

GridGain In-Memory Computing Platform Feature Comparison: Redis

This document presents a summary and detailed feature comparison of the [GridGain® in-memory computing platform](#) (GridGain) and Redis® Enterprise for in-memory computing applications.

Major Advantages of GridGain vs. Redis

- Comprehensive in-memory computing platform
- Native ANSI-99 SQL support
- Distributed ACID Transaction Support
- Slides in-Between SQL-Based Applications and RDBMSs with Full Support for SQL
- Cross-Language Support for Collocated Processing (MPP)
- Apache Spark Support for DataFrames, RDDs and HDFS
- Support for Machine Learning and Deep Learning

REDIS AND THE GRIDGAIN IN-MEMORY COMPUTING PLATFORM COMPARED

Redis is an open-source (BSD licensed), in-memory data structure used as a database, cache and message broker, as stated on Redis.io. Redis, when combined with Redis Enterprise (the commercially supported version from Redis Labs), is widely known as one of the most popular caches and key-value stores on the market due to its cost and simplicity for developers. Redis (which we will now use to refer to Redis and Redis Enterprise) is used primarily by developers in applications for session caching, full page cache, message queue applications, leaderboards and counting.

GridGain, built on the Apache® Ignite™ open source project, is an in-memory computing platform that's used as a distributed in-memory data grid, in-memory SQL and key-value database, and stream processing and analytics engine. At first glance, GridGain and Redis seem similar. Both include functionality for caching data in memory. Both can partition data and be scaled out across distributed clusters. But that's where the similarities end.

IN-MEMORY DATA GRID TO ADD SPEED AND SCALE TO EXISTING APPLICATIONS

Redis is primarily an in-memory "cache-aside" key-value cache used by developers to improve the read performance of applications. It requires coding within the application to add the cache, and coding and configuration to keep the data in the cache up to date relative to any underly-

ing data sources. It does not fully support SQL, so relational data must typically be mapped to another model.

GridGain is more than a cache. GridGain can be used as an in-memory data grid slides between and accelerates applications and third-party databases without having to rip out, replace or rewrite the applications or databases. By replacing existing JDBC or ODBC drivers with the GridGain drivers, GridGain can sit in the path of SQL or other queries and supports ACID transactions as an inline read-and write-through cache. It keeps data in sync with the underlying database by updating the cache following a successful commit by the underlying database.

DISTRIBUTED SQL AND KEY-VALUE DATABASE

Redis is also used as a key-value store for application data. Developers appreciate how simple it is to manage data within an application using the Redis APIs.

GridGain can be used as a distributed key-value database. But GridGain can also be used as a distributed SQL database. Its native persistence allows RAM to hold a subset of the full dataset which resides on disk. It provides immediate availability on restart without having to wait for data to load into memory first. GridGain also provides ACID transaction support with pessimistic locking, unlike Redis.

When used as an in-memory data grid or in-memory database, GridGain also provides the ability to partition data based on data affinity and collocate processing with data to improve performance and scalability. GridGain provides built-in

stream processing, analytics and machine learning capabilities that enable companies to deliver new types of applications for digital business, improving the customer experience or real-time regulatory compliance that were either too costly or too complex to deliver in the past. Collocated processing has helped reduce network traffic in deployments by as much as 100x, which is one reason companies have moved from Redis to GridGain.

GridGain Enterprise Edition (EE) adds enterprise-grade security, deployment, and management capabilities needed for most mission critical in-memory data grid applications. The GridGain Ultimate Edition (UE) includes the Enterprise Edition features plus advanced data management and disaster recovery features for using GridGain as an in-memory database.

REDIS AND GRIDGAIN EDITIONS

To compare Redis and Redis Enterprise to Apache Ignite and GridGain, it's important to understand the different available GridGain editions. The GridGain Community Edition (CE) includes the current version of Apache Ignite with LGPL dependencies, as well as bug fixes that have not yet been released in Ignite. The

FEATURE	GRIDGAIN CE 8.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.7	GRIDGAIN UE 8.7	REDIS ENTERPRISE 5.4 (REDIS 5.0)
Native ANSI-99 SQL Support	●	●	●	●
Distributed ACID Transaction Support	●	●	●	● (Only supports optimistic, single shard transactions)
Slide in-between SQL-Based Applications and RDBMSs with support for SQL	●	●	●	● (Requires coding for any app and only supports cache-aside, not read-, write-through)
Cross-Language Support for Collocated Processing (MPP)	● (Supports Multiple Languages, MPP for Java, .NET, C++)	● (Supports Multiple Languages, MPP for Java, .NET, C++)	● (Supports Multiple Languages, MPP for Java, .NET, C++)	● (No support for collocated processing (MPP))
Integration with RDBMSs, NoSQL Databases and Hadoop	● (Out-of-the box support for RDBMSs, NoSQL Databases, HDFS, Spark)	● (Out-of-the box support for RDBMSs, NoSQL Databases, HDFS, Spark)	● (Out-of-the box support for RDBMSs, NoSQL Databases, HDFS, Spark)	● (Requires coding. Limited Spark support. No HDFS support)
Comprehensive In-Memory Computing Solution	● (EE + Multi-datacenter data and disaster recovery management)	● (EE + Multi-datacenter data and disaster recovery management)	● (EE + Multi-datacenter data and disaster recovery management)	● (Not an IMDG, IMDB only has backup, no MPP, limited support for Kafka and Spark, limited ML)
Apache Spark Support for DataFrames, RDDs, HDFS	●	●	●	● (Online supports Spark RDD)
Built-in Machine Learning	●	●	●	● (Redis ML Module implements some machine learning, no deep learning)
Built on a Leading Open Source Project	● (Built on Apache Ignite, a top 5 Apache Software Foundation open source project)	● (Built on Apache Ignite, a top 5 Apache Software Foundation open source project)	● (Built on Apache Ignite, a top 5 Apache Software Foundation open source project)	● (Redis is open source, but mostly supported by Redis Labs)

This document provides a very detailed feature comparison between the various levels of GridGain and Redis Enterprise, highlighting differences between the open source and enterprise-ready products where relevant. But here are the major differences between the two products which should be considered when evaluating either one for any in-memory computing need, as summarized in the above table:

COMPREHENSIVE IN-MEMORY COMPUTING PLATFORM

GridGain supports third-party databases as an IMDG, includes a distributed SQL and key-value hybrid in-memory database that combines memory-centric storage with native persistence, includes extensive streaming analytics support along with Apache Kafka™ and Spark integration, and built-in machine and deep learning with TensorFlow integration. Redis is not an IMDG, only a cache-aside cache. It does not provide a complete IMDB with SQL support or native persistence, provides limited streaming Kafka and Spark support for stream processing, and limited machine learning support only, no deep learning or TensorFlow integration.

As an IMDB, GridGain's persistence is a distributed ACID and SQL-compliant disk store for storing data and indexes on SSD, Flash, 3D XPoint, and other types of non-volatile storages. With persistence enabled, nonvolatile storage houses the full data set and RAM can hold 0-100% of the data and indexes. If a subset of data or an index is not in RAM, it will be used from non-volatile storage. Data in RAM and nonvolatile storage is stored and treated exactly the same way. Any changes are written to a write-ahead log and then to non-volatile storage to ensure low latency. With GridGain, data is immediately available on a cluster startup. It becomes fully operational once all the cluster nodes are interconnected with each other. There is no need to warm up the memory by preloading data from the disk. When used as a database, GridGain also includes centralized backup and recovery, full and incremental snapshots, continuous archiving and point-in-time recovery, network backups, and heterogeneous recovery that enables a cluster to be restarted in a different location on-premise or in the cloud with a different size.

As an IMDB, Redis allows Flash to be used with RAM to cache data, and disk to be used as a storage tier for backup in its paid version Redis Flash. But with Redis, Flash just looks like RAM. RAM and Flash combined must hold the entire data set, and if there's a node failure the data in Flash is lost. To avoid data loss from failures, Redis has to be configured in dual-persistence mode using snapshotting and an Append-Only-File to back up data in RAM and Flash to disk. Upon restart data must all be loaded from disk before Redis can begin to handle requests.

NATIVE ANSI-99 SQL SUPPORT

GridGain supports ANSI-99 compliant SQL, including distributed SQL JOINS for querying, updating, indexing and defining in-memory data. GridGain works with the SQL from an existing application by providing ODBC and JDBC drivers for applications to use in place of their existing drivers for Java, .NET, C++, Python and other languages. Redis does not natively support SQL so developers must code to integrate applications with Redis and map relational data to the Redis key value store.

DISTRIBUTED ACID TRANSACTION SUPPORT

GridGain has full support for ACID transactions, including OPTIMISTIC and PESSIMISTIC concurrency modes, as well as READ_COMMITTED, REPEATABLE_READ, and SERIALIZABLE isolation levels. Redis provides very limited support for client-side OPTIMISTIC transactions and only local transactions within the same shard. In the case of a concurrent update, clients are required to manually retry a transaction.

SLIDES IN-BETWEEN SQL-BASED APPLICATIONS AND RDBMS WITH FULL SUPPORT FOR SQL

GridGain can automatically integrate with all leading RDBMSs including IBM DB2®, Microsoft SQL Server®, MySQL®, Oracle® Database and Postgres®. It also automatically integrates with leading NoSQL databases, such as Apache Cassandra® or MongoDB®, as well as Hadoop via Spark and HDFS. GridGain supports write-through and read-through caching where cache data updates are propagated to the database, and data unavailable in the cache is loaded from the database,

respectively. Redis only supports a cache-aside pattern that sits outside the path. This means developers must add code to integrate Redis to an application and a database, and to keep the cache up to date. The cache-aside pattern can also result in increased loads when refreshing frequently updated data.

CROSS-LANGUAGE SUPPORT FOR COLLOCATED PROCESSING (MPP)

Both GridGain and Redis support multiple languages for developing clients. But Redis has no support for collocated computing. It only allows primitive server-side code execution with the LUA scripting language, along with processing as part of its module extensions. This limits Redis scalability and performance because it requires data to be sent on the network before it can be processed.

GridGain provides general purpose massively parallel processing (MPP) that is used for distributed SQL and machine and deep learning. It also supports user-defined Java, .NET and C++ code. This makes GridGain the only choice for any data intensive applications that cannot wait for data to travel across the network to a client. GridGain is also better suited for lightweight non-Java based clients because it includes a binary protocol that enables client support without requiring a JVM to be deployed with a non-Java client.

GridGain supports automatic data partitioning and replication across the cluster to ensure high availability and data affinity collocation. It automatically rebalances the data in case of any topology changes. While Redis does provide custom partitioning, it is not automatic and does not automatically rebalance upon a node failure or node addition in the cluster. Redis provides sharding with replicas which requires a set of fairly involved manual steps on both clients and servers every time a topology changes and a store cannot change which keys map to which partitions.

APACHE SPARK SUPPORT FOR DATAFRAMES, RDDS AND HDFS

GridGain provides the broadest integration with Spark over any other in-memory computing vendors. It provides native Apache DataFrame, RDD and HDFS support. This integration simplifies the access, writing and saving of data. It also enables state to be shared across Spark

jobs. It can also accelerate dramatically improve SQL performance compared to standalone Apache Spark because unlike Spark, GridGain supports primary and secondary indexes. GridGain also improves overall analytics and machine learning performance by providing access to GridGain’s MPP capabilities, which include built-in distributed joins, machine and deep learning capabilities. Redis only supports Spark RDDs.

SUPPORT FOR MACHINE LEARNING AND DEEP LEARNING

GridGain includes the GridGain® Continuous Learning Framework, built-in machine learning and deep learning and execution with real-time performance on petabytes of data. GridGain provides several out-of-the-box machine learning

algorithms optimized for MPP-style col-located processing including linear and multi-linear regression, k-means clustering, decision trees, k-NN classification and regression. GridGain also includes a multilayer perception and tensorflow integration for deep learning. Developers can develop and deploy their own algorithms across any cluster as well by using the compute grid. With GridGain developers can train models in place against petabytes of data cluster and then execute the built models in production as well as incrementally retrain the models as needed. This eliminates the need to wait for data to be moved into specialized model training infrastructure, which can take hours to move and hours to run.

Redis provides RedisML, an add-on Redis Module which can only execute models that have already been built (trained)

outside of Redis. Redis only supports decision tree classification and regression, linear regression, logistic regression and matrix operation execution. There is no model training, and no deep learning support.

GRIDGAIN AND REDIS DETAILED FEATURE COMPARISON

The following table provides a detailed feature comparison between the GridGain Community, Enterprise, and Ultimate Edition, and Redis. This comparison is based on our best knowledge of the features available at the time this document was created for the product versions indicated.

FEATURE	GRIDGAIN CE 8.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.7	GRIDGAIN UE 8.7	REDIS ENTERPRISE 5.54 (REDIS 5.0)
Use Cases				
In-Memory Data Grid	●	●	●	●
Third party Database Caching and Persistence (Inline)	●	●	●	● (Cache-aside requires invalidation)
SQL Database	●	●	● (+ Multi-datacenter data and disaster recovery management)	●
In-Memory Database	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (In memory key-value Database)
Web Session Clustering	●	●	●	●
Apache Spark Acceleration	●	●	●	●
Hadoop acceleration	●	●	●	●
In-Memory File System (Hadoop Compliant)	●	●	●	●

FEATURE	GRIDGAIN CE 8.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.7	GRIDGAIN UE 8.7	REDIS ENTERPRISE 5.54 (REDIS 5.0)
Third Party Database Support, Persistence				
Automatic support for Leading RDBMSs (Oracle, IBM DB2, Microsoft SQL Server, MySQL, Postgres ...)	●	●	●	●
Automatic integration with Apache Cassandra	●	●	●	●
Inline support for MongoDB	●	●	●	●
Write-Through and Read-Through Caching	●	●	●	●
Write-Behind Caching	●	●	●	●
Auto-Loading of SQL Schema/Data	●	●	●	●
Store Loader (Optimized BulkDB Load)	●	●	●	●
Native Persistence				
Native Persistence	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (Open source - snapshotting and AOF only; enterprise - Redis Pack + Redis Flash)
Stores Superset of Data	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (Open source - backup of RAM; enterprise Flash acts in place of RAM)
Store Indexes on Disk	●	●	● (+ Multi-datacenter data and disaster recovery management)	●
SQL or Key-Value over Disk	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (Enterprise Redis Flash)
Instantaneous Restart (before memory warm-up)	●	●	● (+ Multi-datacenter data and disaster recovery management)	●

FEATURE	GRIDGAIN CE 8.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.7	GRIDGAIN UE 8.7	REDIS ENTERPRISE 5.54 (REDIS 5.0)
Distributed SQL				
SQL Queries	● (Full ANSI-99 Support)	● (Full ANSI-99 Support)	● (Full ANSI-99 Support)	● (Supports Client-Side API Calls, custom query language, third party Redis SQL framework)
Collocated Distributed Joins	●	●	●	●
Non-Collocated Distributed Joins	●	●	●	●
Single Column Indexes	●	●	●	●
Group Indexes	●	●	●	●
Distributed SQL Joins (select * from Person p, Company c where p.c_ id=c.id)	●	●	●	●
Query Consistency	●	●	●	●
Query Fault-Tolerance	●	●	●	●
DML (INSERT, UPDATE, DELETE, MERGE)	●	●	●	● (Over Redis SQL, limited)
DDL (CREATE, DROP, ALTER)	●	●	●	● (Over Redis SQL, limited)
Distributed Queries				
Continuous Queries	●	●	●	●
Predicate-based Queries	●	●	●	●
SQL Drivers				
JDBC Driver	●	●	●	●
ODBC Driver	●	●	●	●
REST API (SQL)	●	●	●	●

FEATURE	GRIDGAIN CE 8.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.7	GRIDGAIN UE 8.7	REDIS ENTERPRISE 5.54 (REDIS 5.0)
Memory Architecture				
On-Heap Memory	●	●	●	● (N/A Written in C)
Off-Heap Memory	●	●	●	● (Written in C)
Off-Heap Indexes	●	●	●	●
Disk as main storage (disk larger than RAM)	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (Disk is only backup. Flash support only extends RAM)
Tiered Storage - On-Heap, Off-Heap and Disk	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (Disk is only backup. Flash support only extends RAM)
ACID Compliant Transactions and Locks				
Atomic Mode (One Operation at a Time)	●	●	●	●
READ_COMMITTED, REPEATABLE_READ, SERIALIZABLE Isolation Levels	●	●	●	●
Deadlock-Free Transactions	●	●	●	●
XA Integration	●	●	●	●
Fault Tolerance (Including Client/Near/Primary/Backup Node Failures)	●	●	●	● (Weak/limited consistency guarantees)
Optimistic & Pessimistic Concurrency (Two-Phase-Commit)	●	●	●	● (Optimistic Only)
One-Phase-Commit Optimization	●	●	●	●
Near Cache Transactions (i.e., Client Cache Transactions)	●	●	●	●
Cross-Partition Transactions	●	●	●	● (Only keys from the same hash slot are supported)
Transactional Entry Processor	●	●	●	●
Eviction / Expiration Policies for Transactional Caches	●	●	●	●
Merge with DB Transactions (e.g., Oracle DB, MySQL, etc.)	●	●	●	●
Explicit Locking	●	●	●	●

FEATURE	GRIDGAIN CE 8.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.7	GRIDGAIN UE 8.7	REDIS ENTERPRISE 5.54 (REDIS 5.0)
Distributed Architecture				
Key-Value Store	●	●	●	●
Partitioning and Replication	●	●	●	● (Predefined partitions only, asynchronous replication, consistency not guaranteed)
Elasticity (add/remove nodes on demand)	●	●	●	●
Client-side (Near / inline) Cache	●	●	●	●
Dynamic Cache Creation	●	●	●	●
EntryProcessor, aka Delta (Partial) Updates	●	●	●	●
Data Redundancy (Key Backups)	●	●	●	● (Master-slave only as a complete replica)
Synchronous and Asynchronous Backup Update	●	●	●	● (Asynchronous Only)
Synchronous APIs	●	●	●	●
Asynchronous APIs	●	●	●	●
Full Sync Mode (Primary and Backups are Sync)	●	●	●	●
Primary Sync Mode (Primary is sync, Backups are async)	●	●	●	●
Full Async Mode (Primary and Backups are Async)	●	●	●	●
Network Segmentation (Split Brain)	●	●	●	● (Using Redis Sentinel)
Data Conflict Resolution	●	●	●	●
Data Affinity and Collocation	● (Rich Support)	● (Rich Support)	● (Rich Support)	● (Not automatic. Requires manual partitioning)
Custom affinity (partitioning) function	●	●	●	● (Can create customer sharding but complicated)
Data Eviction and Expiration	● (LRU, FIFO, Random, Sorted, Custom)	● (LRU, FIFO, Random, Sorted, Custom)	● (LRU, FIFO, Random, Sorted, Custom)	● (LRU, LFU, Random)
Binary Objects	●	●	●	●
Pluggable Interfaces (SPIs) to Customize Grid Subsystems	●	●	●	●
Dynamic Object Version Change (allowing dynamic change to an object's structure)	●	●	●	●

FEATURE	GRIDGAIN CE 8.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.7	GRIDGAIN UE 8.7	REDIS ENTERPRISE 5.54 (REDIS 5.0)
Distributed Data Structures				
Queue	●	●	●	●
Set	●	●	●	●
Atomic Log	●	●	●	●
Atomic Ref	●	●	●	●
Atomic Stamped Ref	●	●	●	●
Atomic Sequence	●	●	●	●
Count Down Latch	●	●	●	●
Reentrant Lock	●	●	●	●
Semaphore	●	●	●	●
Data Snapshots (Backups)				
Full Data Snapshots	●	●	● (+ Multi-datacenter data and disaster recovery management)	●
Incremental Data Snapshots	●	●	● (+ Multi-datacenter data and disaster recovery management)	●
Data Recovery from Snapshots	●	●	● (+ Multi-datacenter data and disaster recovery management)	●
Snapshots Scheduling	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (Enterprise only)
Tools for Snapshotting	●	●	● (+ Multi-datacenter data and disaster recovery management)	●
Datacenter (WAN) Replication				
Active-Active	●	●	●	● (Redis Cloud – depends on cloud provider)
Active-Passive	●	●	●	● (Redis Cloud – depends on cloud provider)

FEATURE	GRIDGAIN CE 8.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.7	GRIDGAIN UE 8.7	REDIS ENTERPRISE 5.54 (REDIS 5.0)
Data Rebalancing				
Sync Data Rebalancing (aka Sync Repartitioning)	●	●	●	●
Async Data Rebalancing (aka Async Repartitioning)	●	●	●	●
Delayed Data Rebalancing (Delay Data Rebalancing until All Nodes Have Started)	●	●	●	●
Grid Management and Monitoring				
Rolling Production Updates	●	●	●	● (Minor updates only)
Management and Monitoring GUI	●	●	●	●
Command line Management Tool	●	●	●	●
Standards				
JCache (JSR-107)	●	●	●	●
SQL (ANSI-99)	●	●	●	●
ODBC	●	●	●	●
JDBC	●	●	●	●
XA/JTA	●	●	●	●
OSGI	●	●	●	●

FEATURE	GRIDGAIN CE 8.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.7	GRIDGAIN UE 8.7	REDIS ENTERPRISE 5.54 (REDIS 5.0)
Out-of-the-Box Integration				
Automatic RDBMS integration	●	●	●	●
Spring Framework	●	●	●	●
Apache® Maven™	●	●	●	●
Web Session Clustering	●	●	●	●
Hibernate L2 Cache	●	●	●	●
MyBatis L2 Cache	●	●	●	●
Vert.x	●	●	●	●
JMS	●	●	●	●
Apache® Flume™	●	●	●	●
MQTT	●	●	●	●
Twitter	●	●	●	●
Apache® Kafka™	●	●	●	●
Apache® Camel™	●	●	●	●
Apache® Storm™	●	●	●	●
Spring Caching	●	●	●	●
Oracle® Golden Gate	●	●	●	●
Cloud and Virtualization Support				
TCP/IP Cluster Protocol	●	●	●	●
Pluggable Discovery	●	●	●	●
Amazon® Web Services	● (S3-Based IP Finder)	● (S3-Based IP Finder)	● (S3-Based IP Finder)	●
Google® Compute	●	●	●	●
Microsoft Azure	●	●	●	●
Apache® JClouds™	●	●	●	●
Docker Container	●	●	●	●
Kubernetes	●	●	●	●
In-Memory Streaming				
Data Streamers	●	●	●	● (Redis Streams)
Complex Event Processing (CEP)	●	●	●	● (With pub/sub paradigm or Redis Lists)

FEATURE	GRIDGAIN CE 8.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.7	GRIDGAIN UE 8.7	REDIS ENTERPRISE 5.54 (REDIS 5.0)
Distributed Messaging and Events				
Topic-based Publish/Subscribe Messaging	● (Ordered, Unordered)	● (Ordered, Unordered)	● (Ordered, Unordered)	●
Point-to-Point Messaging	●	●	●	●
Grid Event Notifications	●	●	●	● (Only for Events Affecting the Data Set)
Automatic Batching of Event Notifications	●	●	●	●
Distributed Computing				
Affinity-Aware Execution	●	●	●	●
Executor Service	●	●	●	●
Managed Services	●	●	●	●
Sub-Grid Messaging / Task Execution	●	●	●	● (Messaging)
Zero Deployment Technology	●	●	●	●
Direct API for MapReduce and ForkJoin	●	●	●	●
Early and Late Load Balancing	●	●	●	●
Fault-Tolerance	●	●	●	●
Computation State Checkpoints	●	●	●	●
Distributed Computation (Task) Sessions	●	●	●	●
Cron-like Task Scheduling	●	●	●	● (Via Quartz, not integrated)
Security and Audit				
SSL Support	●	●	●	● (Spiped Support)
Client Authentication	●	●	●	●
Cluster Member Authentication	●	●	●	●
ACL-Based Passcode Authentication	●	●	●	●
JAAS Authentication	●	●	●	●
Authorization and Permit	●	●	●	● (Redis Cloud – Security Groups of the Cloud Provider)
Audit (Trace Events)	●	●	●	●
Multi-Tenancy	●	●	●	●

FEATURE	GRIDGAIN CE 8.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.7	GRIDGAIN UE 8.7	REDIS ENTERPRISE 5.54 (REDIS 5.0)
Data Visualizations				
Hosted Web Console	●	●	●	● (Redis Cloud)
On-Premises Web Console	●	●	●	●
Apache® Zeppelin™	●	●	●	●
Tableau®	●	●	●	●
Client-Server Protocol				
Memcached Support	●	●	●	●
HTTP REST	●	●	●	●
Supported Platforms				
Java and JVM-based Platforms	●	●	●	●
C++ Client	●	●	●	●
.NET/C# Client	●	●	●	●
Scala DSL	●	●	●	●
Node.JS Client	●	●	●	●
Interoperability between .NET/Java/C++	●	●	●	●
Integration with Spark				
Implementation of Spark RDD and DataFrame	●	●	●	● (RDD only)
Native SQL optimization	●	●	●	●
Deployment				
Apache® Mesos™	●	●	●	●
Hadoop® Yarn	●	●	●	●
Apache® BigTop™	●	●	●	●

Additional Product Comparisons

You can also learn how GridGain compares to other in-memory solutions, including GigaSpaces®, Hazelcast®, Terracotta®, Coherence® and Pivotal GemFire® by visiting www.gridgain.com.

Contact GridGain Systems

To learn more about the GridGain in-memory computing platform, please email our sales team at sales@gridgain.com or call us at +1 (650) 241-2281 (US) or +44 (0) 208 610 0666 (Europe).

About GridGain Systems

GridGain Systems is revolutionizing real-time data access and processing with the GridGain in-memory computing platform built on Apache® Ignite™. GridGain and Apache Ignite are used by tens of thousands of global enterprises in financial services, fintech, software, e-commerce, retail, online business services, healthcare, telecom and other major sectors, with a client list that includes ING, Raymond James, American Express, Societe Generale, Finastra, IHS Markit, ServiceNow, Marketo, RingCentral, American Airlines, Agilent, and UnitedHealthcare. GridGain delivers unprecedented speed and massive scalability to both legacy and greenfield applications. Deployed on a distributed cluster of commodity servers, GridGain software can reside between the application and data layers (RDBMS, NoSQL and Apache® Hadoop®), requiring no rip-and-replace of the existing databases, or it can be deployed as an in-memory transactional SQL database. GridGain is the most comprehensive in-memory computing platform for high-volume ACID transactions, real-time analytics, web-scale applications, continuous learning and hybrid transactional/analytical processing (HTAP). For more information on GridGain products and services, visit www.gridgain.com.