GridGain

Built On Apache Ignite

FEATURE COMPARISON

GRIDGAIN IN-MEMORY COMPUTING PLATFORM FEATURE COMPARISON: Hazelcast® IMDG

This document presents a summary and detailed feature comparison of the GridGain® in-memory computing platform (GridGain) and Hazelcast® IMDG (Hazelcast) for use as in-memory data grids.

Major Advantages of GridGain vs Hazelcast IMDG

- ANSI-99 SQL Support
- Distributed ACID Transaction Support
- Slides In-between SQL-based Applications and RDBMSs with No Custom Coding
- Cross-Language Support for Collocated Processing (Java, .NET and C++)
- Native Integration with RDBMSs, NoSQL Databases and Hadoop
- Comprehensive In-Memory Computing Solution
- Support for Apache® Spark™ DataFrames, RDDs and HDFS
- Built-in Machine Learning and Deep Learning

HAZELCAST IMDG AND THE GRIDGAIN IN-MEMORY COMPUTING PLATFORM COMPARED

Hazelcast IMDG is an in-memory data grid that is used by companies to improve application speed and scale.

Hazelcast, the company, offers both a commercially supported version and an open source version of Hazelcast

IMDG under the Apache® License 2. However, Hazelcast IMDG is not an Apache Software Foundation project.

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 SQL and key-value database, and a stream processing and analytics engine. GridGain Systems donated the original code to the Apache Ignite project and is the largest contributor.

Hazelcast and GridGain both support in-memory data grid use cases and provide strong performance and scalability. But Hazelcast is only a viable contender as an in-memory data grid for new Java-based applications. In almost every other IMDG and in-memory computing use case GridGain is better. Because GridGain is a platform, it is also a better choice for common in-memory computing infrastructure. GridGain supports SQL-based applications with ANSI-99 SQL support and can slide in-between existing SQL-based applications and RDBMSs with no custom coding. Hazelcast requires extensive custom coding. GridGain has better support for ACID transactions. It has stronger out-of-the-box support for RDBMSs, NoSQL databases, Apache Spark™ and HDFS. It has stronger support for .NET and C++, including collocated processing (or massively parallel processing) support. GridGain also has built-in machine learning and deep learning. Hazelcast has no built-in machine learning or deep learning.

HAZELCAST AND GRIDGAIN EDITIONS

To compare Hazelcast and Hazelcast Enterprise to Apache Ignite and GridGain, it is 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 GridGain Enterprise Edition (EE) adds enterprise-grade security, deployment, and management capabilities needed for most mission critical inmemory 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.

FEATURE	GRIDGAIN CE 2.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.5	GRIDGAIN UE 8.5	HAZELCAST ENTERPRISE 3.12
Native ANSI-99 SQL Support	●	●	●	●
Distributed ACID Transaction Support	●	●	●	● (Doesn't do distributed cross-partition trans-actions)
Slide in-between SQL-based Applications and RDBMSs with no Custom Coding	●	●	●	● (Requires code for new apps, adding code and data model for 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++)	● (Supports Multiple Languages, only Java 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	● (IMDG, streaming, machine and deep learning)	● (IMDG, streaming, machine and deep learning)	● (EE + Multi-datacenter data and disaster recovery management)	● (IMDG only. No IMDB, streaming, machine or deep learning)
Apache Spark Support for DataFrames, RDDs, HDFS	●	●	●	● (RDD support only)
Built-in Machine Learning	●	●	●	●

This document provides a very detailed feature comparison between the various levels of GridGain and Hazelcast, 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:

NATIVE ANSI-99 SQL SUPPORT

GridGain supports ANSI-99 compliant SQL, including distributed SQL JOINs for querying in-memory data. GridGain works with the SQL from an existing application by providing ODBC and JDBC drivers applications use in place of their existing drivers. Hazelcast requires developers to change existing applications, and to add code for each table and SQL query to the underlying database.

ACID TRANSACTIONS

GridGain has full support for ACID transactions, including OPTIMISTIC and PESSIMISTIC concurrency modes, as well as READ_COMMITTED, REPEATABLE_READ, and SERIALIZABLE isolation levels. While Hazelcast does provide some ACID transaction support, it does not support crosspartition transactions, and requires explicitly coding transactions using an XA coordinator for transactions with databases, as well as blocking or queuing other writes to ensure a successful transaction across Hazelcast and the database. Hazelcast also has a significant challenge mixing transactional and nontransactional writes. In order to ensure transactions commit to the backend database and Hazelcast properly, developers have to explicitly code the combined transaction and Hazelcast update separately from the existing MapStore as well as queue or block non-transactional writes to avoid potential multiple writes.

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 Cassandra® or MongoDB®, as well as Hadoop via Spark and HDFS.

SLIDES IN-BETWEEN SQL-BASED APPLICATIONS AND RDBMSS WITH NO CUSTOM CODING

GridGain's architecture and out-of-the-box integration enables many use cases to be implemented through configuration, not coding. For example, GridGain can slide in between and accelerate SQL-based applications and third-party databases just by replacing existing application ODBC or JDBC drivers with GridGain's native drivers, and by configuring GridGain to connect to the backend database. Hazelcast on the other hand requires significant code changes to an application because it does not support SQL. For each table accessed, a developer needs to write code that maps the table to a MapStore. Then the developer needs to create methods for the MapStore that wrapper each SQL table operation. Finally, the developer needs to change the application to use the new methods for accessing relational data.

CROSS-LANGUAGE SUPPORT FOR COLLOCATED PROCESSING (MPP)

Both GridGain and Hazelcast have extensive client support for different languages. But Hazelcast has limited programming language support for collocated computing. It can distribute Java code. GridGain provides a general purpose massively parallel processing (MPP) collocated processing compute grid that is used for distributed SQL and machine and deep learning. It also supports user-defined Java, .NET and C++ code, making it a better choice for non-Java-based applications or as shared IMDG infrastructure. GridGain also includes a binary protocol that enables client support without requiring a JVM having to be deployed with a non-Java client.

COMPREHENSIVE IN-MEMORY COMPUTING SOLUTION

GridGain not only supports third-party databases as an IMDG. GridGain also includes a distributed SQL and key-value database with native persistence for building hybrid transactional/analytical processing (HTAP) applications. Native 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 native persistence enabled, nonvolatile storage stores 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 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. GridGain native persistence is immediately available on 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. Hazelcast cannot be used as a database. Its persistence is only used to back up the in-memory data grid for hot restarts.

APACHE SPARK™ SUPPORT FOR DATAFRAMES, RDDS AND HDFS

GridGain enables Apache Spark to use GridGain for in-memory computing by integrating via Spark Apache DataFrame, RDD and HDFS APIs. GridGain simplifies the access, writing and saving of data, and sharing state across Spark jobs. It also accelerates SQL performance up to 1000x compared to standalone Apache Spark and 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. Hazelcast has limited support for Apache Spark through an open source Spark Connector for Hazelcast that provides read/write access to Hazelcast as RDDs.

SUPPORT FOR MACHINE LEARNING AND DEEP LEARNING

GridGain includes the GridGain® Continuous Learning Framework, built-in machine learning and deep learning with near real-time performance on petabytes of data. GridGain provides several machine learning algorithms out of the box optimized for MPP-style collocated processing including linear and multi-linear regression, k-means clustering, decision trees, k-NN classification and regression; as well as a multilayer perceptron and TensorFlow integration for for deep learning. Developers can develop and deploy their own algorithms across any cluster as well using the compute grid. Hazelcast has no integrated machine or deep learning.

GRIDGAIN AND HAZELCAST DETAILED FEATURE COMPARISON

The following table provides a detailed feature comparison between the GridGain Community, Enterprise, and Ultimate Edition, and Hazelcast. 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 2.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.5	GRIDGAIN UE 8.5	HAZELCAST ENTERPRISE 3.12
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)	●
Web Session Clustering	●	●	●	●
Apache Spark Acceleration	●	●	●	●
Hadoop acceleration	●	●	●	●
In-Memory File System (Hadoop Compliant)	●	●	●	●

FEATURE	GRIDGAIN CE 2.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.5	GRIDGAIN UE 8.5	HAZELCAST ENTERPRISE 3.12
Third Party Database Support, Persistence				
Inline 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)	● (Enterprise edition only)
Stores Superset of Data	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (Only stores copy of data that is in memory)
Store Indexes on Disk	●	●	● (+ Multi-datacenter data and disaster recovery management)	●
SQL or Key-Value over Disk	●	●	● (+ Multi-datacenter data and disaster recovery management)	●
Instantaneous Restart (before memory warmup)	●	●	● (+ Multi-datacenter data and disaster recovery management)	●

FEATURE	GRIDGAIN CE 2.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.5	GRIDGAIN UE 8.5	HAZELCAST ENTERPRISE 3.12
Distributed SQL				
SQL Queries	● (Full ANSI-99 Support)	● (Full ANSI-99 Support)	● (Full ANSI-99 Support)	● (Limited support)
Collocated Distributed Joins	●	●	●	●
Non-Collocated Distributed Joins	●	●	●	●
Single Column Indexes	●	●	●	● (Select values are not atomically consistent as a group)
Group Indexes	●	●	●	● (Select values are not atomically consistent as a group)
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	HAZELCAST ENTERPRISE 3.12
Memory Architecture				
On-Heap Memory	●	●	●	●
Off-Heap Memory	●	●	●	● (Enterprise edition only)
Off-Heap Indexes	●	●	●	●
Disk as main storage (disk larger than RAM)	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (For hot restarts only)
Tiered Storage - On-Heap, Off-Heap and Disk	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (Disk is for hot restarts only)
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)	●	●	●	●
One-Phase-Commit Optimization	●	●	●	●
Near Cache Transactions (i.e., Client Cache Transactions)	●	●	●	●
Cross-Partition Transactions	●	●	●	●
Transactional Entry Processor	●	●	●	●
Eviction / Expiration Policies for Transactional Caches	●	●	●	●
Merge with DB Transactions (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	HAZELCAST ENTERPRISE 3.12
Distributed Architecture				
Key-Value Store	●	●	●	●
Partitioning and Replication	●	●	●	●
Elasticity (add/remove nodes on demand)	●	●	●	●
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	●	●	●	● (Asynchronous Only)
Synchronous APIs	●	●	●	●
Asynchronous APIs	●	●	●	●
Full Sync Mode (Primary and Backups are Sync)	●	●	●	●
Primary Sync Mode (Primary is synch, Backups are Async)	●	●	●	●
Full Async Mode (Primary and Backups are Async)	●	●	●	●
Network Segmentation (Split Brain)	●	●	●	●
Data Conflict Resolution	●	●	●	● (non-transactional)
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,Random)
Binary Objects	●	●	●	●
Pluggable Interfaces (SPIs) to Customize Grid Subsystems	●	●	●	●
Dynamic Object Version Change (allowing dynamic change to an object's structure)	●	●	●	● (support for adding datatypes and functions, but SPI changes each release)

FEATURE	GRIDGAIN CE 2.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.5	GRIDGAIN UE 8.5	HAZELCAST ENTERPRISE 3.12
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)	● (Inconsistency is possible by default, has to move the cluster to PASSIVE state)
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)	●
Tools for Snapshotting	●	●	● (+ Multi-datacenter data and disaster recovery management)	● (Enterprise edition only)
Datacenter (WAN) Replication				
Active-Active	●	●	●	● (Enterprise edition only)
Active-Passive	●	●	●	● (Enterprise edition only)

FEATURE	GRIDGAIN CE 2.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.5	GRIDGAIN UE 8.5	HAZELCAST ENTERPRISE 3.12
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	●	●	●	● (Very limited)

FEATURE	GRIDGAIN CE 2.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.5	GRIDGAIN UE 8.5	HAZELCAST ENTERPRISE 3.12
Out-of-the-Box Integration				
Automatic RDBMS integration	●	●	●	●
Spring Framework	●	●	●	●
Apache® Maven™	●	●	●	●
Web Session Clustering	●	●	●	●
Hibernate L2 Cache	●	●	●	●
MyBatis L2 Cache	●	●	●	●
Vert.x	●	●	●	●
JMS	●	●	●	● (Proprietary messaging only)
Apache® Flume™	●	●	●	●
MQTT	●	●	●	●
Twitter	●	●	●	●
Apache® Kafka™	●	●	●	● (Hazelcast Jet only)
Apache® Camel™	●	●	●	● (Hazelcast component for Camel to connect)
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	●	●	●	● (Hazelcast Jet)
Complex Event Processing (CEP)	●	●	●	● (Hazelcast Jet)

FEATURE	GRIDGAIN CE 2.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.5	GRIDGAIN UE 8.5	HAZELCAST ENTERPRISE 3.12
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	●	●	●	● (Very complicated APIs)
Sub-Grid Messaging / Task Execution	●	●	●	●
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	●	●	●	●
Security and Audit				
SSL Support	●	●	●	● (Enterprise edition only)
Client Authentication	●	●	●	● (Enterprise edition only)
Cluster Member Authentication	●	●	●	● (Enterprise edition only)
ACL-Based Passcode Authentication	●	●	●	●
JAAS Authentication	●	●	●	● (Enterprise edition only)
Authorization and Permit	●	●	●	● (Enterprise edition only)
Audit (Trace Events)	●	●	●	●
Multi-Tenancy	●	●	●	●

FEATURE	GRIDGAIN CE 2.7 (APACHE IGNITE 2.7)	GRIDGAIN EE 8.5	GRIDGAIN UE 8.5	HAZELCAST ENTERPRISE 3.12
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	●	●	●	●
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 Redis®, Oracle® Coherence, Terracotta®, GigaSpaces® and Pivotal GemFire® by visiting the GridGain website.

Contact GridGain Systems

To learn more about how GridGain can help your business, 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 complete our contact form to have us 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 gridgain.com.