ORACLE

# Oracle Real Application Clusters (RAC) Cache Fusion Performance Optimizations on Exadata

March, 2026, Version 2.0

Public

## Purpose statement

This document provides an overview of features and enhancements included in release Oracle AI Database 26ai. It is intended solely to help you assess the business benefits of upgrading to Oracle AI Database 26ai and planning for the implementation and upgrade of the product features described.

## Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. 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, timing, and pricing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

ORACLE

# Table of contents


**Executive Summary** **4**

**List of Performance Optimizations** **5**

Exafusion 5

Zero Copy Block Sends 5

Undo Block RDMA Reads 5

In-Memory Commit Cache 6

Shared Data Block and Undo Header RDMA Reads 6

Broadcast-on-Commit Over RDMA 8

Optimized Object Checkpoints 9

**Conclusion** **9**

**References** **9**


**List of figures**


Figure 1. RDMA-based Cache Fusion protocol 7

Figure 2. SCN message traffic reduction in Oracle Database 21c 8

# Executive Summary

**Oracle Real Application Clusters (Oracle RAC)** is a widely used Oracle Database feature that delivers linear horizontal scalability and high availability. A key component of Oracle RAC is **Cache Fusion**, which synchronizes caches across multiple RAC instances. This enables applications to seamlessly leverage the combined computing resources of all RAC instances without requiring any modifications. Cache Fusion uses a dedicated private network to synchronize data, making application scalability dependent on the latency and bandwidth of this underlying network infrastructure.

Exadata incorporates **advanced networking technologies, such as RDMA over Converged Ethernet (RoCE), which enable Oracle to significantly enhance performance and scalability.** In addition to benefiting from the increased speed of the underlying network, Oracle RAC Cache Fusion has been further optimized to leverage the advanced protocols and RDMA capabilities provided by these networking technologies. This paper explores the optimizations available to Oracle databases deployed on Exadata.

# List of Performance Optimizations

## Exafusion

Traditionally, Oracle RAC messaging relied on a standard networking model using network sockets, where all communication passed through the operating system kernel. This approach required context switches and memory copies between user space and the OS kernel for every RAC message, resulting in additional overhead.

Exafusion, available on Exadata since Oracle Database 12c (for both RoCE and InfiniBand), introduces next-generation networking by enabling **direct-to-wire messaging** from user space—**completely bypassing the OS kernel**. Eliminating context switches and kernel overhead allows Exafusion to process round-trip messages in less than 30 microseconds, which is **up to five times faster than traditional socket-based implementations found on generic servers**. Additionally, Exafusion reduces the CPU cost of sending and receiving messages, supporting higher block transfer throughput and increasing the capacity of Cache Fusion server background processes (LMS processes) before saturation occurs.

**Faster messaging improves not only application runtime performance but also accelerates all Oracle RAC operations**—including dynamic lock redirection (DRM), RAC reconfiguration (due to instance or PDB membership changes), and instance recovery.

The adoption of Exafusion forms the foundation for further performance optimizations for RAC on Exadata, such as zero-copy block sends and RDMA adoption.

Notably, Exafusion and its subsequent optimizations do not require additional operating system resources to operate.

## Zero Copy Block Sends

RoCE and InfiniBand network adapters support zero-copy messaging, where user space buffers are registered directly with the Host Channel Adapter (HCA). The HCA is then able to transmit the contents of these buffers directly onto the network, bypassing the OS kernel and avoiding the extra memory copy step required by traditional messaging protocols. This elimination of CPU cycles associated with buffer copying further reduces RAC Cache Fusion transfer latencies on Exadata.

## Undo Block RDMA Reads

When transaction rollbacks or similar operations require undo blocks to be fetched from other Oracle RAC instances, Exadata introduces an important optimization by utilizing an RDMA-based transfer protocol instead of the traditional messaging-based approach. **With RDMA, foreground processes can directly read undo blocks from another instance's SGA** without involving processes on the remote instance. This eliminates server-side CPU usage and context switch overhead that are present in RAC deployments on non-Exadata systems.

As a result, undo block transfers are no longer affected by process congestion or overall CPU load on the remote instance, **ensuring consistently low and predictable read latencies—even during periods of high system load or instability** in the cluster. RDMA-based reads typically finish in **less than 10 microseconds**, representing a **threefold improvement** over the best latencies achieved using the messaging-based protocol.

## In-Memory Commit Cache

Applications with long-running batch jobs and concurrent queries often generate a high volume of "undo header" consistent read (CR) block transfers. To address this, Exadata implements an **in-memory commit cache**. With this feature, each instance maintains a cache of local transaction states (committed or not) in its SGA, which can be accessed remotely by other instances. This remote lookup is significantly faster than transferring 8 KB undo header blocks between instances. Furthermore, the state of multiple transaction IDs (XIDs) can be checked in a single message, reducing both the number of roundtrip RAC messages and the CPU overhead on LMS processes responsible for servicing commit cache lookup requests.

With the in-memory commit cache, **up to 30 XID lookups can be batched into a single roundtrip message**, replacing what would previously have required 30 separate 8 KB block transfers. As a result of this optimization, many of the "*gc cr block 2-way*" waits associated with undo header transfers are expected to be replaced by fewer "*gc transaction table 2-way*" waits (renamed from "*gc transaction table*" in releases prior to Oracle AI Database 26ai). Each "*gc transaction table 2-way*" wait now represents a batched lookup of multiple XIDs in a single message.

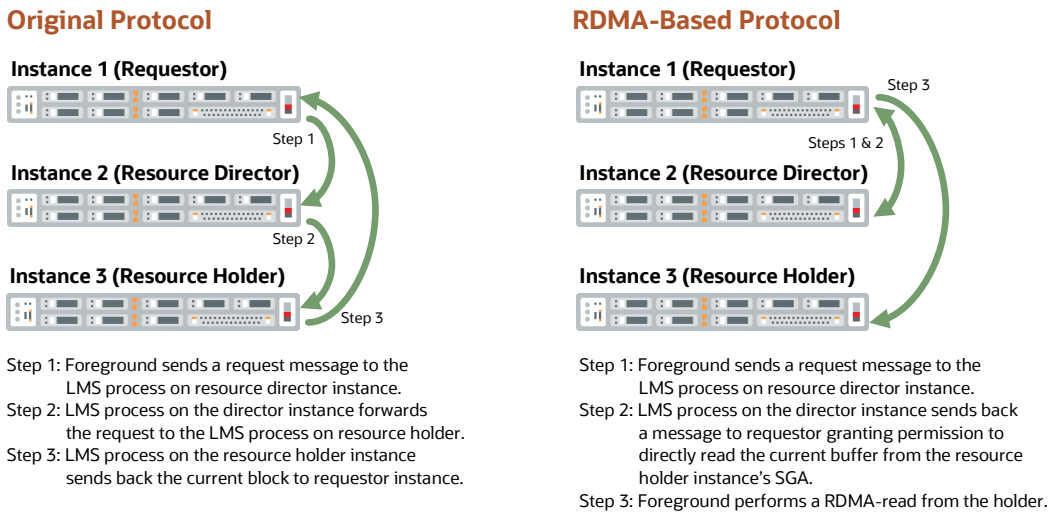
## Shared Data Block and Undo Header RDMA Reads

Starting with Oracle Database 21c, **RDMA support for Cache Fusion has been extended to include reads for data blocks, space blocks, and undo header blocks**. Like the Undo Block RDMA Reads optimization, this enhancement enables faster access to data cached in other instances and further reduces LMS CPU usage, as the LMS process is not invoked when data is read via RDMA.

Traditionally, when a foreground process needed to read a block, it sent a request to the director instance. The director would then forward the request to the holder instance, which ultimately fulfilled the request in a three-way Cache Fusion transfer ("*gc current block 3-way*"). This is a common access pattern in read-intensive OLTP workloads, especially on large clusters with three or more nodes. In such environments, each instance typically has a smaller cache, making it less likely for the required data to be found locally, but more likely cached elsewhere in the cluster.

With RDMA-enabled reads for data and space blocks, the director instance now responds to the requester with a lock grant (permission to read) and the identity of the holder instance for the requested block. The requesting foreground process can then use RDMA to read the block directly from the holder instance's memory. This removes the director-to-holder messaging step, resulting in lower read latencies and reduced LMS CPU consumption on the holder instance, which no longer needs to send the block back to the requester.

Figure 1. RDMA-based Cache Fusion protocol




With RDMA-enabled reads, the foreground process encounters a new sequence of wait events, replacing the traditional "*gc current block 3-way*" wait. The sequence is as follows:

- A "*gc current grant 2-way*" wait, followed by
- A brief "*gc current block direct read*" wait.

The "*gc current block direct read*" waits are **typically less than 10 microseconds**, and the combined duration of the grant and read is generally shorter than that of a traditional three-way transfer.

If the requesting instance is also the director, the "*gc current grant 2-way*" wait can be eliminated, as the instance can immediately grant itself permission to read the data without any additional messaging. In such cases, the request is quickly fulfilled by a single "*gc current block direct read*" wait. This optimization also replaces some of the "*gc current block 2-way*" waits commonly seen in Oracle RAC, including those in two-node clusters.

Additionally, if a non-local director instance happens to be the holder of the data block, the LMS process on that instance will respond with a grant message, after which the requester can use RDMA to read the data directly from the holder (who is also the director). Here, the traditional "*gc current block 2-way*" wait is replaced by a "*gc current grant 2-way*" and "*gc current block direct read*." While RDMA may not significantly reduce read latency in this scenario, **it does lower LMS CPU usage because granting a lock requires less overhead than sending a data block back to the requester**.

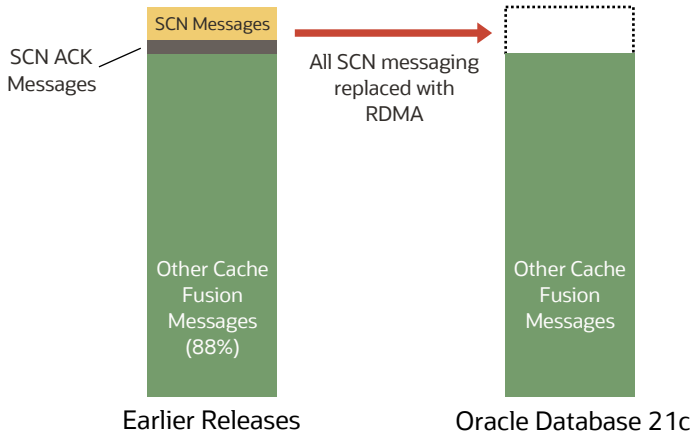
## Broadcast-on-Commit Over RDMA

Before a transaction is committed, the Broadcast-on-Commit protocol ensures that the system change number (SCN) on all cluster instances is at least as high as the commit SCN. This process guarantees Oracle's consistent read (CR) property for transactions. Traditionally, the protocol worked by broadcasting the commit SCN from the LGWR process to the LMS processes on all instances. Each LMS process would update its local SCN upon receipt and send back an acknowledgment (SCN ACK) to the initiating instance. After the redo I/O completes, LGWR verifies that all instances have acknowledged the redo SCN before notifying foreground processes that the commit has completed. If any acknowledgments are missing at the time of I/O completion, the commit will be delayed, potentially increasing "*log file sync*" wait times for foreground sessions.

**In Oracle Database 21c, the Broadcast-on-Commit protocol has been optimized to use RDMA.** By leveraging RDMA, SCN broadcasts now incur lower latency and impose less load on LMS processes—an improvement particularly valuable in OLTP environments and clusters with many instances, where SCN messages make up a significant portion of messaging traffic. While these messages are seldom the direct cause of commit latency (since redo I/O usually dominates), reducing their overall volume lessens LMS workload, providing more headroom for the system to absorb load spikes and support higher throughput.

For example, measurements on a large CRM (OLTP) workload running on a three-instance cluster showed that SCN broadcasts accounted for 12% of total RAC messages. With RDMA, these broadcasts no longer require LMS process involvement, reducing both messaging overhead and CPU consumption.

Figure 2. SCN message traffic reduction in Oracle Database 21c




**In Broadcast-on-Commit over RDMA mode, the LGWR process directly updates the SCN on each instance in the cluster using atomic RDMA operations.** This makes the commit protocol faster since it bypasses the LMS process, removing it as a potential bottleneck caused by context switch latency or CPU load on other instances.

## Optimized Object Checkpoints

Workloads that perform frequent Exadata Smart Scans or parallel query (PQ) scans can encounter performance bottlenecks related to object checkpoints. Typical symptoms include elevated wait times for "*enq: KO - fast object checkpoint*" and "*reliable message*" events. This is particularly pronounced in applications that run very short scans at high concurrency, where object checkpoints are requested at a high rate.

**In Oracle AI Database 26ai, a quick RDMA-based check has been introduced to determine if an object checkpoint is required across the cluster.** If no dirty buffers are found in the global cache for the object, the object checkpoint is skipped entirely. This check occurs under the "*gc obj ckpt direct read*" wait event and **typically completes in less than 10 microseconds, without invoking any background processes in the cluster**.

This optimization greatly enhances Exadata Smart Scan performance for Vector Database workloads in Oracle AI Database 26ai, as AI Vector Search scans are often extremely short. An internal study demonstrated that this approach could eliminate 99% of object checkpoints for AI Vector Search workloads.

## Conclusion

This document highlights several ways in which Oracle RAC leverages the advanced networking capabilities available on Exadata to further optimize Cache Fusion performance, resulting in substantial application scalability improvements—all without requiring any changes to applications. With Exadata, Oracle continues to drive innovation by engineering database software to fully exploit the latest hardware advancements, delivering significant productivity gains for Oracle customers.

## References


- The New Generation Oracle RAC
- Oracle RAC Internals – The Cache Fusion Edition
- Oracle RAC features on Exadata

ORACLE

## Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle