GridGain

GRIDGAIN IN-MEMORY DATA FABRIC

Feature Comparison: GigaSpaces®

July, 2016

This document presents a feature comparison of the GridGain In-Memory Data Fabric (GridGain) and GigaSpaces® for in-memory computing applications.

About the GridGain In-Memory Data Fabric

GridGain, built on Apache® Ignite™, is an in-memory computing platform that enables you to dramatically accelerate and scale out your existing data-intensive applications without ripping and replacing your existing databases. GridGain can reduce query times by 1,000x versus disk-based systems. You can scale out by adding new nodes to your GridGain cluster, which can handle hundreds of terabytes of data from multiple databases.

You can modernize your existing data-intensive architecture by inserting GridGain between your existing application and database layers. GridGain integrates seamlessly with RDBMS, NoSQL and Apache® Hadoop™ databases. It features a unified API which supports SQL, C++, .NET, PHP, MapReduce, JAVA/Scala/Groovy, and Node.js protocols for the application layer. Your GridGain cluster, applications, and databases can run on premise, in a hybrid environment, or on a cloud platform such as AWS® or Microsoft Azure.

GridGain is available as a Professional Edition and an Enterprise Edition. The GridGain Professional Edition includes the current version of Apache Ignite plus bug fixes that are not yet released in Ignite. The GridGain Enterprise Edition includes enterprise-grade security, deployment, and management features which have been added to the core features of Apache Ignite. A subscription to the GridGain Professional Edition is part of a GridGain Standard Support package and a subscription to the GridGain Enterprise Edition is included in GridGain Enterprise Support packages.

Key Differences Between GridGain and GigaSpaces

The GridGain In-Memory Data Fabric includes an in-memory data grid feature. GigaSpaces is an in-memory data grid. The data grid capabilities of both products include functionality which partitions and caches data in memory. Both of the data grid solutions can be scaled out across distributed clusters. However, there are many differences in the way caching, transactions, and data querying are supported. The GridGain in-memory computing platform also includes many additional features not included in GigaSpaces that are often highly valuable for companies that are moving to in-memory computing.

The following are major differences between the two products which should be considered when choosing an in-memory solution:

- **JCache (JSR 107)** – GridGain and Apache Ignite data grid are implementations of the JCache (JSR 107) specification that provides a simple to use, yet very powerful API for data access. GigaSpaces provides JCache implementation that is based on the JavaSpaces API.

- **Memory Formats** – GridGain supports storing data in on-heap or off-heap memory, depending on the configuration. GridGain supports storing query indexes off-heap as well. GigaSpaces supports off-heap memory as a separate add-on (MemoryXtend) to its commercial offering, but does provide any support for off-heap indexes.

- **SQL Queries** - GridGain supports complete SQL (ANSI-99) syntax, including distributed SQL JOINs (collocated and non-collocated) for querying in-memory data. GigaSpaces does not support SQL and users have to perform JOINS manually by combining multiple query results.

- **Deadlock-Free Transactions** – GridGain supports deadlock-free, optimistic transactions, which do not acquire any locks, and free users from worrying about the lock order. Such transactions also provide much better performance. With GigaSpaces, you always need to worry about updating data in the same order to avoid deadlocks, which is often impossible, especially in large projects.

- **Cross-Partition Transactions** – GridGain Transactions can be performed on all partitions of a cache across the whole cluster. GigaSpaces does not support transactions across multiple cache partitions.

- **Data Streaming** – GridGain provides support for in-memory streaming, including support for maintaining and querying sliding windows of streaming data. GigaSpaces does not offer any support for streaming.

GridGain and GigaSpaces Detailed Feature Comparison

The following table provides a detailed feature comparison between the GridGain Professional Edition, GridGain Enterprise Edition, and GigaSpaces. 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 PE 1.7	GridGain EE 7.6	GigaSpaces 9.7
Distributed Caching			
Key-Value Store	●	●	●
Partitioning and Replication	●	●	●
Client-side (Near) Cache	●	●	●
Dynamic Cache Creation	●	●	●
EntryProcessor, aka Delta (Partial) Updates	●	●	●
Data Redundancy (Key Backups)	●	●	●
Synchronous and Asynchronous Backup Update	●	●	● (Synchronous Only)
Synchronous APIs	●	●	●
Asynchronous APIs	●	●	●
Full Async Mode (Primary and Backups are Async)	●	●	●
Data Affinity and Collocation	● (Rich Support)	● (Rich Support)	●

Feature	GridGain PE 1.7	GridGain EE 7.6	GigaSpaces 9.7
Data Eviction and Expiration	● (LRU, FIFO, Random, Sorted, Custom)	● (LRU, FIFO, Random, Sorted, Custom)	● (LRU, Custom)
Binary Objects	●	●	●
Pluggable Interfaces (SPIs) to Customize Grid Subsystems	●	●	●
Memory Formats			
On-Heap Memory	●	●	●
Off-Heap Memory	●	●	● (Commercial Offering Only)
Off-Heap Indexes for Off-Heap data	●	●	●
Disk Overflow	●	●	●
Tiered Storage - On-Heap to Off-Heap to Disk	●	●	●
ACID Compliant Transactions and Locks			
Atomic Mode (One Operation at a Time)	●	●	●
READ_COMMITED, REPEATABLE_READ, SERIALIZABLE Isolation Levels	●	●	●
Deadlock-Free Transactions	●	●	●
XA Integration	●	●	●
Fault Tolerance (Including Client/Near/Primary/Backup Node Failures)	●	●	●
Optimistic & Pessimistic Concurrency (Two-Phase-Commit)	●	●	●
One-Phase-Commit Optimization	●	●	●
Custom Affinity (Partitioning) Function	●	●	●
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	●	●	●
Distributed Data Structures			
Queue	●	●	●
Set	●	●	●

Feature	GridGain PE 1.7	GridGain EE 7.6	GigaSpaces 9.7
Atomic Long	●	●	●
Atomic Ref	●	●	●
Atomic Stamped Ref	●	●	●
Atomic Sequence	●	●	●
Count Down Latch	●	●	●
ReentrantLock	●	●	●
Semaphore	●	●	●
Distributed Queries (Searches)			
SQL Queries	● (Full ANSI-99 Support)	● (Full ANSI-99 Support)	● (Limited Support)
Continuous Queries	●	●	●
Predicate-based Queries	●	●	●
Single Column Indexes	●	●	●
Group Indexes	●	●	●
Affinity Collocation Based SQL Joins	●	●	●
Non-Collocated Distributed SQL Joins	●	● (Will Be Available in 7.7)	●
Query Consistency	●	●	●
Query Fault-Tolerance	●	●	●
Custom Query API	●	●	●
SQL Drivers			
JDBC Driver	●	●	●
ODBC Driver	●	●	● (Via JDBC-ODBC Bridge, Not Integrated)
Persistence and Data Loading			
Write-Through and Read-Through	●	●	●
Write-Behind Caching	●	●	●
Auto-Loading of SQL Schema/Data	●	●	●
Local Recoverable Store	●	●	●
Data Streamer (Optimized Bulk Put or Load Operations)	●	●	●
Store Loader (Optimized Bulk DB Load)	●	●	●
Data Rebalancing			
Sync Data Rebalancing (aka Sync Repartitioning)	●	●	●

Feature	GridGain PE 1.7	GridGain EE 7.6	GigaSpaces 9.7
Async Data Rebalancing (aka Async Repartitioning)	●	●	●
Delayed Data Rebalancing (Delay Data Rebalancing until All Nodes Have Started)	●	●	●
Standards			
JCache (JSR-107)	●	●	●
SQL (ANSI-99)	●	●	●
ODBC	●	●	● (Via JDBC-ODBC Bridge, Not Integrated)
JDBC	●	●	●
XA/JTA	●	●	●
OSGI	●	●	●
Integrations			
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)	●

Feature	GridGain PE 1.7	GridGain EE 7.6	GigaSpaces 9.7
Google® Compute	● (Implemented Using Google Compute Engine Storage Based IP Finder)	● (Implemented Using Google Compute Engine Storage Based IP Finder)	●
Apache® JClouds™	●	●	●
Docker Container	●	●	●
Apache® Mesos™	●	●	●
Hadoop® Yarn	●	●	●
In-Memory Streaming			
Data Streamers	●	●	●
Complex Event Processing (CEP)	●	●	●
Sliding Data Windows	●	●	●
Continuous Queries over Data Windows	●	●	●
Distributed Messaging and Events			
Topic-based Publish/Subscribe Messaging	●	●	●
Point-to-Point Messaging	●	●	●
Grid Event Notifications	●	●	●
Automatic Batching of Event Notifications	●	●	●
Distributed Compute			
Affinity-Aware Execution	●	●	●
Executor Service	●	●	●
Managed Services	●	●	●
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	●	●	●
Configuration & Grid Management			
Spring XML Configuration	●	●	●

Feature	GridGain PE 1.7	GridGain EE 7.6	GigaSpaces 9.7
Programmatic Configuration	●	●	●
Elasticity (Ability to Add/Remove Grid Nodes on Demand)	●	●	●
Dynamic Schema Changes (Allowing Dynamic Change to an Object's Structure)	● (Portable Objects)	● (Portable Objects)	●
Datacenter (WAN) Replication (Active-Active, Active-Passive)	●	●	●
Rolling Production Updates	●	●	●
Network Segmentation (Split Brain)	●	● (Transactional)	●
Data Conflicts Resolution	●	●	●
Management and Monitoring GUI	●	●	●
Command-Line Management Tool	●	●	●
Security and Audit			
SSL Support	●	●	●
Client Authentication	●	●	●
Cluster Member Authentication	●	●	●
ACL-Based Passcode Authentication	●	●	●
JAAS Authentication	●	●	●
Authorization and Permit	●	●	●
Audit (Trace Events)	●	●	●
Multi-Tenancy	●	●	●
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++	●	●	●
.NET/C#	●	●	●

Feature	GridGain PE 1.7	GridGain EE 7.6	GigaSpaces 9.7
Scala DSL	●	●	●
Interoperability between .NET/Java/C++	●	●	●
Integration with Spark			
Implementation of Spark RDD	●	●	●
SQL Queries	●	●	●
Hadoop Deployment			
Apache® Mesos™	●	●	●
Hadoop® Yarn	●	●	●
Apache® BigTop™	●	●	●
Hadoop Acceleration			
Hadoop Accelerator	●	●	●
In-Memory File System (Hadoop Compliant)	●	●	●

Additional Product Comparisons

You can also learn how GridGain compares to other in-memory solutions, including Oracle® Coherence, Hazelcast®, Redis®, Pivotal GemFire® and Terracotta® by visiting http://www.gridgain.com/resources/feature-comparisons/.

Contact GridGain

To learn more about the GridGain In-Memory Data Fabric, please email our sales team at sales@gridgain.com or call us at +1 (650) 241-2281 (US) or +44 (0)7775 835 770 (Europe).

ABOUT GRIDGAIN

GridGain is revolutionizing real-time data access and processing by offering the enterprise-grade GridGain In-Memory Data Fabric built on Apache Ignite™. The solution is used by global enterprises in financial, tech, retail, healthcare and other major sectors. GridGain solutions connect traditional and emerging data stores (SQL, NoSQL, and Hadoop) with cloud-scale applications and enable massive data throughput and ultra-low latencies across a scalable cluster of commodity servers. A converged data platform, the GridGain In-Memory Data Fabric offers the most comprehensive, enterprise-grade in-memory computing solution for high-volume transactions, real-time analytics and hybrid data processing. The company is funded by Almaz Capital, MoneyTime Ventures and RTP Ventures.