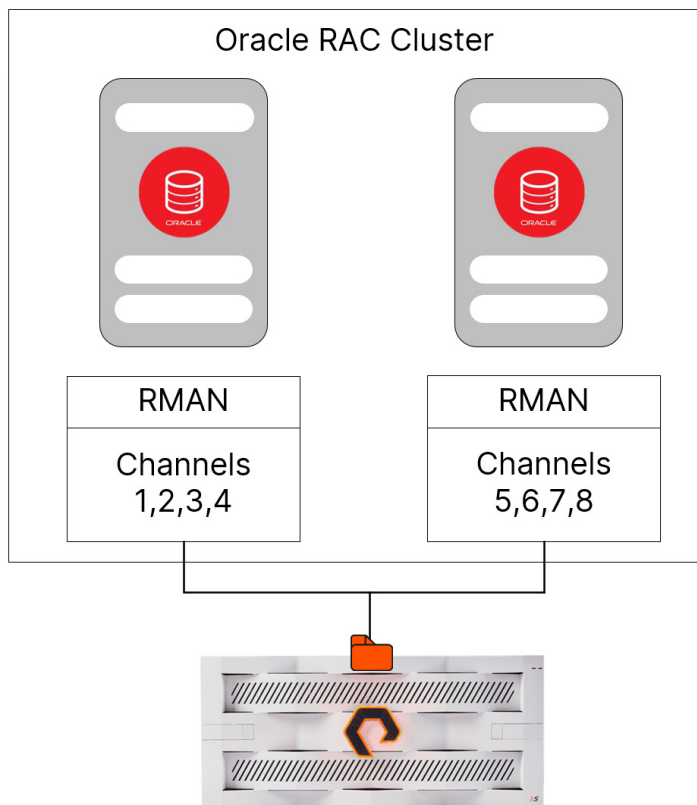


FIGURE 5 Channel distribution for RMAN backup and restore operations



Backup options

Single-node backup: All channels run on the node where RMAN is connected. This is simple to configure, but throughput is limited to that node's capacity. **In this mode, RMAN is connected to a single instance, so all channels execute on that node.** The following RMAN example demonstrates this configuration:

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 8;
BACKUP DATABASE;
```

Multi-node RAC backup: When RMAN connects through a RAC service that runs across multiple instances, Oracle automatically distributes backup channels across available nodes. As parallelism is increased, throughput scales with the number of participating RAC instances.

Node-specific CONNECT clauses are not required and are discouraged, as they bypass RAC load balancing and introduce operational and security risks.

The following RMAN example demonstrates a multi-node RAC backup configuration:

```
# The rac_backup_svc service is configured to run across all RAC instances
rman target /@rac_backup_svc

# Example (conceptual): when connected through a multi-instance RAC service, channels are typically
distributed across RAC instances.

RUN {
  -- Conceptual example: channels are distributed across RAC instances by the RAC service
  ALLOCATE CHANNEL ch1 DEVICE TYPE DISK; -- example: instance/node 1
  ALLOCATE CHANNEL ch2 DEVICE TYPE DISK; -- example: instance/node 1
  ALLOCATE CHANNEL ch3 DEVICE TYPE DISK; -- example: instance/node 1
  ALLOCATE CHANNEL ch4 DEVICE TYPE DISK; -- example: instance/node 1
  ALLOCATE CHANNEL ch5 DEVICE TYPE DISK; -- example: instance/node 2
  ALLOCATE CHANNEL ch6 DEVICE TYPE DISK; -- example: instance/node 2
  ALLOCATE CHANNEL ch7 DEVICE TYPE DISK; -- example: instance/node 2
  ALLOCATE CHANNEL ch8 DEVICE TYPE DISK; -- example: instance/node 2

  BACKUP
    SECTION SIZE 4G
    DATABASE
    PLUS ARCHIVELOG;
}
```

Note: Actual channel-to-instance placement is managed dynamically by Oracle RAC services and may vary based on load and availability.



Recovery options

Single-node restore: Run RMAN from one node and restore proceeds at single-node throughput. This is appropriate when RTO allows or when only one node is available.

Multi-node RAC restore: RMAN channels are distributed across multiple RAC nodes to parallelize the restore operation. This approach is critical for meeting aggressive RTOs on large databases. The following RMAN example demonstrates a multi-node RAC restore using a service configured across all RAC instances:

```
# The rac_restore_svc service is configured to run across all RAC instances

rman target /@rac_restore_svc

# Example (conceptual): when connected through a multi-instance RAC service, channels are typically
distributed across RAC instances.

RUN {

  -- Channels are distributed across RAC instances by the RAC service

  ALLOCATE CHANNEL ch1 DEVICE TYPE DISK; -- example: instance/node 1
  ALLOCATE CHANNEL ch2 DEVICE TYPE DISK; -- example: instance/node 1
  ALLOCATE CHANNEL ch3 DEVICE TYPE DISK; -- example: instance/node 1
  ALLOCATE CHANNEL ch4 DEVICE TYPE DISK; -- example: instance/node 1
  ALLOCATE CHANNEL ch5 DEVICE TYPE DISK; -- example: instance/node 2
  ALLOCATE CHANNEL ch6 DEVICE TYPE DISK; -- example: instance/node 2
  ALLOCATE CHANNEL ch7 DEVICE TYPE DISK; -- example: instance/node 2
  ALLOCATE CHANNEL ch8 DEVICE TYPE DISK; -- example: instance/node 2

  RESTORE DATABASE;

  RECOVER DATABASE;

}
```

Note: Example placement is illustrative only; actual channel-to-instance assignment is dynamic and managed by Oracle RAC services.



When to use each scenario

In RAC environments, RMAN operations can use one or multiple nodes depending on recovery objectives and cluster availability; Oracle RAC services manage how work is distributed. Table 2 outlines when to use single-node vs. multi-node approaches.

Scenario	Approach
Single-instance database	Single-node backup/restore
RAC database, backup window comfortable	Single-node backup from one RAC instance
RAC database, tight backup window, maximum throughput required	Multi-node RAC backup using a RAC service
RAC database, relaxed RTO, limited node availability	Single-node restore from one available RAC instance
RAC database, aggressive RTO, multiple nodes available	Multi-node RAC restore using a RAC service
Partial cluster outage, limited nodes available	Restore using available RAC nodes only

TABLE 2 Recommended RMAN backup and restore approaches based on scenarios

Considerations

The following considerations help determine how RMAN should operate across RAC nodes:

- **Node availability:** Multi-node backup or restore operations require multiple RAC instances to be available at runtime.
- **Throughput scaling:** Backup and restore throughput increases as additional RAC instances participate, assuming adequate network and storage bandwidth.
- **Operational simplicity:** Single-node operations reduce complexity and are appropriate when backup windows or RTOs are relaxed.
- **RAC services:** Multi-node operations assume connection through a RAC service that allows Oracle to distribute RMAN channels across instances.

Pattern C: Incremental merge (single instance and RAC)

Use case: Very large databases (single instance or RAC) where full backup windows are impractical or disruptive to production workloads.

Problem addressed: A 100TB database requires 25TB/hour or more throughput to complete a full backup in four hours. Even with sufficient backup throughput, the I/O load on production storage during a full backup scan can impact application performance. As databases grow, full backups become operationally untenable without offloading work from primary storage.

Solution: Eliminate full backups after an initial seed. Use RMAN incremental merge to maintain a continuously updated image copy on FlashBlade. Daily incremental merge backups capture only changed blocks, reducing capacity consumed, backup duration, and production storage I/O.



How it works:

1. **Initial setup:** Create a level 0 image copy (equivalent to a full backup).
2. **Daily operations:** Take a level 1 incremental backup (changed blocks only).
3. **Merge:** Apply the incremental backup to the image copy, updating it in place.
4. **Result:** The image copy is always current and ready for instant recovery.

Considerations:

- **Block change tracking** (only available with Oracle Enterprise edition): This is required for practical incremental performance on large databases. Without block change tracking, RMAN must scan all blocks to identify changes.
- **Archive log retention:** Recovery requires archive logs from the time of the image copy to the present. Ensure archive logs are retained and accessible.
- **Merge I/O:** The merge operation reads and writes the image copy on FlashBlade. This does not impact production storage but consumes FlashBlade bandwidth. Schedule merges during low-activity periods if concurrent backup throughput is a concern. Figure 6 provides a workflow for the incremental merge lifecycle and backup planning.

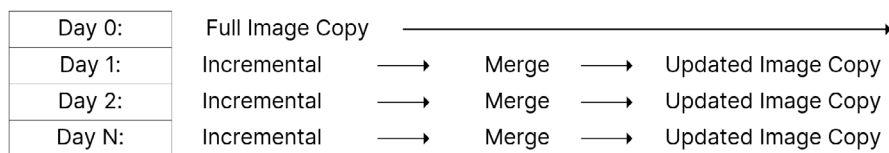


FIGURE 6 Incremental merge lifecycle and backup planning workflow

Initial configuration

Enable block change tracking for efficient incremental identification as follows:

```
ALTER DATABASE ENABLE BLOCK CHANGE TRACKING USING FILE '/u01/app/oracle/bct/bct.dbf';
```

Create the initial image copy

Create an image copy and perform daily incremental updates as follows:

```
BACKUP AS COPY
  INCREMENTAL LEVEL 0
  DATABASE
  FORMAT '/rman/image_copy/%U'
  TAG 'incr_update';

Daily incremental backup and merge:
RECOVER COPY OF DATABASE WITH TAG 'incr_update';

BACKUP
  INCREMENTAL LEVEL 1
  FOR RECOVER OF COPY WITH TAG 'incr_update'
  DATABASE;
```

The RECOVER COPY command merges the previous day's incremental backup into the image copy, and the BACKUP INCREMENTAL command creates a new incremental backup. Run both commands daily, in this order.



Recovery from image copy

Recovery is fast because no restore is required. Switch to the image copy and apply archive logs as follows:

```
RUN {  
    SET NEWNAME FOR DATABASE TO '/rman/image_copy/%U';  
    SWITCH DATABASE TO COPY;  
    RECOVER DATABASE;  
}  
ALTER DATABASE OPEN RESETLOGS;
```

Production I/O impact

Table 3 summarizes how RMAN backup types read data from production storage.

Backup type	Blocks read from production
Full backup	All allocated blocks
Incremental (with block change tracking)	Only changed blocks since last backup

TABLE 3 Production I/O impact for full and incremental backups

For a 100TB database with a 2% daily change rate, the incremental backup reads approximately 2TB instead of 100TB.

Periodically validate the image copy as follows to detect corruption:

```
VALIDATE COPY OF DATABASE WITH TAG 'incr_update';
```



Performance validation

This section presents measured performance from testing Oracle RMAN backup and restore operations on FlashBlade. All tests used Oracle 19c with dNFS enabled. All performance results are established by measuring the duration of RESTORE DATABASE (where the data files are copied from backup to primary storage) divided by the database size.

Environment

Table 4 details the RAC configuration.

Component	Detail
Compute	<p>Servers: Cisco UCS C480 M5 (4-socket)</p> <p>CPU: 4x Intel Xeon Platinum 8176 at 2.10GHz (112 cores/224 threads)</p> <p>Memory: 1.5TiB RAM/NUMA: 8 nodes</p>
Operating system	<p>Oracle Linux Server 8.10</p> <p>Kernel: UEK 5.4.17</p>
Network	<p>2x Mellanox ConnectX-6 Dx 100GbE NICs</p> <p>Bonding: IEEE 802.3ad LACP, layer 3+4 hash</p> <p>Aggregate bandwidth: 200Gbps</p>
Primary storage	Pure Storage FlashArray//XL170 R5 (8x32Gb Fibre Channel)
Backup storage	<p>FlashBlade//E*, //S100, //S200</p> <p>Protocol: NFSv4.1 with dNFS, with 1 NFS share and 1 VIP</p> <p>//S100 and //S200 configuration: 10 blades, 4 DirectFlash® Modules per blade</p>
Oracle Database	<p>Oracle Database 19c Enterprise Edition</p> <p>dNFS: Enabled (ODM Library Version 6.0)</p> <p>RAC: 2-node cluster</p> <p>Database size: 10TB in a single bigfile tablespace</p>

TABLE 4 RAC configuration



Table 5 details the standalone instance configuration.

Component	Detail
Compute	Servers: 8x Cisco UCS C220 M5 (2-socket) CPU: 2x Intel Xeon Platinum 8160 at 2.10GHz (48 cores/96 threads) Memory: 512GB RAM
Operating system	Oracle Linux Server 8.10 Kernel: UEK 5.4.17
Network	2x 25GbE NICs per host Bonding: IEEE 802.3ad LACP, layer 3+4 hash Aggregate bandwidth: 50Gbps
Primary storage	2x Pure Storage FlashArray//XL170 R5 (8x32GB Fibre Channel) (4 instances per array)
Backup storage	FlashBlade//E, //S100, //S200 Protocol: NFSv4.1 with dNFS, 1 NFS share with dedicated directories for each instance //S100 and //S200 configuration: 10 blades, 4 DirectFlash Modules per blade
Oracle Database	Oracle Database 19c Enterprise Edition dNFS: Enabled (ODM Library Version 6.0) Database size: 1TB per instance in a single bigfile tablespace

TABLE 5 Standalone instance configuration

RMAN parameter optimization

Testing explored the interaction between RMAN's parallelism parameters to identify optimal settings for FlashBlade. The test matrix varied channels (8–128), section size (2G–32G), and FilesPerSet (1–8) systematically on a 10TB database.

With SECTION SIZE enabled, varying the section size value (2G–32G) and FilesPerSet (1–8) had minimal impact on throughput. Channel count proved to be the primary throughput driver, with single-host throughput reaching a ceiling of approximately 8–10 GB/s (~30–35TB/hour) regardless of additional parallelism.

Key finding: For single-node operations, 64 channels with 4G section size optimizes backup throughput. Restore throughput is less sensitive to configuration but benefits from adequate channel allocation.

Backup performance

Channel scaling determined backup throughput. Testing increased channel count while holding section size (4G) and FilesPerSet (1) constant, as shown in Figure 7.



Backup Performance/ Channel Scaling

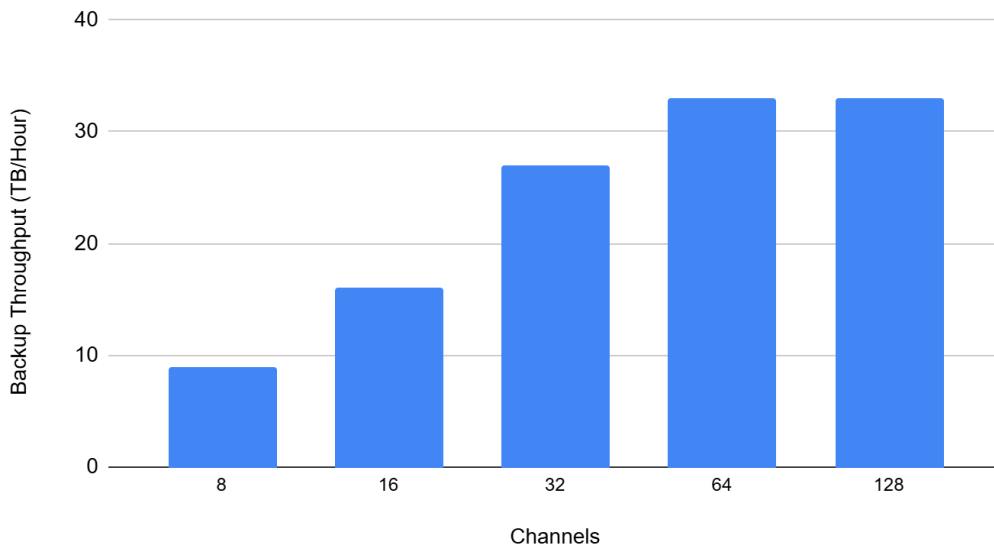


FIGURE 7 Backup channel scaling

Backup throughput scaled linearly with channel count up to 64 channels, then plateaued as coordination overhead offset additional parallelism.

Restore performance

Restore testing revealed that throughput depends primarily on restore-time channel count, not on how the backup was created. Backups created with different configurations (varying channels, section sizes, and FilesPerSet) restored at similar speeds when using the same restore channel count, as shown in Figure 8.

Restore Performance / Channel Scaling

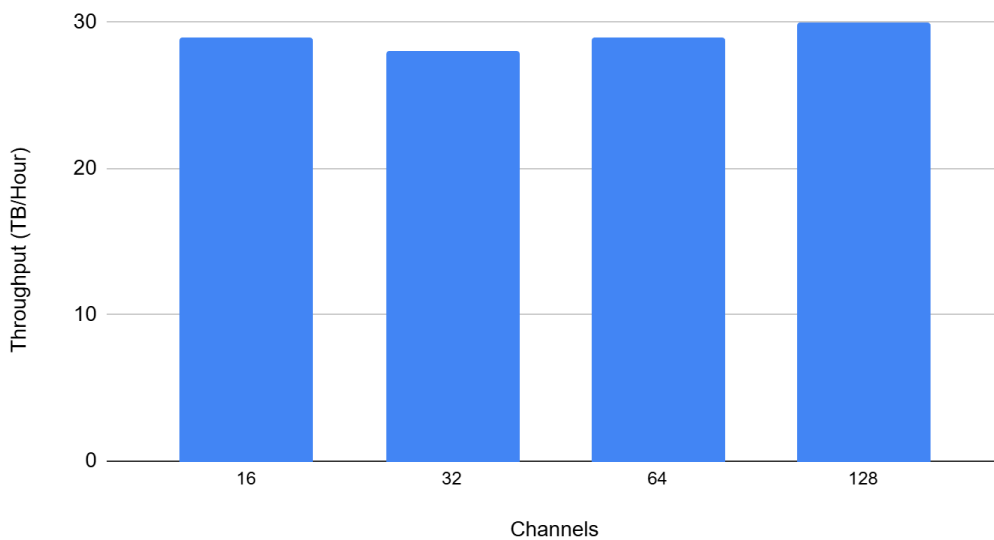


FIGURE 8 Restore channel scaling

Restore throughput remained consistent across channel counts, indicating the bottleneck was elsewhere in the stack (source storage, CPU, or network) rather than RMAN parallelism.



RAC: Distributed backup and recovery

Single-node RMAN operations are constrained by the throughput ceiling of one database server. For the test environment, this ceiling was approximately 8–10 GB/s (~30–35TB/hour) regardless of channel count or FlashBlade model.

Distributing RMAN channels across RAC nodes breaks through this limit. Each node contributes its own network bandwidth, CPU, and dNFS connections. With 64 channels per node (128 total), restore throughput scaled beyond what any single node could achieve.

Testing compared single-node operations (128 channels on one RAC node) against distributed operations (64 channels per node, 128 total across two nodes).

Figure 9 shows the restore throughput for a single node vs. two-node RAC cluster.

Distributed Recovery Performance

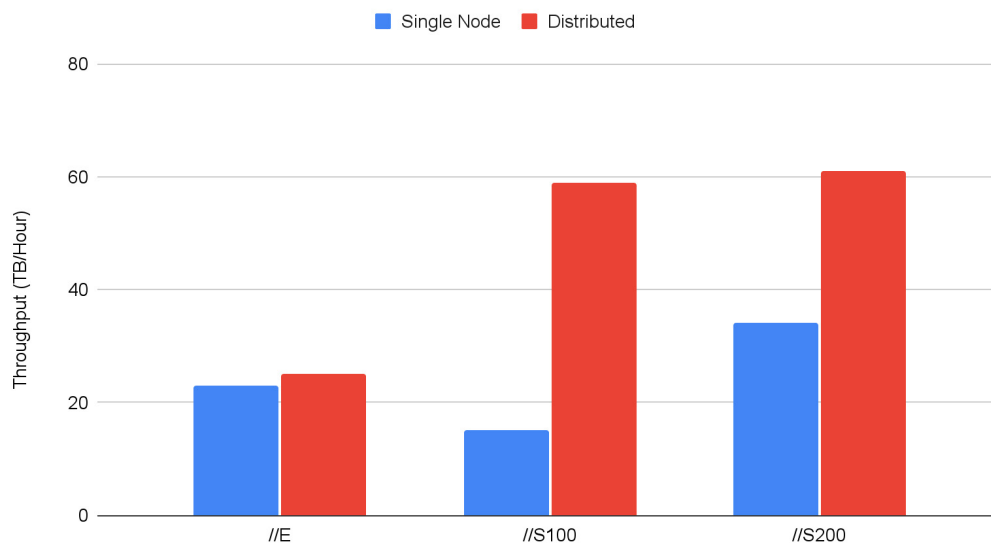


FIGURE 9 Distributed recovery performance

The FlashBlade//S™ models—FlashBlade//S100 and FlashBlade//S200—both achieved approximately 60TB/hour with distributed restore, breaking through the single-server ceiling by aggregating I/O across both RAC nodes.

Table 6 shows the recovery time for a 10TB database.

Model	Throughput	Restore time
//E	20TB/hour	~18 minutes
//S100	20TB/hour	~10 minutes
//S200	32TB/hour	~10 minutes

TABLE 6 FlashBlade recovery times for 10TB database



Standalone instances: Consolidation and scale

Testing measured aggregate throughput when running concurrent backups across multiple standalone instances (see Figure 10).

Restore Throughput Scaling by Instance Count and Model

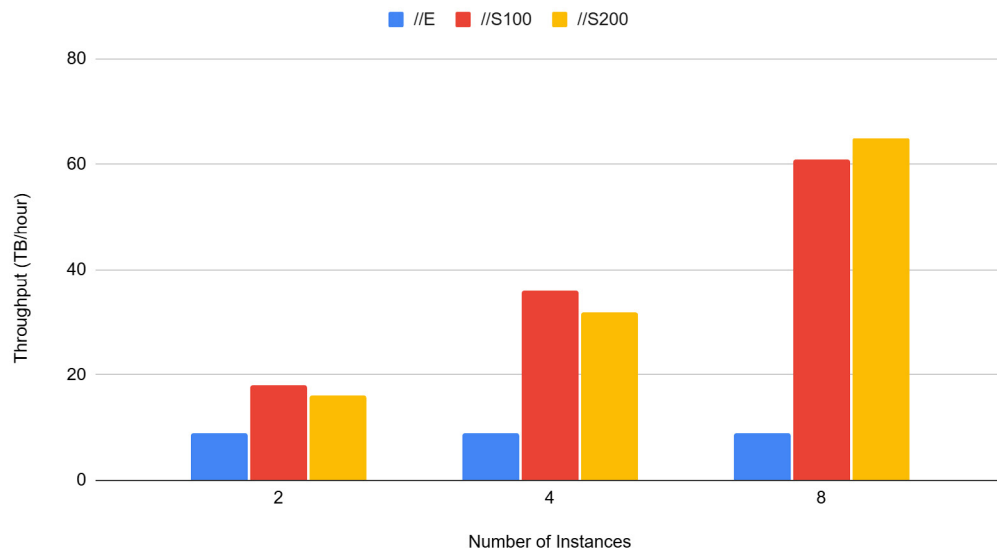


FIGURE 10 Restore throughput testing results

With eight concurrent instances, FlashBlade//S200 delivered 65TB/hour aggregate restore throughput—scaling near-linearly as instances were added. FlashBlade//S100 showed similar scaling. FlashBlade//E™ throughput remained flat regardless of instance count, indicating it was saturated at single-instance load.

Conclusion

Enterprise Oracle environments require backup and recovery infrastructure that keeps pace with database growth. Traditional approaches force compromise between backup frequency, recovery speed, and operational complexity. FlashBlade eliminates these trade-offs.

The validated configurations in this paper demonstrate:

- Single-node throughput of 8–10 GB/s (~30–35TB/hour) with optimized RMAN settings
- Distributed RAC operations exceeding 15GB/s (~55–60TB/hour) for backup and restore
- 10TB database recovery in approximately 10 minutes using distributed channels
- Near-linear scaling when consolidating multiple databases on a single FlashBlade

The symmetric read/write performance of FlashBlade ensures recovery operations run as fast as backups. When an incident occurs, restore throughput matches the infrastructure's full capability rather than degrading to a fraction of backup speed.

For Oracle environments where backup windows are shrinking and recovery SLAs are tightening, FlashBlade provides the throughput foundation to meet both requirements with margin to spare.

[Experience the FlashBlade firsthand with a free test drive.](#)

