WHITE PAPER

GridGain

High Performance, Real-time Data Architecture for the Internet of Things

A GridGain Systems In-Memory Computing White Paper

OVERVIEW

Over the last decade, Internet of Things (IoT) adoption has exploded. Today there are more connected devices than people. These devices range from smart phones and wearables, to connected vehicles, to industrial equipment, and more. By 2020, Gartner expects the IoT to have over 20 billion connected things, a conservative estimate compared to other analysts.

The information generated by billions of connected devices requires an enormous amount of real-time processing and storage. Some connected cars now upload 25GB of data per hour. A connected airplane can generate a petabyte a flight. But to realize the benefits of IoT, you need to decide which use cases to tackle first. You also need to choose the right architecture and set of technologies that can process large data streams, identify important events and react in real-time.

Many companies who have succeeded with IoT have solved their challenges around speed, scalability and real-time analytics with in-memory computing. Across these deployments some common architectural patterns have emerged. This whitepaper explains some of the most common use cases and challenges; the common technology components, including in-memory computing technologies; and how they fit into an IoT architecture. It also explains how Apache Ignite™ (Ignite) and GridGain® are used for IoT.

IOT USE CASES AND OPPORTUNITIES

According to 451 Research, 65% of companies are using IoT and 69% of organizations are gathering data from end points while 94% of the companies that collect that data use it for business purposes. The highest usage is among Utilities (92%) and Manufacturing (77%).

Some of the more common use cases that we've seen across our customer base include:

- Smart homes – smart meters and home automation
- Wearables – clothes, watches, findable keys, virtual and augmented reality devices
- Manufacturing, retail, and industrial – sensory tracking for production line monitoring
- Transportation – intelligent traffic control and self-driving vehicles

Smart Homes

Smart meters are already implemented in the USA, and British Gas is working toward government targets of having smart meters in all UK businesses and homes by 2020. Many companies are also working on home automation systems, with use cases emerging around thermostats and other sensing devices, as well as robotics and tracking. Applications are being developed around pets, kitchen appliances, and gardens, as well as other areas of the home.

Wearables

The Android and Apple watches were early entries in this category. Smart clothes are also in the works, with details of how they will interact still being worked out. Another major use case involves Bluetooth key rings and other types of tags that let users tag physical items and track them in a digital world.

The wearables category also includes devices such as headsets that allow users to explore virtual or augmented reality. Facebook's substantial investment in Oculus, maker of the Oculus Rift headset, helped convince other investors and developers to focus on virtual reality. This triggered substantial innovation in new sensors, new human-computer interaction devices, and new ways to map physical reality to virtual reality.

Manufacturing

Using sensors to track production lines is becoming a useful way to streamline industrial processes and improve uptime through increased preventative maintenance. This type of tracking is already happening in car manufacturing and other industries.

One example of how sensors have been helpful is in the process of oil drilling. Every time an oil drill breaks, it causes a very expensive disruption in oil production. It does not just reduce labor productivity. It also reduces oil revenues. Sensors provide a way to identify patterns and predict when well heads are going to fail. By tracking these leading indicators and taking action before the well heads fail, one company has been able to increase oil revenues by ten percent.

Smart Homes

With both ride-sharing companies and taxi drivers using apps to broadcast anonymous information about their travel, there is a lot of data becoming available for intelligent traffic control. Finding a better route through traffic is a very popular use case that is likely to complement the expanding development of self-driving vehicles by Google, Mercedes, and other companies. It is also helping improve supply chain planning and logistics.

BASIC COMPONENTS OF IOT

Connected devices themselves may look very different, and their data bandwidth and processing requirements vary widely. However, at a high level these IoT deployments have a very similar set of building blocks:

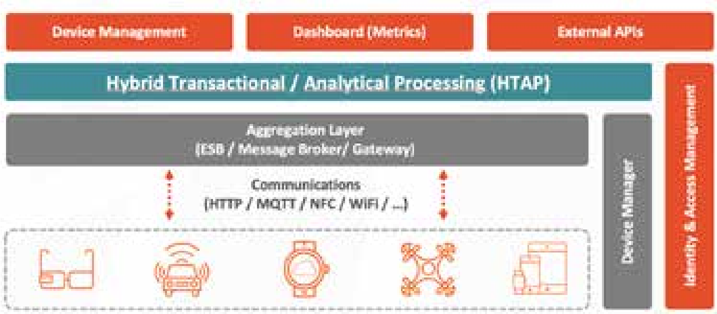

Figure 1. IoT Reference Architecture

- **Things (a.k.a., devices) that can communicate digitally**. For example, smart phones, cars, and a wide-range of items tagged with sensors or built around small, single-board processors or microcontrollers (for example, Raspberry Pi, BeagleBone, or Arduino.)
- **Communication mechanisms**. Through which the devices communicate with backend servers either directly, in the case of intelligent devices communicating through Wi-Fi or Ethernet, or indirectly, as when readers gather data from Bluetooth devices or devices with NFC or RFID tags and send the data through a gateway.
- Server-side infrastructure. For processing the data from these devices and performing any real-time analytics and automation, as well as for providing device and data to users. While it may be referred to as the data center, the server-side infrastructure is often distributed and can be any combination of on-premise, private and public cloud infrastructure.
- **Aggregation layer**. Data from devices is streamed or "micro-batched" using a host of transport layers. While many deployments implement their own protocols over HTTP or SSL, others might use IoT-centric messaging protocols such as MQTT, synchronous APIs, or other messaging protocols ranging from JMS to Kafka™. A gateway and an Enterprise Service Bus (ESB) or message broker-based might be used for security, policy management, routing and transformation.
- **Transactional and analytical processing layer**. As the data arrives into a data center, any combination of hybrid transactional/analytical processing (HTAP) can take place. The ability to perform transactions and analytics against the same up-to-date data is a key IoT requirement that we will explore in greater detail as part of the architecture section.
- **Device management**. All devices need to be centrally managed. This includes the ability to provision and de-provision devices, manage device authentication and authorization, manage user access, and perform maintenance tasks. There is both a server (a device manager) and a (device management) user console.
- **Metric dashboard**. Business intelligence and analytics tools are needed to provide reporting and dashboards for end-to-end visibility, adhoc analytics and streaming analytics that help sense and respond to events as they occur.
- **External APIs**. Most IoT implementations provide integration and end user access through APIs that are secured and managed using various API management tools.
- **Identity and access management**. All device, user and API or application security is typically integrated with identity and access management software that allows administrators to centrally manage authentication and authorization rights across all layers of the architecture.

THE CHALLENGES OF IOT

Given the basic similarities in architecture across IoT projects, it should not be surprising that the challenges across projects are also similar. First is the ability to provision, secure and manage thousands, even millions of devices and a distributed network. Next is the ability to then stream, process, analyze and react to massive amounts of real-time information in real-time. For most companies, these are very new challenges. And even the most experienced technical teams, these challenges are very difficult to solve.

Scale

Typical deployments can involve thousands to millions of connected devices, and petabytes or more of data. The ability to take this amount of streaming information from so many devices and derive useful meaning from this data in near real-time is very challenging.

Speed

Many projects require an architecture that supports transactions and real-time analytics using the latest data. In the example of connected well heads, the analytics need to be processed in time to prevent a complete failure and unplanned downtime.

Distributed Architecture

Many deployments require anytime, anywhere availability. With IoT, it is not just that devices must remain connected at all times from any location. Their location and regional connectivity matters too. Devices need to be provisioned and balanced across different gateways and data centers to handle the data volumes and maximize performance.

Security

Most devices are outside the firewall, which makes securing the devices and related information critical. With the large number of connected sensors and other devices, there will be a lot of personal information being relayed at all times — information about people's houses, locations, health, and so on. Traditional device, user and API level authentication and authorization, along with data encryption are not enough. There also needs to be location-based security monitoring and attack prevention. For example, an attempted connection from Europe to a connected car in the USA should be identified as a possible attack.

Omni-Channel

This term originated in e-commerce. It refers to the ability to perform a single task, such as a financial transaction, across multiple contact channels – from bank branches, ATMs and call centers to web applications and mobile devices. Whether people are using phones, watches, laptops, cars, or other devices, they should be able to see the same up-to-date information, perform the same operations, and end up with the same results.

Transaction and Analytical Workloads

For any IoT applications that require near real-time responsiveness, the data infrastructure has to support both transactions and analytic workloads. Otherwise decisions are made against stale data. In the case of the connected well head, it is not enough to simply schedule a truck roll to the well head to perform maintenance. Real-time analytics also need to identify the impact of the well head maintenance on the overall system against the most up-to-date information. Having a separate ETL process with out-of-date information no longer works.

IOT ARCHITECTURES AND THE RISE OF HTAP

Many components of an IoT architecture have evolved from other types of projects. For example, data distribution has evolved from more traditional messaging technologies over the last few decades. Device management has evolved from mobile device management. But HTAP is different. Whenever analytics have been needed to support a decision during a transaction, the traditional approach has been to create a new data pipeline that Extracts, Transforms and Loads (ETL) data with minimal impact from applications and databases to data marts, data warehouses and data lakes. Over time, for operational business intelligence, companies began to try and accelerate their ETL architecture using expensive hardware. But ETL creates a copy of the data that is almost immediately out of date. That is a problem for real-time analytics supporting IoT.

As data sets grew larger over the last decade, some companies turned to the Lambda architecture defined by Nathan Marz for IoT applications. The Lambda architecture (see Figure 2a) combined stream-and batch-based processing to get the scalability of batch with the low latency of real-time processing.

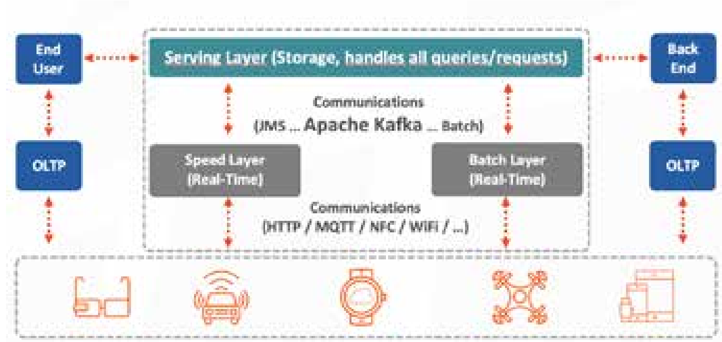

Figure 2a. Lambda Architecture

The primary capabilities of the Ignite's hybrid IMDB include:

- A batch/storage layer—a batch-based data processing and storage layer, such as Apache Hadoop® with Hive™, or a specialized data warehouse that handled all batch-based input and large data sets.
- A high-speed layer—a real-time data management and processing pipeline, such as Apache Spark™, that could take the latest data and process it quickly to provide more up-to-date visibility.
- Data from the two pipelines were then merged for analytics to provide near real-time visibility. Companies accepted the additional complexity of building and merging two separate pipelines out of necessity. At the time, it was nearly impossible to cost-effectively process "big data" in real-time without an approach like the Lambda architecture. But this didn't work well for real-time IoT applications. By the time the analytical data came out of the high-speed layer of the lambda architecture, it was outdated. What was needed instead was the ability to process live and historical data together in real-time.

IoT architectures changed with the rise of new in-memory computing technologies like Apache Ignite. Ignite is able to perform both transactions and analytics against massive data sets with linear horizontal scalability. The growth of Ignite as a hybrid in-memory database (IMDB) that could store petabytes of data provided an alternative to the traditional Lambda architecture. The use of Ignite and other HTAP technologies led to the rise of the IoT HTAP architecture (Figure 2b.).

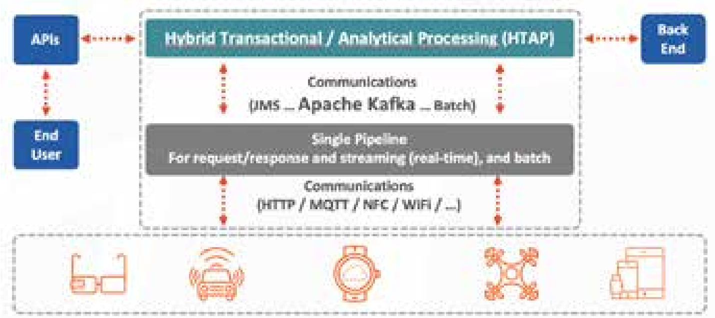

Figure 2b. IoT HTAP Architecture

With HTAP companies began to implement a single real-time data architecture that could support transactions and large scale real-time and batch analytics. The question now is which HTAP technologies to choose for IoT.

HTAP OPTIONS FOR IOT

Several options now existing for implementing HTAP for IoT. One approach has been to use the SMACK stack, or some combination of Apache Spark, Mesos™, Akka™, Cassandra™, and Kafka™:

- Spark as the overall compute framework and processing engine
- Mesos to manage containers
- Akka for event processing
- Cassandra as a distributed database
- Kafka as the messaging layer for streaming data

There are other technologies as well, such as HTAP databases. While these technologies are becoming quite established, there are still several challenges in using them for IoT that you need to consider.

Skillsets

First, significant up-front learning investment is required if you don't know all the technologies. While it is possible to find people who know each technology in the SMACK stack, it is not so easy to find people who know all of them. Maintaining the SMACK stack over time can also be quite complex.

HTAP

Not all databases are well suited for HTAP. For example, Cassandra is not well-suited for real-time HTAP because Cassandra sacrificed immediate consistency for scalability. This happened because at the time it made sense given the current state of technologies, just like the Lambda architecture made sense. Writes in Cassandra are written as "append only" records to a Memtable. When the Memtable becomes full it is flushed to an SSTable, also as append-only records. A value only becomes consistent once the SSTable is compacted. That is when all write appends are consolidated with the last value of the data. As a result, Cassandra is not appropriate for any transactions that require immediate consistency. With Cassandra, for example, you could withdraw too much from a bank account balance or over-sell the existing inventory and only find out once the SSTable is compacted. Cassandra is also not well-suited for analytics. It does not support SQL, which is the de-facto data access and analytics language of choice. The Cassandra Query Language (CQL) is limited to one (big) table and doesn't even support basic analytics features such as joins.

In-Memory Computing

Third, real-time IOT requires integrated in-memory processing and data management. While Spark is a great in-memory stream processing engine, it requires a lot of coding to make HTAP run fast. That is because Spark has no out-of-the-box in-memory data management, and the SMACK stack does not help much. Cassandra does not simplify data access and storage for Spark in part because it does not support SQL or accelerate SparkSQL. Cassandra's disk-based storage and retrieval also slows down Spark's in-memory processing each time Spark has to access data.

No Rip and Replace

HTAP databases can be a good option for a new SaaS application with no legacy infrastructure. But most companies have existing applications or data that must be integrated with IoT, and HTAP databases do not sit on top of other databases. They replace them. This is a problem because most of the time you cannot easily replace a database or move data. You need to integrate the data together. For example, data from devices might need to be accessible across existing channels. And eventually any new IoT application will hopefully become a legacy application that must be integrated with future initiatives. After all, the definition of a legacy application is that it got used. So, what HTAP architecture can you adopt that will integrate more effectively with existing applications and databases?

APACHE IGNITE FOR IOT HTAP

One architecture that has emerged in place of SMART is a combination of Kafka for messaging; Ignite as the core HTAP layer; and Spark for stream processing. In order to understand Ignite's role, one must first understand what Ignite is, and how it works.

Apache Ignite is the leading Apache Software Foundation (ASF) project for in-memory computing. It is one of the top 5 ASF projects in terms of commits and email list activity, with over twice the level of commits compared to other in-memory computing projects such as Spark. GridGain Systems contributed the code that became Ignite to the Apache Software Foundation and continues to be the project's lead contributor. The GridGain in-memory computing platform is the only enterprise-grade, commercially supported version of Ignite. It includes enterprise-grade security, deployment, management and monitoring capabilities which are not in Ignite plus global support and services for business-critical systems.

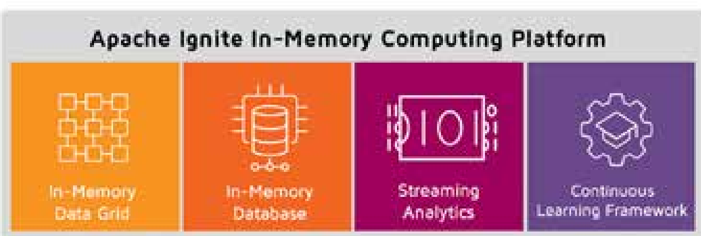

Figure 3. Apache Ignite In-Memory Computing Platform

Companies use Ignite and GridGain to add in-memory speed and unlimited horizontal scalability to:

- Existing or new OLTP or OLAP applications
- New or existing hybrid transactional/analytical processing (HTAP) applications
- Stream ingestion, processing and analytics
- Continuous learning use cases involving machine or deep learning

Ignite is perfectly suited to provide HTAP for IoT because it brings transaction processing, with support for ANSI-99 SQL and ACID transactions, together with analytics and more generic computing on an in-memory data storage and computing layer. With Ignite all stream and batch processing for transactions, analytics and any decision automation can be performed through a single in-memory computing layer in real-time.

Ignite and GridGain are used by the largest companies in the world to ingest, process, store and publish streaming data for large-scale, mission critical business applications. They are used by several of the largest banks in the world for trade processing, settlement and compliance; by telecommunications companies to deliver call services over telephone networks and the Internet; by retailers and e-commerce vendors to deliver an improved real-time experience; and by leading cloud infrastructure and SaaS vendors as the in-memory computing foundation of their offerings. Companies have been able to ingest and process streams with millions of events per second on a moderately-sized cluster.

IOT HTAP ARCHITECTURE WITH KAFKA, IGNITE AND SPARK

In an IoT HTAP architecture, Ignite acts as the HTAP data management layer. Ignite provides a collection of integrated functionality:

- Extensive support for third-party messaging and other streaming technologies, including extensive integration with Spark
- An in-memory data grid (IMDG) layer for all data access that sits on top of third-party databases on the backend systems or Ignite's native persistence and supports both transactions and analytics
- Ignite's native persistence, which acts as a distributed in-memory database (IMDB)
- A compute and service grid that enables massively parallel processing (MPP).
- A continuous learning framework for machine and deep learning

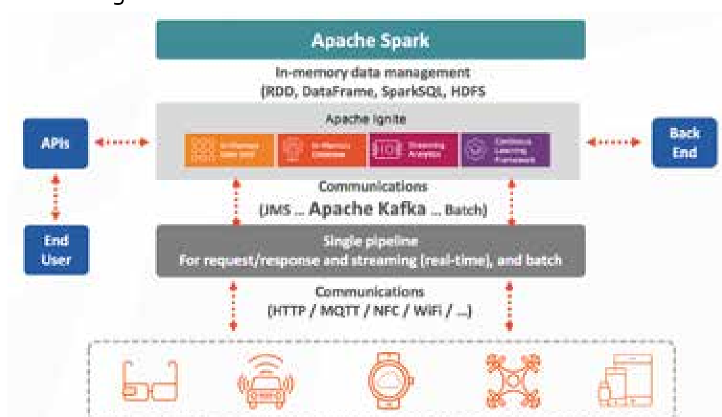

Figure 4. Ignite as Part of an IoT HTAP Architecture

Third-Party Messaging and Streaming Technologies

Ignite is integrated and used with major streaming technologies including Apache Camel™, Kafka, Spark, Storm™, Java Message Service (JMS) and MQTT to ingest, process and publish streaming data.

In-Memory Data Grid (IMDG) for Adding Speed and Scalability to Applications

A core Ignite capability and most common use case is as an IMDG. Ignite can increase the performance and scalability of existing applications and databases by sliding in-between the application and data layer with no rip-and-replace of the database or application and no major architectural changes. Ignite supports all common RDBMSs including IBM DB2, Microsoft SQL Server, MySQL, Oracle and PostgreSQL, NoSQL databases such as Cassandra and MongoDB, and Hadoop.

Ignite generates the application domain model based on the schema definition of the underlying database. It then loads the data and acts as the data platform for the application. Ignite handles all reads and coordinates transactions with the underlying database in a way that ensures data consistency in both the database and Ignite. By utilizing RAM in place of disk, Ignite lowers latency by orders of magnitude compared to traditional disk-based databases.

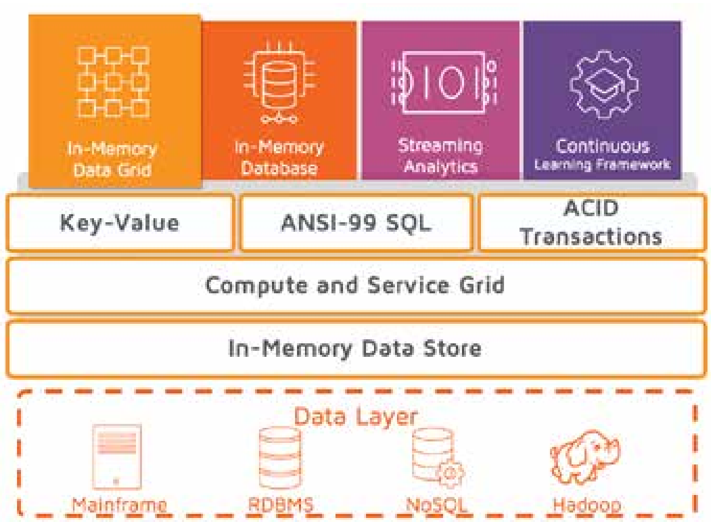

Figure 5. Ignite as an In-Memory Data Grid (IMDG)

The primary benefits and capabilities of the Ignite IMDG include:

- ANSI SQL-99 support including DML and DDL
- ACID transaction support
- In-memory performance orders of magnitude faster than disk-based RDBMSs
- Distributed in-memory caching that offloads queries from the existing database
- Elastic scalability to handle up to petabytes of in-memory data
- Distributed in-memory queue and other data structures
- Web session clustering
- Hibernate L2 cache integration
- Tiered off-heap storage
- Deadlock-free transactions for fast in-memory transaction processing
- JCache (JSR 107), Memcached and Redis client APIs that simplify migration from existing caches

Hybrid In-Memory Database (IMDB) for High Volume, Low Latency Transactions and Data Ingestion

An Ignite cluster can also be used as a distributed, transactional IMDB to support high volume, low latency transactions and data ingestion, or for low cost storage.

The Ignite IMDB combines distributed, horizontally scalable ANSI-99 SQL and ACID transactions with Ignite's native persistence. It supports all SQL, DDL and DML commands including SELECT, UPDATE, INSERT, MERGE and DELETE queries and CREATE and DROP table. Ignite parallelizes commands whenever possible, such as distributed SQL joins. It allows for cross-cache joins across the entire cluster, which includes joins between data persisted in third-party databases and Ignite's native persistence. It also allows companies to put 0-100% of data in RAM for the best combination of performance and cost.

The in-memory distributed SQL capabilities allow developers, administrators and analysts to interact with the Ignite platform using standard SQL commands through JDBC or ODBC or natively developed APIs across other languages as well.

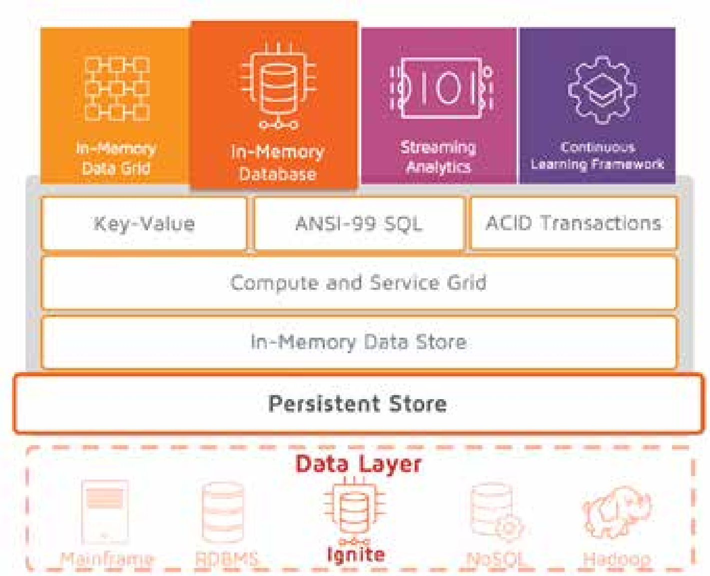

Figure 6. Ignite as an IMDB

The primary capabilities of the Ignite's hybrid IMDB include:

- ANSI SQL-99 compliance
- ACID transactions support
- Full support for SQL DML including SELECT, UPDATE, INSERT, MERGE and DELETE
- Support for DDL commands including CREATE and DROP table
- Support for distributed SQL joins, including cross-cache joins across the entire cluster
- SQL support through JDBC and ODBC without custom coding
- Geospatial support
- Hybrid memory support for RAM, HDD, SSD/Flash, 3D XPoint and other storage technologies
- Support for maintaining 0-100% of data in RAM with the full data set in non-volatile storage
- Immediate availability on restart without having to wait for RAM warm-up

Continuous Learning Framework

Ignite also provides Ignite Machine Learning (ML), MPP-style machine learning and deep learning with real-time performance on petabytes of data. Ignite provides several standard machine learning algorithms optimized for collocated processing including linear and multi-linear regression, k-means clustering, decision trees, k-NN classification and regression.

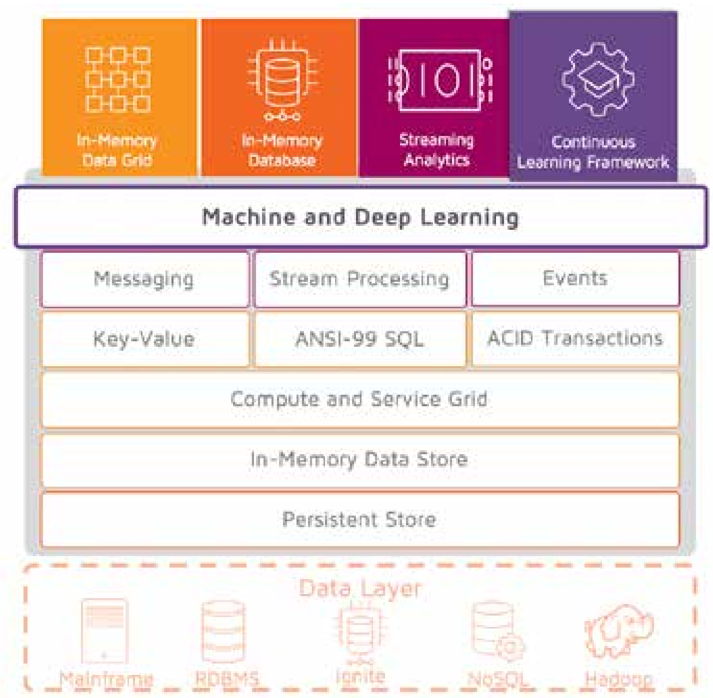

Figure 7. Ignite for Machine Learning and Deep Learning

It also includes a multilayer perceptron for deep learning along with TensorFlow integration. Developers can develop and deploy their own algorithms across any cluster as well as using the compute grid.

Compute Grid

Ignite provides a compute grid which enables parallel, in-memory processing of CPU-intensive or other resource-intensive tasks. It can be used for any High-Performance Computing (HPC) applications that leverage Massively Parallel Processing (MPP). The compute grid helps optimize overall cluster performance by collocating processing with data to optimize data processing and minimize network traffic. The system includes a comprehensive library of functions that includes machine and deep learning. Developers can develop and distribute their own code for any combination of transactions, analytics, stream processing or machine learning using Java, .NET or C++. The primary capabilities of the compute grid include:

- Zero code (peer-class loading) deployment
- 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
- Collocated processing support for multiple languages including Java, .NET and C++

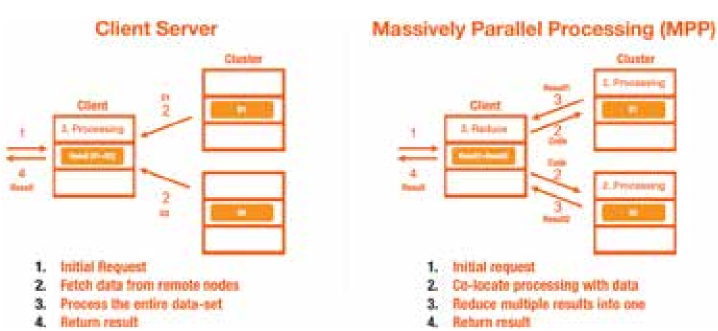

Figure 8. Ignite Compute Grid – Client Server vs Collocated Processing (MPP)

They can also leverage data affinity with collocated processing to achieve linear scalability as data sets grow. Ignite is event-driven with support for messaging and events, which makes it easy to identify and act on important events as part of an HTAP IoT architecture.

GridGain

Service Grid

Ignite provides a service grid to deploy and scale microservices across the cluster for digital business and other initiatives. It allows users to control how many instances of their service are deployed on each cluster – as a cluster singleton, node singleton, or as multiple instances across the cluster. The service grid guarantees continuous availability of all deployed services in case of node failures, including guaranteeing a single cluster or node singleton, or load balancing with multiple instances across the cluster.

Streaming Analytics with Apache Spark

Apache Spark is an open source fast and general-purpose engine for large-scale data processing of event-driven streaming data in memory. Ignite is the ideal underlying in-memory data management technology for Apache Spark because of its in-memory support for managing stored "data at rest" and ingesting and processing streaming "data in motion".

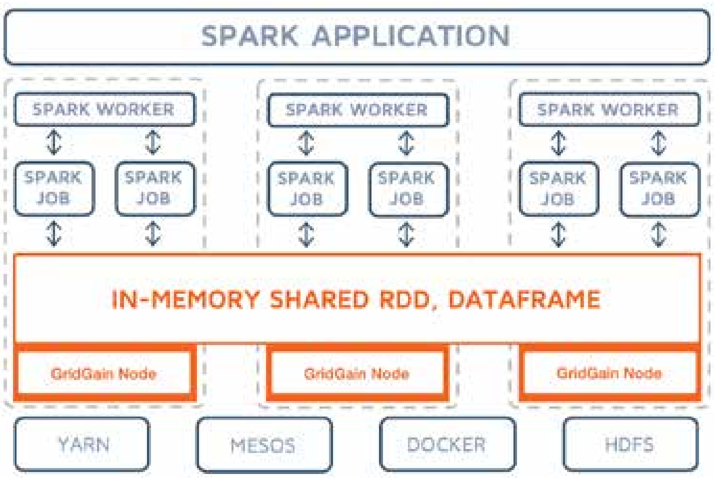

Figure 9. Ignite and Apache Spark

Ignite provides the broadest in-memory computing integration with Apache Spark. Native support for Spark DataFrames allows Spark developers to access data from and save data to Ignite to share both data and state across Spark jobs. The Ignite RDD API lets developers read from and write to Ignite caches as mutable RDDs, unlike existing immutable Spark RDDs. Both the RDD and DataFrame support make Ignite caches accessible locally in RAM inside Spark processes executing Spark jobs.

Ignite also integrates its distributed SQL into SparkSQL plans. This allows Spark to take advantage of the advanced indexing and MPP-style distributed joins in Ignite. The combination can improve Spark SQL query performance by as much as 1000x.

The Ignite File System (IGFS) provides in-memory access via HDFS. Spark developers are able to leverage all of Ignite's in-memory storage and processing capabilities including machine learning to train models in near real-time to improve outcomes for in-process HTAP applications.

WHAT GRIDGAIN ADDS TO IOT

Business-critical IoT requires more than just processing. The majority of devices require additional security because they are outside traditional firewalls and often hold personal or business-sensitive information. Any external access, which typically are supports using APIs, should also be scalable and secure. In addition, it's important to ensure data is never lost, maintain always on availability, and provide complete end-to-end visibility into operations.

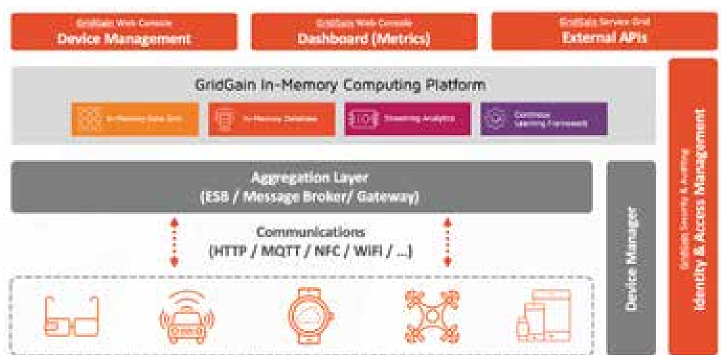

Figure 10. GridGain IoT Reference Architecture

This is where GridGain is typically used instead of Ignite. GridGain is 100% compatible with Ignite. What GridGain adds is the enterprise-grade security, deployment, management and monitoring capabilities critical to securing, monitoring and managing data, device management and communications.

GridGain is the only enterprise-grade, commercially supported version of the Apache Ignite open source project. GridGain Systems contributed the code that became Ignite to the Apache Software Foundation and continues to be the project's lead contributor. GridGain is 100% compatible with Ignite. GridGain adds enterprise-grade security, deployment, management and monitoring capabilities to Ignite. GridGain Systems also offers global support and professional services for business-critical systems. With Apache Ignite, patches are only released as part of each software release from the ASF, which happen every 3-6 months. Even though the community is committed to improving Ignite, there is no guarantee that a critical patch you might need makes it into the next release. GridGain Systems provides commercial SLAs with rapid response times and the ability to provide software patches much faster as needed. GridGain Systems also offers a professional services organization that has assisted with deployments across a wide range of customer use cases to help ensure your success and speed up your in-memory computing deployment.

GRIDGAIN SUPPORT

GridGain is the only company to provide commercial support for Apache Ignite. GridGain Basic Support for Apache Ignite and the GridGain Community Edition includes timely access to professional support via web or email. The team can help troubleshoot performance or reliability issues and suggest workarounds or patches, if necessary. A two-hour initial consultation allows our support team to understand your current environment for more effective support in the future. The consultation helps identify issues and improve the performance or reliability of your deployment.

Standard Support is for companies deploying the GridGain Enterprise Edition or the GridGain Community Edition in production. With 24x7 support hours and web, email and phone access, Standard Support is perfect for ongoing production deployments. An annual license to the GridGain Enterprise or Community Edition is available with the subscription.

Premium Support is for companies deploying the GridGain Enterprise Edition or the GridGain Ultimate Edition for mission-critical applications. Premium Support is available 24x7 with the fastest initial response time, more named support contacts than Standard Support, and web, email and phone access. Premium Support is available with a license to the GridGain Enterprise or Ultimate Edition.

SOLUTION	UNSUPPORTED	BASIC SUPPORT	STANDARD SUPPORT	PREMIUM SUPPORT
Software	Ignite/Community	Ignite/Community	Community/Enterprise	Enterprise/Ultimate
Troubleshoot Performance or Reliability Issues		●	●	●
Identify Bugs and Product Limitations & Workarounds		●	●	●
Support Hours		9x5	24/7	24/7
Initial Response Time		4 Hours	2 Hours	1 Hour
Unlimited Support Incidents		●	●	●
Named Technical Contacts		2	3	4
Initial Consultation Support		2 Hours		
Maintenance Releases			●	●
Hot Bug Fixes			●	●
Enhanced Features (Enterprise & Ultimate Editions)			●	●
Custom Contracts			●	●
Login Support to Customer Environment			●	●
Support Channels		Web/Email	Web/Email/Phone	Web/Email/Phone

Figure 11. GridGain Support Level Comparison

SUMMARY

HTAP has emerged as a clear successor to Lambda and other architectures in IoT. By using in-memory computing to implement HTAP, companies have been able to implement a single data ingestion and processing pipeline that supports both transactional and analytical needs, and delivers the speed, scalability and real-time responsiveness required by IoT. The combination of Apache Kafka, Ignite and Spark has become a clear open source in-memory computing solution for implementing HTAP for IoT. GridGain's enterprise-grade security, monitoring and management capabilities, along with professional services and 24x7 support, has helped make this combination a preferred choice for business-critical IoT.

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 the form at www.gridgain.com/contact 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 www.gridgain.com.