WHITE PAPER

GridGain

Introducing the GridGain In-Memory Computing Platform

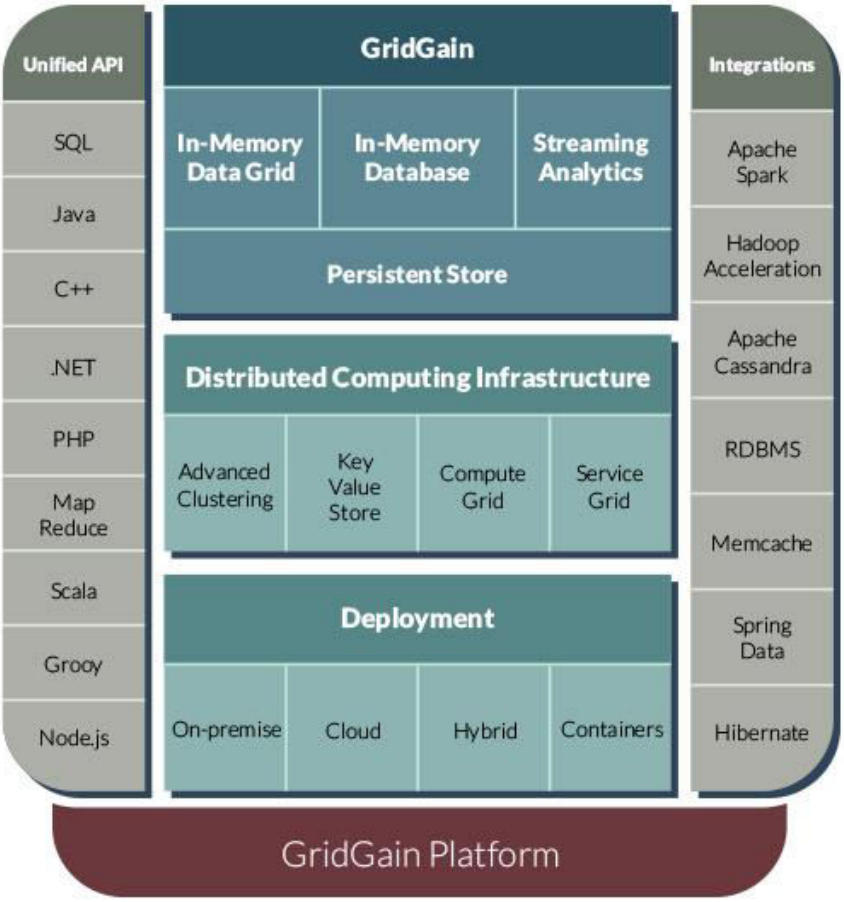
A GridGain Systems In-Memory Computing White Paper

September 2017

Contents

Overview

In-memory computing is characterized by using high-performance, integrated and distributed memory systems to compute and transact on large data sets in real-time. It performs orders of magnitude faster than possible with traditional disk-based or flash technologies.



GridGain is an in-memory computing platform that delivers unprecedented speed and massive scalability to modern data processing. Built on the Apache® Ignite™ open source project, GridGain enables high-performance transactions, real-time streaming, and fast analytics in a single, comprehensive data access and processing layer. GridGain easily powers both existing and new applications in a distributed, massively parallel architecture on affordable, industry-standard hardware. GridGain can run on premise, in a hybrid environment, or on a cloud platform such as AWS, Microsoft Azure or Google Cloud Platform.

GridGain provides a unified API which supports SQL, C++, .NET, Java/Scala/Groovy, Node.js and more access for the application layer. The unified API connects cloud-scale applications with multiple data stores containing structured, semi-structured and unstructured data (SQL, NoSQL, Hadoop). It offers a high-performance data environment that allows companies to process ACID transactions and generate valuable insights using ANSI-99 SQL from real-time, interactive and batch queries. In-memory computing platforms offer a strategic approach to in-

memory computing. They deliver performance, scale and comprehensive capabilities which include those of in-memory databases (IMDBs), in-memory data grids (IMDGs), streaming analytics engines, and other in-memory-based point solutions in one solution.

Unlike in-memory databases, GridGain can work on top of existing databases and requires no rip-and-replace or any changes to an existing RDBMS. Users can keep their existing RDBMSs in place and deploy GridGain as a layer above it. GridGain can even automatically integrate with different RDBMS systems, such as Oracle, MySQL, Postgres, DB2, Microsoft SQL server and others. This automatic integration feature generates the application domain model based on the schema definition of the underlying database and then loads the data. Moreover, IMDBs typically only provide a SQL interface while GridGain provides a much wider ecosystem of supported access and processing paradigms in addition to ANSI SQL. GridGain supports key/value stores, SQL access, MapReduce, HPC/MPP processing, streaming/CEP processing and Hadoop acceleration, all-in-one well-integrated in-memory computing platform.

When comparing GridGain with in-memory data grids, it should be noted that an in-memory data grid is just one of the capabilities that GridGain provides. In addition to the data grid function, GridGain also supports HPC/MPP processing, distributed SQL, streaming, clustering, and Hadoop acceleration, allowing for a much broader set of use cases than a typical IMDG.

GridGain can also be deployed as a distributed, transactional in-memory database. When used as an IMDB, the solution provides ACID transaction guarantees with ANSI-99 SQL compliance including DDL and DML support. The optional Persistent Store feature solves memory overflow problems and allows fast restarts because GridGain can process data both in memory and on disk while data is being loaded into memory during warm up.

GridGain Software Editions

The GridGain Professional Edition is a binary build of Apache Ignite™ created by GridGain, which includes optional LGPL dependencies, such as Hibernate L2 cache integration and Geospatial Indexing. It benefits from ongoing QA testing by GridGain engineers and contains bug fixes which have not yet been released in the Apache Ignite™ code base. It is suitable for small-scale deployments which do not require advanced resilience or enterprise-grade security.

The GridGain Enterprise Edition adds enterprise-grade features, including data center replication, enterprise-grade security, rolling upgrades, and more. The Enterprise Edition is extensively tested by GridGain and is recommended for production use in large-scale or mission-critical deployments, or environments with heightened security requirements.

The GridGain Ultimate Edition enables you to put GridGain Persistent Store into production with confidence. The Ultimate Edition includes all the features of the GridGain Enterprise Edition plus a Cluster Snapshots feature for automated backups of the GridGain cluster. When using GridGain with the Persistent Store feature, the system provides the capabilities of a distributed transactional SQL database. The system holds the full dataset on disk with all or a portion of the dataset maintained in memory. SQL queries can be run on the combined memory/disk dataset which provides fast data processing which is improved by increasing the amount of data held in memory.

All GridGain editions are subscription-based products which are included with GridGain Support.

Architecture

GridGain is JVM-based distributed middleware software. It is based on a homogeneous cluster topology implementation that does not require separate server and client nodes. All nodes in a GridGain cluster are equal and can play any logical role per runtime application requirement.

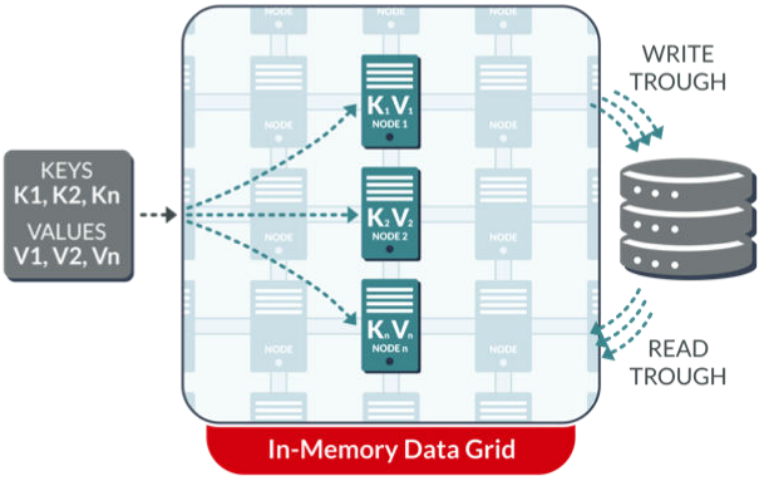
At the core of GridGain is a Service Provider Interface (SPI) design. The SPI-based design makes every internal component of GridGain fully customizable and pluggable by the developer. This enables tremendous configurability of the system, with adaptability to any existing or future server infrastructure.

Another core tenet of GridGain is the direct support for parallelization of distributed computations based on Fork/Join, MapReduce or MPP-style processing, the largest implementation ecosystem of distributed processing algorithms. GridGain uses distributed parallel computations extensively internally and they are fully exposed at the API level for user-defined functionality.

Key Features

IN-MEMORY DATA GRID

One of the core GridGain capabilities is an in-memory data grid. The data grid handles distributed in-memory data management including ACID transactions, failover and advanced load balancing, ANSI-99 SQL including DDL and DML support and many other features. The GridGain IMDG is a distributed, object-based, ACID transactional, in-memory key-value store. GridGain stores its data in memory as opposed to traditional database management systems, which utilize disk as their primary storage mechanism. By utilizing system memory rather than disk, GridGain is orders of magnitude faster than traditional DBMS systems.



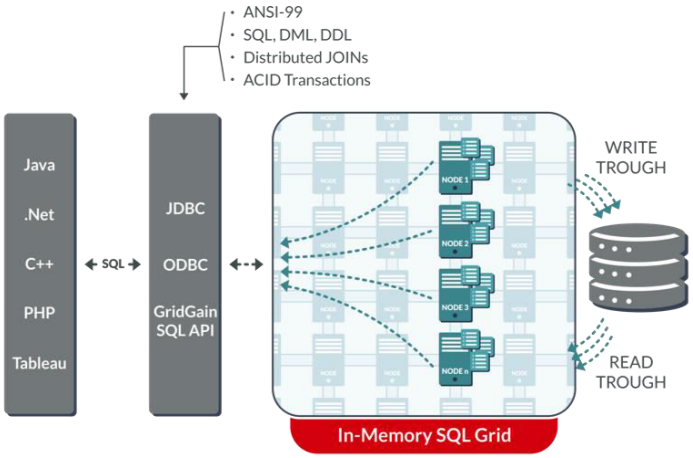
The primary benefits and capabilities of the **In-Memory Data Grid** include:

- ANSI SQL-99 queries with distributed joins
- Lightning-fast performance
- Distributed in-memory caching
- Elastic scalability to handle up to petabytes of data in-memory
- Distributed in-memory ACID transactions
- Distributed in-memory queue and other data structures
- Web session clustering
- Hibernate L2 cache integration
- Tiered off-heap storage
- Unique deadlock-free transactions implementation for fastest speed in in-memory transaction processing

IN-MEMORY DATABASE

The GridGain in-memory database leverages the systems distributed SQL capabilities. It is horizontally scalable, fault tolerant and ANSI-99 SQL compliant. It also supports all SQL, DDL and DML commands including SELECT, UPDATE, INSERT, MERGE and DELETE queries and CREATE and DROP table.

The SQL syntax is ANSI SQL-99 compliant. GridGain can use any SQL function, aggregation, or grouping. GridGain supports distributed SQL joins and allows for cross-cache joins. Joins between partitioned and replicated caches work without limitations while joins between partitioned data sets require that the keys are collocated. GridGain supports the concept of fields queries as well to help minimize network and serialization overhead.



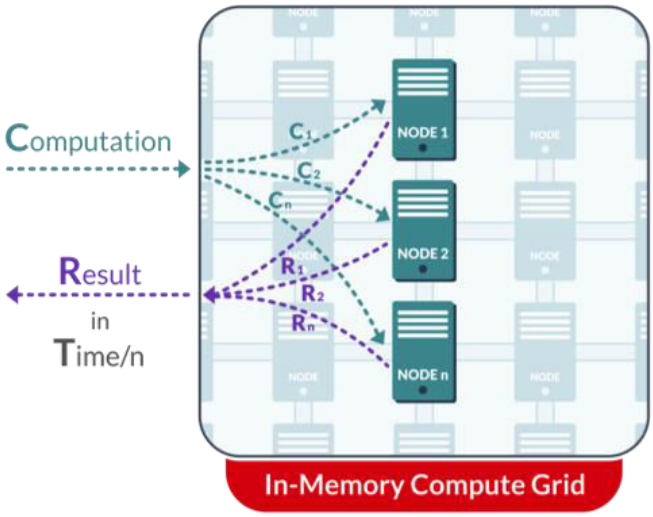
The in-memory distributed SQL capabilities allow users to interact with the GridGain platform not only with the usage of natively developed APIs for Java, .NET and C++ but also using standard SQL commands through the GridGain JDBC or ODBC APIs. This provides a true cross-platform connectivity from languages such as PHP, Ruby and more.

The primary capabilities of the **In-Memory Distributed SQL** include:

- ANSI SQL-99 compliance
- Full support for SQL and DML commands including SELECT, UPDATE, INSERT, MERGE and DELETE
- Support for DDL commands including CREATE and DROP table
- Distributed SQL
- SQL communication through the GridGain JDBC and ODBC APIs without custom coding
- Geospatial support

IN-MEMORY COMPUTE GRID

GridGain includes a compute grid which enables parallel, in-memory processing of CPU-intensive or other resource-intensive tasks, including traditional High Performance Computing (HPC) and Massively Parallel Processing (MPP).

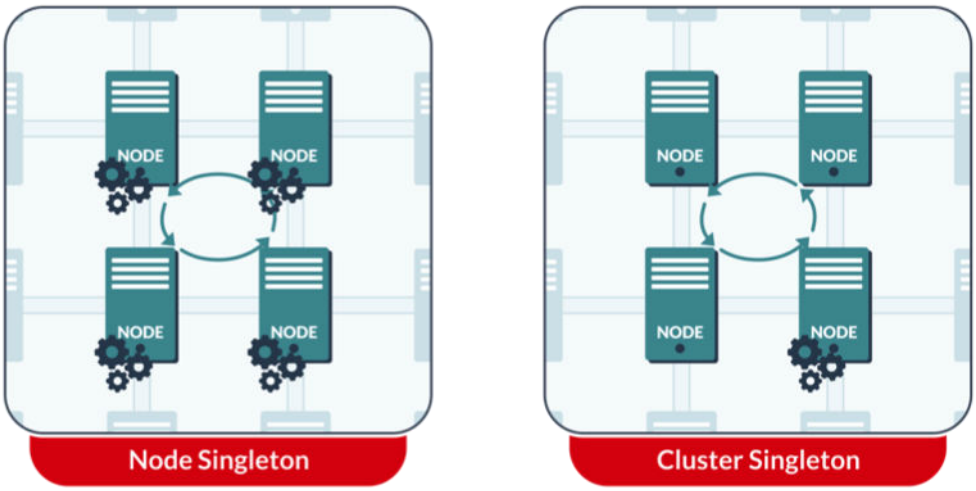


The primary capabilities of the **In-Memory Compute Grid** include:

- Dynamic clustering
- Fork-Join and MapReduce processing
- Distributed closure execution
- Load balancing and fault tolerance
- Distributed messaging and events
- Linear scalability
- Standard Java ExecutorService support

IN-MEMORY SERVICE GRID

The GridGain Service Grid provides users with complete control over services being deployed on the cluster. It allows users to control how many instances of their service should be deployed on each cluster node, ensuring proper deployment and fault tolerance. The Service Grid guarantees continuous availability of all deployed services in case of node failures.

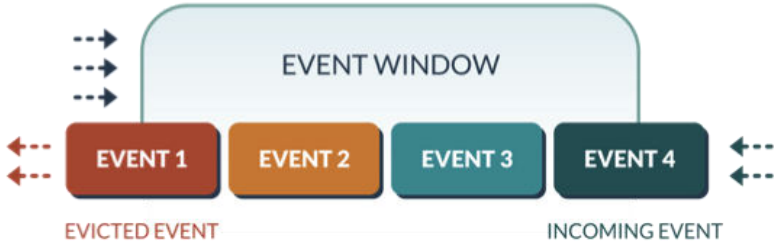


The primary capabilities of the **In-Memory Service Grid** include:

- Automatic deployment of multiple instances of a service
- Automatic deployment of a service as singleton
- Automatic deployment of services on node start-up
- Fault tolerant deployment
- Removal of deployed services
- Retrieval of service deployment topology information
- Remote access to deployed services via service proxy

IN-MEMORY STREAMING

In-memory streaming processing addresses a large family of applications for which traditional processing methods and disk-based storages, such as disk-based databases or file systems, are inadequate. Such applications are extending the limits of traditional data processing infrastructures.



Streaming support enables querying so-called rolling windows of incoming data to enable users to answer questions such as “What are the 10 most popular products over the last 2 hours?” or “What is the average product price in a certain category for the past day?”.

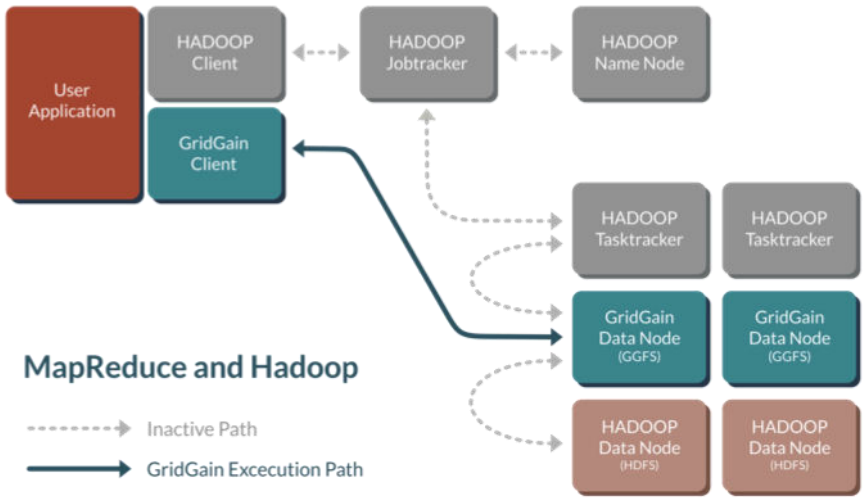
Another common use case for stream processing is controlling and properly pipelining a distributed events workflow. As events are coming into the system at high rates, the processing of events is split into multiple stages and each stage has to be properly routed within a cluster for processing.

The primary capabilities of **In-Memory Streaming** in GridGain include:

- Programmatic window-based querying
- Customizable event workflow / Complex Event Processing (CEP)
- At-least-once guarantee
- Built-in, user-defined sliding windows
- Streaming data indexing
- Distributed streaming queries
- Co-location with in-memory data grid

IN-MEMORY HADOOP ACCELERATION

The GridGain accelerator for Hadoop enhances existing Hadoop environments by enabling fast data processing using the tools and technology your organization is already using today.



In-Memory Hadoop Acceleration in GridGain is based on the industry's first dual-mode, high-performance in-memory file system that is 100% compatible with Hadoop HDFS and an in-memory optimized MapReduce implementation. In-memory HDFS and in-memory MapReduce provide easy to use extensions to disk-based HDFS and traditional MapReduce, delivering up to 100x faster performance.

This plug-and-play feature requires minimal to no integration. It works with open source Hadoop or any commercial version of Hadoop, including Cloudera, HortonWorks, MapR, Apache, Intel, AWS, as well as any other Hadoop 1.x or Hadoop 2.x distribution.

The main capabilities of **In-Memory Hadoop Acceleration** include:

- Up to 100x faster performance for MapReduce and HIVE jobs
- In-memory MapReduce
- Highly optimized in-memory processing
- Dual mode – standalone Ignite File System (IGFS) file system & primary caching layer for HDFS
- Highly tunable read-through and write-through behavior

DISTRIBUTED IN-MEMORY FILE SYSTEM

One of the unique capabilities of GridGain is a file system interface to its in-memory data called the Ignite File System (IGFS). IGFS delivers similar functionality to Hadoop HDFS, including the ability to create a fully functional file system in memory. IGFS is at the core of the GridGain In-Memory Accelerator for Hadoop.

The data from each file is split on separate data blocks and stored in cache. Developers can access the data in each file with a standard Java streaming API. For each part of the file a developer can calculate an affinity and process the file's content on corresponding nodes to avoid unnecessary networking.

The primary capabilities of the **Distributed In-Memory File System** include:

- Standard file system "view" on in-memory data
- Listing of directories or information for a single path
- Create/move/delete of files or directories
- Write/read of data streams into/from files

ADVANCED CLUSTERING

The GridGain in-memory computing platform provides one of the most sophisticated clustering technologies on Java Virtual Machines (JVM). With GridGain, nodes can automatically discover each other. This helps scale the cluster when needed, without having to restart the entire cluster. Developers can also take advantage of the hybrid cloud support in GridGain which allows users to establish connections between private clouds and public clouds such as Amazon Web Services or Microsoft Azure.

The main capabilities of **Advanced Clustering** in GridGain include:

- Dynamic topology management
- Automatic discovery on LAN, WAN, and public clouds
- Automatic "split-brain" (i.e., network segmentation) resolution
- Unicast, broadcast, and group-based message exchange
- On-demand and direct deployment
- Support for virtual clusters and node groupings

DISTRIBUTED MESSAGING

GridGain provides high performance, cluster-wide messaging functionality to exchange data via publish- subscribe and direct point-to-point communication models.

The primary capabilities of **Distributed Messaging** include:

- Support for topic-based publish-subscribe model
- Support for direct point-to-point communication
- Pluggable communication transport layer
- Support for message ordering
- Cluster-aware message listener auto-deployment

DISTRIBUTED EVENTS

The distributed events functionality in GridGain allows applications to receive notifications about cache events occurring in a distributed grid environment. Developers can use this functionality to be notified about the execution of remote tasks or any cache data changes within the cluster.

In GridGain, event notifications can be grouped together and sent in batches and/or timely intervals. Batching notifications help attain high cache performance and low latency.

The main capabilities of **Distributed Events** in GridGain include:

- Subscribing of local and remote listeners
- Ability to enable and disable any event
- Local and remote filters for fine-grained control over notifications
- Automatic batching of notifications for enhanced performance

DISTRIBUTED DATA STRUCTURES

GridGain allows for most of the data structures from the `java.util.concurrent` framework to be used in a distributed fashion. For example, you can take `java.util.concurrent.BlockingDeque` and add to it on one node and poll it from another node. Or you could have a distributed Primary Key generator, which would guarantee uniqueness on all nodes.

Distributed Data Structures in GridGain includes support for these standard Java APIs:

- Concurrent map
- Distributed queues and sets
- AtomicLong
- AtomicSequence
- AtomicReference
- CountDownLatch

UNIFIED API

The GridGain unified API supports a wide variety of common protocols for the application layer to access data. Supported protocols include:

- SQL
- JAVA
- C++
- .NET
- PHP
- MapReduce
- Scala
- Groovy
- Node.js

GridGain supports several protocols for client connectivity to Ignite clusters including Ignite Native Clients, REST/HTTP, SSL/TLS, and Memcached.

GRIDGAIN PERSISTENT STORE

The GridGain Persistent Store feature is a distributed ACID and ANSI-99 SQL-compliant disk store available in Apache Ignite that transparently integrates with GridGain as an optional disk layer. It may be deployed on spinning disks, solid state drives (SSDs), Flash, 3D XPoint or other similar storage technologies. Persistent Store keeps the full dataset on disk while putting only user-defined, time-sensitive data in memory. With Persistent Store enabled, you do not need to keep all active data in memory or warm up your RAM following a cluster restart to utilize the system's in-memory computing capabilities.

The Persistent Store keeps the superset of data and all the SQL indexes on disk, which allows GridGain to be fully operational from disk. The combination of this new feature and the platform's advanced SQL capabilities allows GridGain to serve as a distributed transactional SQL database, spanning both memory and disk. Persistent Store allows your organization to maximize your return on investment by establishing the optimal tradeoff between infrastructure costs and application performance by adjusting the amount of data that is kept in-memory.

Features:

- Distributed ACID and ANSI-99 SQL-compliant disk store
- May be deployed on spinning disks, Solid State Drives (SSDs), Flash, 3D XPoint or similar storage technologies
- Keeps the superset of data and all the SQL indexes on disk
- Allows GridGain to be fully operational from disk
- Can function as a distributed transactional SQL database, spanning both memory and disk

Key Integrations

APACHE SPARK™

Apache Spark is an open source fast and general purpose engine for large-scale data processing. GridGain and Spark are complementary in-memory computing solutions that can be used together in many instances to achieve superior performance and functionality.

Apache Spark and GridGain address somewhat different use cases. They rarely "compete" for the same task. Some differences:

Solution	Apache Spark	GridGain
Data Retention	Apache Spark loads data for processing from other storages, usually disk-based, and discards the data when the processing is finished. It doesn't store data.	GridGain provides a distributed in-memory key-value store (distributed cache or data grid) with ACID transactions and SQL querying capabilities which retains data in memory and can write through to an underlying database
OLAP/OLTP	Apache Spark is for non-transactional, read-only data (RDDs don't support in-place mutation) so it is used for OLAP	GridGain supports non-transactional (OLAP) payloads as well as fully ACID compliant transactions (OLTP)
Data Types	Apache Spark is based on RDDs and works only on data-driven payloads	GridGain fully supports pure computational payloads (HPC/MPP) that can be "data-less"

Apache Spark is for in-memory processing of event-driven data. Spark doesn't provide shared storage, so ETL-ed data from HDFS or another disk storage must be loaded into Spark for processing. State can only be passed from Spark job to job by saving the processed data back into external storage. GridGain can share Spark state directly in memory, without storing the state to disk.

The GridGain Shared RDD API is one of the main integrations for GridGain and Apache Spark. GridGain RDDs are essentially wrappers around GridGain caches which can be deployed directly inside of Spark processes that are executing Spark jobs. GridGain RDDs can also be used with the cache-aside pattern, where GridGain clusters are deployed separately from Spark, but still in-memory. The data is still accessed using Spark RDD APIs.

GridGain RDDs are used through IgniteContext which is the main entry point into GridGain RDDs. It allows users to specify different GridGain configurations. GridGain can be accessed in client mode or server mode. Users can create new shared RDDs, which means new GridGain caches are created with different configurations and different indexing strategies. GridGain supports a variety of partitioning and replication strategies with fully replicated or partitioned caches.

Everything that can be done in GridGain can be done with IgniteContext by passing a proper GridGain configuration. The RDD syntax is native so it can be accessed using the native Spark RDD syntax. The main difference is GridGain RDDs are mutable while Spark RDDs are immutable. Mutable GridGain RDDs can be updated at the end of or during every job or task execution and ensure that other applications and jobs can be notified and can read the state.

Apache Spark Plus GridGain for Faster SQL Queries

Apache Spark supports a fairly rich SQL syntax but it doesn't support data indexing, so Spark must do full scans all the time. Spark queries may take minutes even on moderately small data sets. GridGain supports SQL indexes, resulting in much faster queries, so Spark SQL can be accelerated over 1,000x when using Spark with GridGain. The result set returned by GridGain Shared RDDs also supports Spark Dataframe API, so it can be further analyzed using standard Spark data frames as well. Both Apache Spark and GridGain natively integrate with Apache YARN and Apache Mesos so they can easily be used together.

Shared In-Memory File System with Apache Spark Plus GridGain

When working with files instead of RDDs, it is still possible to share state between Spark jobs and applications using the Apache Ignite In-Memory File System (IGFS). IGFS implements the Hadoop FileSystem API and can be deployed as a native Hadoop file system, just like HDFS. GridGain plugs in natively to any Hadoop or Spark environment. The in-memory file system can be used with zero code changes in plug-n-play fashion.

APACHE CASSANDRA™

Apache Cassandra can be a high performance solution for structured queries. However, Cassandra requires that the data must be modeled such that each pre-defined query results in one row retrieval. This pre-planning requires knowledge of the required queries before modeling the data.

While very powerful in certain use cases, Cassandra lacks an in-memory option which can severely limit performance. Cassandra can be a good match for OLAP applications but lacks support for transactions, ACID or otherwise, so it is not used for OLTP. Cassandra can be efficient for pre-defined queries but lacks SQL support and does not support joins, aggregations, groupings, or usable indexes. Cassandra is useful for pre-defined queries but does not support ad hoc queries.

GridGain offers native support for Apache Cassandra to address the limitations noted above. When used together, GridGain provides Cassandra users with very powerful new capabilities such as the ability to leverage in-memory computing to reduce query times by 1,000x. With GridGain, Cassandra users can leverage ANSI-99 compliant SQL support to run ad hoc and structured queries and to perform joins, aggregations, groupings and usable indexes.

Summary

GridGain is the leading open source in-memory computing platform. It is a high-performance, integrated and distributed in-memory solution for computing and transacting on large-scale data sets in real-time. It performs orders of magnitude faster than is possible with traditional disk-based or flash technologies. As an in-memory

data management software layer, it sits between applications and various data sources, and does not require the rip-and-replacement of existing databases.

GridGain comprises, in one well-integrated framework, a set of key in-memory capabilities, including:

- An in-memory data grid
- An in-memory transactional SQL database
- An in-memory compute grid
- In-memory streaming processing
- An in-memory service grid
- In-memory acceleration for Hadoop

Despite the breadth of its feature set, GridGain is very easy to use and deploy. There are no custom installers. The code base comes as a single .zip file with only one mandatory dependency: ignite-core.jar. All other dependencies, such as integration with Spring for configuration, can be added to the process a la carte. The project is fully Mavenized and is composed of over a dozen Maven artifacts that can be imported and used in any combination. GridGain is based on standard Java APIs. For distributed caches and data grid functionality, GridGain implements the JCache (JSR107) standard.

The GridGain large scale, distributed in-memory framework offers transactional and analytical applications performance gains of 100 to 1,000 times faster throughput and/or lower latencies. GridGain is an important open source foundation that holds the key to the world of Fast Data across high-volume transactions, real-time analytics and the emerging class of hybrid transaction/analytical workloads (HTAP).

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)7775 835 770 (Europe), or complete our contact form to have us contact you.

About GridGain Systems

GridGain Systems is revolutionizing real-time data access and processing by offering an in-memory computing platform built on Apache® Ignite™. GridGain solutions are used by global enterprises in financial, software, e-commerce, retail, online business services, healthcare, telecom and other major sectors, with a client list that includes Barclays, ING, Sberbank, Misys, IHS Markit, Workday, and Huawei. 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 and HTAP. For more information, visit gridgain.com.