

FEATURE COMPARISON & OVERVIEW

# GridGain 8 vs. Oracle Coherence

## Overview

This document provides a comparison between GridGain 8 and Oracle Coherence, including an analysis of features, strengths, weaknesses, and trade-offs. Both are in-memory data grid platforms designed for high-performance, scalable, and resilient data processing.

For many organizations, Oracle Coherence has long served as a reliable in-memory caching layer for Java applications tightly coupled to Oracle databases. However, as architectures evolve toward cloud-native, elastic, and data-intensive systems, the limitations of a cache-only, Java-centric platform become increasingly apparent.

GridGain was designed to address these modern requirements by unifying in-memory data, distributed SQL, transactions, and compute in a single, memory-first data platform. Unlike Coherence, which remains optimized for static caching use cases, GridGain enables teams to move beyond cache acceleration toward real-time analytics, transactional workloads, and compute-on-data—without introducing additional databases or operational complexity

## Advantages of GridGain vs. Coherence

- **Unified data + compute platform**
- **ANSI SQL with HTAP Support**
- **Elastic, cloud-native architecture**
- **Built-in durability and fast restart**
- **Modern use case coverage**
- **Reduced licensing and infrastructure costs**
- **Lower operational overhead**
- **Higher developer productivity**
- **Demonstrated business impact**

## KEY DIFFERENTIATORS

GridGain consistently outperforms Coherence when applications evolve beyond simple caching into data-centric and real-time workloads:

### Unified Data + Compute Platform

GridGain combines in-memory data storage with distributed compute, enabling SQL queries, aggregations, joins, and application logic to execute directly where the data resides. Coherence remains a Java cache with limited, entry-level processing.

### ANSI SQL with HTAP Support

GridGain supports true ANSI SQL, including distributed joins and aggregations, alongside ACID transactions and HTAP workloads. Coherence's CohQL is limited to simple, read-heavy queries and lacks cluster-wide transactional guarantees.

### Elastic, Cloud-Native Architecture

GridGain supports dynamic node scaling with automatic data rebalancing, making it well-suited for Kubernetes and cloud environments. Coherence scaling is static, manual, and not optimized for frequent topology changes.

### Built-in Durability and Fast Restart

GridGain provides native persistence with WAL, checkpoints, and on-disk indexes, allowing data to survive full cluster restarts and enabling database-like durability without an external system. Coherence relies on external databases or snapshot mechanisms that do not provide true system-of-record guarantees and require lengthy cache rehydration.

### Modern Use Case Coverage

GridGain excels in real-time analytics, event-driven microservices, systems of record, cloud-native scaling, and compute-on-data scenarios—areas where Coherence encounters architectural or operational constraints.

## GRIDGAIN'S LOWER COST OF OWNERSHIP

While Coherence is often perceived as "free" due to Oracle bundling, real-world deployments typically incur higher infrastructure, operational, and opportunity costs. GridGain reduces total cost of ownership across licensing, operations, and developer productivity:

### Reduced Licensing and Infrastructure Costs

GridGain eliminates dependencies on Oracle RAC, external databases for persistence, and bundled WebLogic/Coherence licensing. Its elastic architecture allows infrastructure to scale efficiently with demand, rather than over-provisioning static clusters.

### Lower Operational Overhead

GridGain automatically rebalances data and compute as nodes are added or removed, reducing the need for manual intervention and specialized operational expertise required by Coherence environments.

### Higher Developer Productivity





With SQL-first access, polyglot APIs, native integrations (Kafka, Kubernetes, JDBC/ODBC), distributed compute, and ACID transactions, GridGain enables faster development cycles and simpler architectures compared to Java-only cache patterns.






### Demonstrated Business Impact

Organizations adopting GridGain report 30–50% cost savings, improved reliability under load, faster scaling, and reduced architectural complexity compared to Coherence-based solutions.

# GridGain and Coherence Feature Comparison

The following table provides a feature comparison for GridGain 8.9 and Oracle Coherence 15.1

Capability	GridGain 8	Oracle Coherence
<b>Core Data Grid / Caching</b>	Full in-memory data grid, partitioned and replicated caches; supports SQL and key-value models.	Distributed caching, partitioned and replicated caches, near and local cache support.
	 Coherence is more traditional for caching workloads; GridGain adds advanced SQL and compute features.	
<b>ANSI SQL Support</b>	ANSI SQL:99.	Partial: Limited CohQL, simple selects.
	 GridGain has powerful search and leverages existing skillsets.	
<b>Distributed Joins/ Aggregations/ multi-entity queries</b>	Yes. Distributed multi-table joins across partitions. Query optimizer, indexes, and SQL execution engine. GridGain works as a distributed RDBMS.	Manual: Entry processors. Does not support SQL joins. Data is stored as key-value entries, no native join engine. No cross-partition relational operations.
	 GridGain simplifies application logic + delivers higher performance. Coherence requires any join-like logic be in the app code or using filters on individual entries.	
<b>Native Persistence</b>	Native and 3rd party persistence. Durable, hierarchical, and scalable. Built-in for high-volume, low latency transactions. Can function as a system of record.	No native data persistence, no ACID-grade persistence. Cannot persist data to disk by itself. Saves cache content only (snapshots); not a durable DBMS, Oracle DB is the system of record.
	 GridGain offers consistency and faster restart.	

Capability	GridGain 8	Oracle Coherence
Hybrid Transactional-Analytical Processing	<p>Yes.</p> <p> GridGain can deliver real-time decisions.</p>	No.
Data locality	<p>Yes. Related data records can be directed to be stored together on the same nodes. Thus, processing of related data avoids unnecessary network traffic and data movement.</p> <p> GridGain has a fully distributed resilient compute engine that runs logic across entire partitions of data, while Coherence can only modify one key/value pair at a time.</p>	Partially. Entry processor executes small, atomic functions on a single key/value entry. Can combine with filters.
Clustering	<p>Elastic. Add/delete nodes based on demand. System automatically rebalances data.</p> <p> GridGain's elasticity won't impact performance.</p>	Static. Node add/removes not automatic. Resource-intensive rebalancing.
Integration & Ecosystem	<p>Integrates with Kafka, Spark, MariaDB, PostgreSQL, AI/ML systems, and cloud services.</p> <p> GridGain offers open integration strategy with modern microservices and cloud platforms.</p>	Tight Oracle integration (WebLogic, Oracle DB, GoldenGate HotCache).
Streaming & Event Processing	<p>Continuous queries, streaming ingestion, and real-time transformations.</p> <p> Both support event-driven models; GridGain's integration with stream processing is tighter.</p>	Continuous queries, events, pub/sub topics, and listener-based eventing.

Capability	GridGain 8	Oracle Coherence
<b>Licensing &amp; Cost</b>	Open-core model; enterprise license for advanced features; cost-effective scaling.	Proprietary Oracle licensing; per-core or per-processor pricing.
	 GridGain scales more economically; Coherence can be costly for large deployments.	
<b>Transactions / Consistency</b>	ACID transactions, optimistic/pessimistic concurrency, distributed locking.	Transactional operations with write-behind and concurrency control.
	 GridGain's distributed transactions scale better for hybrid workloads; Coherence is stable but older design.	
<b>Multi-site / Federation</b>	Supports cross-datacenter deployment with enterprise extensions.	Built-in federated caching, active-active and hub-spoke modes.
	 Coherence has more mature multi-site replication; GridGain is improving through enterprise extensions.	
<b>Client APIs / Language Support</b>	Java, .NET, C++, Python, JDBC, REST, thin clients.	Java, C++, .NET, REST, JCache, gRPC proxy for polyglot clients.
	 Coherence recently added gRPC; GridGain has broader open-source language coverage.	
<b>Management &amp; Observability</b>	Unified CLI tool, Control Center UI, integration with Prometheus and Grafana.	CLI, JMX, and metrics integration with Oracle Enterprise Manager.
	 GridGain offers modern DevOps-friendly tooling; Coherence fits Oracle's traditional stack better.	

## SQL SUPPORT WITHOUT FRICTION

GridGain distinguishes itself through robust, out-of-the-box ANSI-99 SQL support, including distributed joins and standard JDBC/ODBC connectivity. This allows SQL-based applications to integrate seamlessly without extensive code rewrites.

In contrast, Oracle Coherence utilizes CohQL—a SQL-like language that lacks full relational capabilities like subqueries, positioning it primarily as a cache query platform.

These approaches create divergent implementation paths: GridGain lets developers treat caches as familiar SQL tables, simplifying database offload for SQL-centric systems, whereas Coherence typically necessitates custom key/value access patterns and application-side logic. Consequently, GridGain is often better suited for accelerating existing SQL operational workloads with minimal friction.

## DISTRIBUTED & MULTI-ENTITY QUERIES

GridGain and Coherence both leverage data affinity to optimize performance, but their query models differ significantly.

GridGain offers native distributed SQL joins and affinity-based colocation, enabling architects to execute complex operational queries across related entities directly through SQL without heavy application-side orchestration. Conversely, while

Coherence provides robust querying, indexing, and entry-processing capabilities; it lacks a relational join engine. Consequently, Coherence typically requires developers to design multi-entity operations via custom keys or application logic.

GridGain is the superior choice when the primary goal is performing live SQL queries on operational data while maintaining partitioned data performance.

## ACID TRANSACTIONS & CONSISTENCY

GridGain provides a robust distributed key-value transaction model supporting various concurrency modes (optimistic/pessimistic) and standard isolation levels (READ\_COMMITTED, REPEATABLE\_READ, SERIALIZABLE). Its transactions integrate seamlessly with its broader persistence and SQL capabilities.

While Oracle Coherence offers atomic partition-level transactions via EntryProcessor and its Transaction Framework API, it faces significant limitations. Specifically, Coherence's transactional caches lack support for standard database integration, eviction, persistence, and explicit locking.

Consequently, GridGain is better suited for architectures where transactions must coexist with durable storage, SQL-centric access, and comprehensive operational data-platform features.

## INTEGRATIONS

GridGain can be integrated with existing relational databases and external stores as part of a database offload or cache-aside architecture. It can connect applications to hot operational data with SQL, key-value APIs, JDBC, ODBC, REST, and thin clients while keeping the backend system of record in place. For buyers using SQL-based applications, this reduces the amount of application-specific code needed to introduce an in-memory data layer.

Coherence has mature integration patterns, including cache stores, read-through and

write-through designs, WebLogic integration, Oracle stack integration, and a strong Java API model. Those integration patterns are valuable, but they are cache-centric rather than SQL-centric. Developers generally model how data is loaded, queried, updated, and written back through Coherence services and application logic.

The practical difference is that GridGain provides a more database-like access layer for mixed SQL and key-value workloads. Coherence provides a powerful cache integration toolkit. If the buyer's primary goal is application-state caching in an Oracle/Java estate, Coherence can be a good fit. If the goal is database acceleration, SQL offload, mixed API access, and long-term operational data reuse, GridGain has the stronger story.

## **COLOCATED PROCESSING**

Both GridGain and Oracle Coherence support sending computation to the cluster nodes that own the data, reducing network data movement.

Coherence's approach centers on per-entry and per-key operations. Its `EntryProcessor` and `InvocableMap` APIs are well-suited for atomic mutations and transformations on individual cache entries—particularly in Java environments where the API is native and mature.

GridGain's approach integrates colocated compute with SQL execution and distributed joins. You can execute `affinityRun()` or `affinityCall()` to run code on the node owning a specific partition, and that compute can invoke SQL queries, access multiple tables, and process data in parallel across the cluster. This tight coupling between compute and SQL makes GridGain more flexible for analytical or

multi-step transformations that span multiple data partitions.

The practical difference is Coherence excels when your workload is primarily transactional (update single keys atomically). GridGain is broader when you need to combine computation with complex queries or multi-step data processing—work that would otherwise require moving large result sets off the cluster to a client for processing.

## **NATIVE PERSISTENCE, RESTART, AND RECOVERY**

GridGain native persistence is one of the strongest differences versus Coherence. GridGain persists data partitions, indexes, metadata, and write-ahead-log state so the cluster can maintain a durable dataset on disk while keeping hot data in memory. This enables datasets that exceed RAM and supports operational patterns that require restart predictability, durable indexes, and recovery tools.

Coherence persistence provides on-demand, active, asynchronous, and active-backup modes; it can persist cache service data to disk; and it can recover cached data after catastrophic failure or planned restart.

The distinction is scope. Coherence persistence is a set of tools and technologies for managing persistence and recovery of distributed caches. GridGain native persistence is part of a broader memory-plus-disk operational data platform.

Coherence has operational caveats, including the need to shut down the entire cluster to change persistence modes and the fact that transactional caches do not support persistence. GridGain therefore is stronger when persistence, SQL, and transactions are all part of your evaluation.

## STREAMING, CONTINUOUS QUERIES, AND EVENT PROCESSING

GridGain provides a Data Streaming API for high-throughput ingestion and Continuous Queries to react to cache updates without polling. When an update matches a query predicate, your application can execute a handler—which may issue SQL, run compute tasks on the owning node, or write to persistent storage.

Coherence also supports event-driven patterns through cache events, Continuous Query Cache (which maintains a local materialized view of query results), and a topics API for pub-sub messaging.

The difference: GridGain treats ingestion, querying, reactions, and persistence as a single transactional system. A data stream update, a matching continuous query, a compute task, and a durability write can all participate in the same transaction with coordinated failure

## Summary

Oracle Coherence remains a proven solution for traditional, Java-centric caching use cases, particularly within tightly integrated Oracle environments. However, as modern architectures demand real-time analytics, elastic cloud scaling, transactional consistency, and compute-on-data capabilities, its cache-focused design introduces growing limitations.

GridGain addresses these challenges by providing a unified, memory-first data platform that combines in-memory storage, ANSI SQL,

handling. Coherence handles events and cache operations as separate concerns; coordinating them requires application logic.

## SECURITY & GOVERNANCE

Both Coherence and GridGain support TLS, authentication, and role-based authorization. Coherence's security is cache-focused: access control for who can read and modify cached data in Java/WebLogic environments. GridGain adds capabilities for persistent storage: transparent data encryption at rest, audit trails, secure deserialization controls, enterprise directory integration, and multi-tenancy patterns.

The distinction: Coherence secures cached data. GridGain secures persistent operational data—useful when the platform also serves as durable SQL storage, requires compliance auditing, or isolates data between tenants.

distributed compute, ACID transactions, and native persistence. This enables organizations to consolidate caching, analytics, and transactional workloads into a single system while simplifying application logic and reducing operational overhead.

For teams moving beyond simple cache acceleration toward cloud-native, data-intensive, and real-time systems, GridGain delivers broader functionality, greater elasticity, and lower total cost of ownership—making it a more future-ready alternative to Oracle Coherence.

# About GridGain

GridGain® Systems, a leader in in-memory computing and a MariaDB company, provides a distributed computing solution designed to create an AI-Ready Operating Platform. Built on Apache® Ignite™, the GridGain in-memory computing platform addresses the application speed and scalability limitations of digital enterprises, including delivering the low latency required for agentic AI. By maintaining data in memory and using massively parallel processing (MPP) to leverage pooled CPU power, GridGain accelerates application performance by up to 1,000 times. GridGain Systems customers in financial services, fintech, eCommerce, and telecommunications, including American Airlines, American Express, Barclays, Citi, RBC, and UPS, rely on GridGain for use cases that demand high speed and scalability. To learn more, please visit [www.gridgain.com](http://www.gridgain.com).