GridGain

Built On Apache Ignite

FEATURE COMPARISON

GridGain In-Memory Computing Platform Feature Comparison: Software AG Terracotta

This document presents a summary and detailed feature comparison of the GridGain® in-memory computing platform (GridGain) and Software AG Terracotta® BigMemory™ (Terracotta) for use as in-memory data grids (IMDG) and in-memory computing use cases.

Major Advantages of GridGain vs. Terracotta

- Comprehensive in-memory computing platform
- Native ANSI-99 SQL support
- Distributed pessimistic, optimistic and deadlock-free ACID transaction support
- Automatic integration with RDBMSs, NoSQL databases and Hadoop
- Sits in-between SQL-based applications and RDBMSs, eliminating the need to replace SQL with code
- Cross-language support for Massively Parallel Processing (MPP) for Java, .NET and C++
- Distributed in-memory database (IMDB) with better transaction support, scalability, availability and reliability
- Cross-language support for Massively Parallel Processing (MPP) for Java, .NET and C++
- Apache® Spark™ support for DataFrames, RDDs, HDFS and SparkSQL acceleration
- Built-in Continuous Learning Framework with support for machine and deep learning
- Built on Apache Ignite, a top 5 Apache Software Foundation project

SOFTWARE AG TERRACOTTA BIGMEMORY DATA MANAGEMENT PLATFORM AND GRIDGAIN IN-MEMORY COMPUTING PLATFORM COMPARED

The Software AG Terracotta BigMemory Max data management platform is used by companies to deliver low, predictable latency at any scale for applications. There is also a single server version, BigMemory Go, which is not considered here in part because it has a subset of the functionality of BigMemory Max. BigMemory includes Terracotta EHCache™, which is available both as a commercial version and as open source under the Apache 2.0 license.

GridGain, built on the Apache Ignite™ open source project, is an in-memory computing platform that includes a distributed in-memory data grid (IMDG), an in-memory SSQL and key-value in-memory database (IMDB), and a stream processing and analytics engine. GridGain Systems donated the original code to the Apache Ignite project and is the largest contributor.

Terracotta has many of the core capabilities expected in an IMDG, including the ability to distribute and partition data, and scale out across a cluster. But Terracotta hasn't evolved much in the years since Software AG acquired Terracotta. Five years after the release of BigMemory 4.0, the current release is 4.3.6. Even EHCache, which exists under an Apache 2.0 license, has not evolved much in comparison to other caching and IMDG vendors.

GridGain is better than Terracotta in almost every IMDG use case. This is due, in part to the innovation that comes from being built on Apache Ignite. GridGain Systems donated the original Apache Ignite code to the Apache Software Foundation (ASF) in 2014 and remains the most active contributor. Ignite became a top level ASF project in 2015. It is now one of the top five Apache Software Foundation open source projects in commits and list activity.

In the last three years Apache Ignite and GridGain have added major new capabilities, including:

- Distributed SQL and ACID transaction support
- All the capabilities of a SQL and key-value IMDB
- The broadest Spark support
- A Continuous Learning Framework for machine and deep learning

One key area of innovation is GridGain's support for ANSI-99 SQL and ACID transactions. These capabilities support the use of GridGain as an IMDG on top of existing databases, and as a distributed in-memory database. GridGain is the only IMDG with support for SQL. As a result, it is the only IMDG that allows applications, analytics and reporting tools that rely on SQL to easily access data. GridGain is also the only IMDG that can sit in-between existing SQL-based applications and RDBMSs and use ANSI-99 SQL instead of code for the integration. Terracotta only supports limited SQL for queries. It has no way to provide easy access to data for all the tools and applications that rely on SQL. It also requires extensive custom coding in applications to replace existing SQL when used as an IMDG.

GridGain also has proven high-performance support for distributed pessimistic, optimistic and deadlock-free ACID transactions. While Terracotta has some pessimistic transaction support, it does not support optimistic or deadlock-free transactions, and requires the use of XA to coordinate distributed pessimistic transactions.

GridGain is also better for supporting IMDG and many other non-IMDG in-memory computing use cases on a single common deployment. It is a broader in-memory computing platform that includes an IMDG, in-memory database (IMDB), streaming analytics with support for Apache Spark, machine learning and deep learning. Terracotta is just an IMDG. Software AG sells a separate product, Terracotta DB, which is a NoSQL (key-value) disk database and EHCache instance for in-memory caching that are bundled together but not integrated. Neither BigMemory Max nor Terracotta DB support SQL. Software AG sells yet another product, Apama™, for streaming analytics and complex event processing (CEP). These products are not integrated together, and there is no support for Spark or machine and deep learning.

GridGain also provides better transactional support, performance, scalability, availability and reliability across IMDG and IMDB use cases as an integrated platform. It supports fully distributed optimistic and deadlock-free, not just pessimistic transactions. GridGain's memory-centric architecture allows any data and indexes to be stored in heap memory, off-heap memory and disk. Nodes can be dynamically added or removed, and data automatically rebalanced without downtime for increased uptime. And centralized backup and recovery ensures data is never lost. Neither BigMemoryMax nor Terracotta DB support these capabilities. In addition, Terracotta's requirement that all data stored off-heap and disk must be serializable significantly impacts performance.

To improve scalability across all use cases, GridGain supports massively parallel processing (MPP), or collocating processing – distributing Java, .NET and C++ code – with the data. This eliminates the network bottleneck of moving data to the computing. Terracotta does not support any form of MPP.

TERRACOTTA AND GRIDGAIN EDITIONS

To compare Terracotta to Apache Ignite and GridGain, it is important to understand the differences between Terracotta and the GridGain editions. This document will compare Terracotta to the various GridGain editions. This document does not compare Apache Ignite to EHCache, since EHCache only provides a portion of Terracotta BigMemory Max's IMDG capabilities. It also does not compare to BigMemory Go, which is a single-node version of BigMemory Max.

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 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 rec(PEovery features for using GridGain as an in-memory database. For this comparison, all the features of Terracotta BigMemory Max are used unless otherwise noted.

FEATURE	GRIDGAIN CE 8.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.7	GRIDGAIN UE 8.7	TERRACOTTA 4.3.6
Native ANSI-99 SQL Support	●	●	●	●
Distributed ACID Transaction Support	●	●	●	● (No optimistic or deadlock-free)
Slide in-between SQL-based Applications and RDBMSs with no Custom Coding	●	●	●	● (No SQL support. Requires code to replace SQL in existing apps)
Cross-Language Support for Collected 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 collocated processing or affinity-based partitions)
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. Doesn't support Cassandra, MongoDB, Spark, Hadoop)
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)	● (Supports IMDG only)
Apache Spark Support for DataFrames, RDDs, HDFS	●	●	●	●
Built-in Machine 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)	● (Built on EHCache, caching only, under an Apache 2.0 License)

This document provides a detailed feature comparison between the various levels of GridGain and Terracotta, highlighting differences between the open source and enterprise-ready products where relevant.

COMPREHENSIVE IN-MEMORY COMPUTING PLATFORM

GridGain can support multiple third-party databases as an IMDG, act as an IMDB, support streaming analytics, and perform machine and deep learning for different applications, all on the same deployment. Terracotta's strengths are as an IMDG. Terracotta BigMemory Max is not intended for these other use cases. Terracotta DB is needed for IMDB use cases. Streaming requires the use of Apama. There is no support for machine and deep learning. Because of this you can end up with multiple technologies to support different projects. This ends up fragmenting data and adding developer and administrative costs.

NATIVE ANSI-99 SQL SUPPORT

GridGain supports ANSI-99 compliant SQL, including distributed SQL JOINs for querying, updating, indexing and defining data. GridGain works with the SQL from an application by providing ODBC and JDBC drivers for Java, .NET, C++, Python and other languages. Terracotta has limited query support for SQL through an API, with no support for JDBC or ODBC drivers. In order to integrate an application with Terracotta, developers have to change existing applications. They have to add code that replaces access to each table and each SQL query or update to the underlying database. A lack of SQL support also means various SQL-based analytics and reporting tools cannot access the data in Terracotta.

DISTRIBUTED PESSIMISTIC, OPTIMISTIC AND DEADLOCK-FREE 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 both as an IMDG and IMDB. Terracotta's support is more limited. For example, Terracotta only supports pessimistic, not optimistic or deadlock-free transactions and does not provide deadlock detection facilities like those in GridGain.

If you want to performance distributed pessimistic transactions in Terracotta you need to use XA and rely on an external transaction coordinator. With GridGain distributed transaction support is built-in.

AUTOMATIC INTEGRATION WITH RDBMSS, NOSQL DATABASES AND HADOOP

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. Terracotta = requires writing custom code to integrate with third-party RDBMSs and does not support Cassandra, MongoDB, Spark or Hadoop.

SITS IN-BETWEEN SQL-BASED APPLICATIONS AND RDBMSS WITH FULL SUPPORT FOR SQL

GridGain's architecture and out-of-the-box integration enables many use cases to be implemented by working with SQL with no custom coding. GridGain can sit in-between and accelerate SQL-based applications and third-part databases by allowing the SQL applications to use SQL to query and insert/update data in GridGain. The applications use GridGain's native JDBC/ODBC drivers in place of existing drivers. GridGain supports any ANSI-99 SQL requests from the application through the drivers, which means developers can focus on SQL and not have to write custom code that replaces SQL. Terracotta may require significant code changes to an application because it does not support SQL. It requires developers to add code that replaces SQL calls to the underlying database with calls to Terracotta, and more code that enables Terracotta to query and write to the underlying database.

CROSS-LANGUAGE SUPPORT FOR COLLOCATED PROCESSING (MPP)

Both GridGain and Terracotta support multiple languages for developing clients. But Terracotta has no support for collocated computing. 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 light-weight 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.

DISTRIBUTED IN-MEMORY DATABASE (IMDB) WITH BUILT-IN PERSISTENCE

GridGain includes a distributed SQL and key-value hybrid in-memory database that combines memory-centric storage with built-in persistence. As an IMDB, GridGain provides better transaction support, scalability, availability and reliability. GridGain's memory-centric architecture with built-in 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 while RAM holds 0-100% of the data and indexes. GridGain scales better by allowing and data and indexes to be stored in heap, offheap or disk. If a subset of data or an index is not in RAM, it will be used from non-volatile storage. Data in RAM and non-volatile 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. Terracotta requires any data stored off-heap or on disk to be serializable and can only store indexes in heap. This means either yo have to keep all indexes in heap and either keep more data in heap or incur the cost of serialization and deserialization for data access, which is significant impact on performance and scalability. GridGain also has immediate availability on cluster restarts for SQL and key-value operations before any data or indexes are loaded into RAM. 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. Terracotta does not support SQL and, as a key-value store, requires all data to be loaded first before handling requests. Terracotta also requires a restart for any node or data rebalancing, which increased downtime. GridGain rebalancing is automatic and without any downtime. GridGain also includes centralized backup and recovery of all data

to ensure data is never lost. Capabilities include full and incremental snapshots, continuous archiving and point-in-time recovery, down to the transaction. It also includes network backups, and heterogeneous recovery of a cluster where you can redeploy an existing cluster to any other location on premise or in the cloud, with a different number of nodes and amount of memory. Terracotta lacks the same level of backup and recovery. Each backup must be explicitly performed. They are only full backups, not incremental or continuous backups, which means they will be slower and any data written after the last backup can be lost. Terracotta also cannot backup data that overflows to disk. If servers fail, the overflowed data is lost.

APACHE SPARK™ SUPPORT FOR DATAFRAMES, RDDS, HDFS, AND SPARKSQL ACCELERATION

GridGain provides the broadest integration with Spark compared to other in-memory computing vendors. It provides native Apache Spark 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 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, and GridGain's machine and deep learning capabilities. Terracotta provides no support for Spark.

BUILT-IN CONTINUOUS LEARNING FRAMEWORK WITH SUPPORT FOR MACHINE AND DEEP LEARNING

GridGain includes the GridGain® Continuous Learning Framework, which provides built-in machine learning and deep learning with real-time performance on petabytes of data. GridGain provides several out-of-the-box machine learning algorithms optimized for MPP-style collocated 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 create and deploy their own algorithms across any cluster as well by using the compute grid. Terracotta does not support built-in MPP-style machine and deep learning that runs in place. Terracotta provides no support for machine or deep learning.

BUILT ON APACHE IGNITE, A TOP 5 APACHE SOFTWARE FOUNDATION PROJECT

GridGain is built on Apache Ignite, one of the top five Apache Software Foundation projects in terms of commits and community activity. Terracotta is built on the EHCache open source project, which is an Apache 2.0 license. There hasn't been much innovation in EHCache the last few years. In addition, EHCache only provides the caching portion of Terracotta. The rest is proprietary which means there is no real open source option.

GRIDGAIN AND TERRACOTTA DETAILED FEATURE COMPARISON

The following table provides a detailed feature comparison between the GridGain Community, Enterprise, and Ultimate Editions, and Terracotta. 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	TERRACOTTA 4.3.6
Use Cases				
In-Memory Data Grid	●	●	●	●
Third party Database Caching and Persistence (Inline)	●	●	●	●
SQL Database	●	●	● (+ Multi-datacenter data and disaster recovery management)	●
In-Memory Database	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (Terracotta DB offers disk-based persistence and separate EHCache instance)
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	TERRACOTTA 4.3.6
Third Party Database Support, Persistence				
Automatic support for Leading RDBMSs (Oracle, IBM DB2, Microsoft SQL Server, MySQL, Postgres ...)	●	●	●	● (It works well, but requires coding to implement. It's not out of the box)
Inline Support for 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)	● (Only stores data in heap or off-heap RAM to disk for fast restart)
Stores Superset of Data	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (Can overflow data to disk from memory, but it is lost on restart. Only data in memory is backed up)
Store Indexes on Disk	●	●	● (+ Multi-datacenter data and disaster recovery management)	●
SQL or Key-Value over Disk	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (Only serialized data can be stored on disk)
Instantaneous Restart (before memory warm-up)	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (requires all data to be reloaded first)

FEATURE	GRIDGAIN CE 2.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.5	GRIDGAIN UE 8.5	TERRACOTTA 4.3.5
Distributed SQL				
SQL Queries	● (Full ANSI-99 Support)	● (Full ANSI-99 Support)	● (Full ANSI-99 Support)	● (Limited SQL query support)
Collocated Distributed Joins	●	●	●	●
Non-Collocated Distributed Joins	●	●	●	● (Limited SQL query support)
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)	●	●	●	●
DDL (CREATE, DROP, ALTER)	●	●	●	●
Distributed Queries				
Continuous Queries	●	●	●	●
Predicate-based Queries	●	●	●	●
SQL Drivers				
JDBC Driver	●	●	●	●
ODBC Driver	●	●	●	●
REST API (SQL)	●	●	●	●

FEATURE	GRIDGAIN CE 2.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.5	GRIDGAIN UE 8.5	TERRACOTTA 4.3.5
Memory Architecture				
On-Heap Memory	●	●	●	●
Off-Heap Memory	●	●	●	●
Off-Heap Indexes	●	●	●	●
Disk as main storage (disk larger than RAM)	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (Disk only supports serialized data. Disk overflow not saved to disk)
Tiered Storage - On-Heap, Off-Heap and Disk	●	●	● (+ Multi-datacenter data and disaster recovery management)	●
ACID Compliant Transactions and Locks				
Atomic Mode (One Operation at a Time)	●	●	●	●
READ_COMMITTED, REPEATABLE_READ, SERIALIZABLE Isolation Levels	●	●	●	● (READ_COMMITTED ONLY)
Deadlock-Free Transactions	●	●	●	●
XA Integration	●	●	●	●
Fault Tolerance (Including Client/Near/ Primary/Backup Node Failures)	●	●	●	●
Optimistic & Pessimistic Concurrency (Two-Phase-Commit)	●	●	●	● (Pessimistic only)
One-Phase-Commit Optimization	●	●	●	●
Near Cache Transactions (i.e., Client Cache Transactions)	●	●	●	●
Cross-Partition Transactions	●	●	●	● (Only through external XA transaction manager)
Transactional Entry Processor	●	●	●	●
Eviction / Expiration Policies for Transactional Caches	●	●	●	●
Merge with DB Trans-actions (e.g., Oracle DB, MySQL, etc.)	●	●	●	● (only through external Transaction Manager)
Explicit Locking	●	●	●	●

FEATURE	GRIDGAIN CE 2.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.5	GRIDGAIN UE 8.5	TERRACOTTA 4.3.5
Distributed Architecture				
Key-Value Store	●	●	●	●
Partitioning and Replication	●	●	●	● (Replication not recommended a active-active or single WAN cluster, only for DR/backup over WAN)
Elasticity (add/remove nodes on demand)	●	●	●	● (adding/removing requires restart)
Client-side (Near / inline) Cache	●	●	●	● (no transactional capabilities)
Dynamic Cache Creation	●	●	●	●
EntryProcessor, aka Delta (Partial) Updates	●	●	●	●
Data Redundancy (Key Backups)	●	●	●	●
Synchronous and Asynchronous Backup Update	●	●	●	● (Synchronous 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)	●	●	●	●
Data Conflict Resolution	●	●	●	● (Limited. No custom logic)
Data Affinity and Collocation	● (Rich Support)	● (Rich Support)	● (Rich Support)	●
Custom affinity (partitioning) function	●	●	●	●
Data Eviction and Expiration	● (LRU, FIFO, Random, Sorted,Custom)	● (LRU, FIFO, Random, Sorted,Custom)	● (LRU, FIFO, Random, Sorted,Custom)	● (LRU, LFU, FIFO, Custom)
Binary Objects	●	●	●	●
Pluggable Interfaces (SPIs) to Customize Grid Subsystems	●	●	●	●
Dynamic Object Version Change (allowing dynamic change to an object's structure)	●	●	●	●

FEATURE	GRIDGAIN CE 2.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.5	GRIDGAIN UE 8.5	TERRACOTTA 4.3.5
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)	●
Continuous Archiving (WAL)	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (Async or sync backup with each write)
Data Recovery from Snapshots	●	●	● (+ Multi-datacenter data and disaster recovery management)	●
Snapshots Scheduling	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (Backup happens with each write)
Tools for Snapshotting	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (Can explicitly trigger a backup)
Datacenter (WAN) Replication				
Active-Active	●	●	●	●
Active-Passive	●	●	●	●

FEATURE	GRIDGAIN CE 2.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.5	GRIDGAIN UE 8.5	TERRACOTTA 4.3.5
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	●	●	●	●
Management and Monitoring GUI	●	●	●	●
Command line Management Tool	●	●	●	●
Standards				
JCache (JSR-107)	●	●	●	●
SQL (ANSI-99)	●	●	●	●
ODBC	●	●	●	●
JDBC	●	●	●	●
XA/JTA	●	●	●	●
OSGI	●	●	●	●

FEATURE	GRIDGAIN CE 2.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.5	GRIDGAIN UE 8.5	TERRACOTTA 4.3.5
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	●	●	●	●
Complex Event Processing (CEP)	●	●	●	● (Have Apama, separate CEP product and not integrated)

FEATURE	GRIDGAIN CE 2.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.5	GRIDGAIN UE 8.5	TERRACOTTA 4.3.5
Distributed Messaging and Events				
Topic-based Publish/Subscribe Messaging	● (Ordered, Unordered)	● (Ordered, Unordered)	● (Ordered, Unordered)	●
Point-to-Point Messaging	●	●	●	●
Grid Event Notifications	●	●	●	●
Automatic Batching of Event Notifications	●	●	●	●
Distributed Computing				
Affinity-Aware Execution	●	●	●	●
Executor Service	●	●	●	●
Managed Services	●	●	●	●
Sub-Grid Messaging / Task Execution	●	●	●	●
Zero Deployment Tech-nology	●	●	●	●
Direct API for MapReduce and ForkJoin	●	●	●	●
Early and Late Load Balancing	●	●	●	●
Fault-Tolerance	●	●	●	●
Computation State Checkpoints	●	●	●	●
Distributed Computation (Task) Sessions	●	●	●	●
Cron-like Task Scheduling	●	●	●	●
Security and Audit				
SSL Support	●	●	●	● (Limited, done externally)
Client Authentication	●	●	●	●
Cluster Member Authentication	●	●	●	●
ACL-Based Passcode Authentication	●	●	●	●
JAAS Authentication	●	●	●	●
Authorization and Permit	●	●	●	●
Audit (Trace Events)	●	●	●	●
Multi-Tenancy	●	●	●	●

FEATURE	GRIDGAIN CE 2.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.5	GRIDGAIN UE 8.5	TERRACOTTA 4.3.5
Data Visualizations				
Hosted Web Console	●	●	●	●
On-Premises Web Console	●	●	●	●
Apache® Zeppelin™	●	●	●	●
Tableau®	●	●	●	●
Client-Server Protocol				
Memcached Support	●	●	●	●
HTTP REST	●	●	●	●
Supported Platforms				
Java & JVM-based Platforms	●	●	●	● (Java thick client only)
C++ Client	●	●	●	●
.NET/C# Client	●	●	●	●
Scala DSL	●	●	●	●
Node.JS Client	●	●	●	●
Interoperability between .NET/Java/C++	●	●	●	● (Limited support)
Integration with Spark				
Implementation of Spark RDD and DataFrame	●	●	●	●
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 Redis®, Hazelcast®, Pivotal GemFire®, Oracle Coherence® and GigaSpaces® by visiting the 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, call us at +1 (650) 241-2281 (US) or +44 (0) 208 610 0666 (Europe), or fill out our contact page at www.gridgain.com/contact and we will contact you.

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.