WHITE PAPER

GridGain

In-Memory Computing Options for SAP Customers

When to Use SAP HANA, and When to Use Something Else

If you're one of the four hundred thousand SAP customers worldwide, then you know that in 2025, SAP S/4HANA will be the only option for running SAP ERP applications. Your only database option for SAP will be SAP HANA. This does not just impact your SAP applications. It also impacts SAP customizations and the integration with the rest of your IT infrastructure that support your business processes.

Any SAP S/4HANA adoption, along with customization and integration, will be part of the larger journey to innovate and to improve the customer experience. Any new infrastructure must support the ever-increasing need for speed, scale, agility and a host of new technologies. Existing IT infrastructure, including SAP ERP, have been unable to support these innovations on their own. Over the last decade, the growth of digital business, IoT and other technologies have helped increase query volumes and transaction loads 10-1000 times. The amount of data collected has growth by 50 times.

Speed is also a major challenge. New customer-facing web and mobile apps and their underlying APIs all require sub-second roundtrip latencies. These are nearly impossible to support given all the network hops, software layers and data queries and joins that need to happen between the new customer apps and existing applications (such as SAP). In addition, the amount of data needed for different types of analytics and other types of computing—including machine and deep learning—is too big to move across the network fast enough.

Companies need an architecture that not only prepares them for the migration to SAP S/4HANA, but supports speed, scale and rapid change. Digital transformation and other initiatives must deliver new capabilities in days or weeks in order to keep pace with the customer expectations and the competition, especially new "digital" entrants. But most existing applications are inflexible to change. Even packaged applications such as SAP S/4HANA cannot change in days. Most applications take months to deliver for even minor changes. Most applications also don't support many newer technologies like streaming analytics or machine and deep learning, which are required to help streamline, automate and improve real-time processes.

This white paper explains your options for adding speed, scale and agility to end-to-end IT infrastructure—from SAP HANA to third-party vendors and open source. It also explains how to evolve your architecture over time for speed and scale, become more flexible to change, and support new technologies as needed.

THE CHALLENGE WITH SPEED, SCALE, AGILITY, AND AUTOMATION

Adopting new web, mobile, and other self-service channels, adding personalization and other automation, and using new types of data require greater performance and scalability than what even SAP HANA can cost-effectively deliver on its own. The reason is that speed and scale is needed not just at the database, but at other systems layers as well.

Over the last decade, these innovations have led to query and transaction volumes growing 10 to 1000 times. They have resulted in 50 times more data about customers, products, and interactions. They also shrunk the expected end-to-end response times from days or hours to seconds (or less). It means the roundtrip latency required from a mobile device calling an API that accesses applications over the network (including SAP ERP, and eventually at some point, SAP HANA) must always be less than one second.

It is impossible to deliver all of this on top of the existing architecture with existing applications and databases. For one, vertical scalability is not a long-term option because the 10-1000 times growth rates are faster than Moore's Law. This trend shows no signs of slowing down. Even if you spend more on hardware initially, eventually any single server will fail to keep up with the growth. Latency is also a major issue. The combination of multiple network hops and queries that involve large data sets makes current architectures practically impossible. As one architect put it: "You cannot violate the laws of physics." You need to lower end-to-end latency, not just database latency.

Even if you could address the challenges with speed and scale, the applications themselves introduce a third major challenge. Customers have come to expect rapid change. The new innovators that are disrupting just about every business model deliver new capabilities in days or weeks. Existing applications are inflexible to change in comparison. They take months or years to change. While SAP S/4HANA is a modern application, it is nonetheless a packaged application. You cannot make customizations and deploy them in days.

Then there is the final challenge: automation. Streamlining a process, creating a one-click shopping experience, or monitoring for issues and proactively fixing them all need you to implement real-time intelligence and automation. Existing applications and databases were architected in a way that separates online transactions processing (OLTP) from online analytical processing (OLAP), data warehousing

GridGain

and business intelligence. OLTP systems could not take the additional load and were not designed for analytics. Each system solved a point problem. To create a single view of the business for analytics companies had to:

- Extract, Transform and Load (ETL) data into a warehouse
- Cleanse data
- Enrich it
- Make it consistent across different types of data

This was all to make reporting and ad hoc analytics work. At the time, neither the freshness of the data nor the speed of reporting was as important as the accuracy.

Now the analytics and the decisions must happen in real-time—often even during a transaction—in order to deliver a great sub-second experience. That is impossible to do with existing applications because they do not include analytics or other technologies such as machine or deep learning.

THE EVOLUTION TOWARDS IN-MEMORY COMPUTING

Many innovators have been able to address these challenges with speed, scale, agility and automation on top of an architecture that Gartner calls Hybrid Transactional/Analytical Processing (HTAP). One of the most common foundations for HTAP is in-memory computing.

Many companies started by implementing an in-memory data grid (IMDG) in-between existing databases and applications (instead of buying additional database licenses or adding new database hardware). This offloaded reads from the databases (giving the databases substantially more room for growth on their existing hardware) and lowered latency.

But their IMDG deployments also gave them a way to evolve the agility of their existing applications by unlocking the application data for use by any other projects. For each project, as more data was added into their common in-memory computing infrastructure, all data could be used together to:

- Perform real-time analytics
- Add speed and scale for new APIs
- Process new types of data for streaming analytics
- Help with automation by adding machine and deep learning capabilities

These deployments also helped with agility because new workloads were easy to add without impacting existing applications. You only had to add more nodes as needed. Ignite is also cloud-native, making its use in a DevOps environment straightforward.

Adopting HTAP and in-memory computing is the right long-term approach for adding speed, scale, agility and for building a real-time business. But there are other shorter- and longer-term options for helping with speed and scale. Companies that rely on SAP do have SAP HANA as a foundation for SAP applications. SAP HANA is the only long-term viable database for SAP applications. But outside of SAP ERP, for other applications and use cases, there are other options. This white paper provides a comparison of SAP HANA and some of the more popular options, as well as guidelines for how to move towards in-memory computing over time. What follows first is a discussion of SAP HANA.

ADDING SPEED AND SCALE TO SAP APPLICATIONS WITH SAP HANA

SAP describes HANA as an in-memory data platform, which is essentially an in-memory database (IMDB) to improve the performance of SAP applications including SAP ERP and SAP BW (Business Warehouse). HANA's main market is as an IMDB for SAP ERP and SAP BW and the successor products, SAP S/4HANA and SAP BW/4HANA. Out of the four hundred thousand SAP customers, approximately thirty thousand use HANA, and less than ten thousand have adopted SAP S/4HANA. HANA implements HTAP, which means it can perform online transactional processing (OLTP) and online analytical processing (OLTP) together. It is proprietary, not open source. While its primary server-side language is SQL, it also supports server-side C++ and JavaScript.

HANA is the only viable long-term choice for SAP applications, in part because by 2025 it will be the only choice according to SAP. But it is a very expensive option for non-SAP applications. One analyst firm, Brightwork, called it the most expensive database they track. When purchased as an SAP application-specific license, the HANA runtime typically cost 8% of the SAP application price, 15% of the application and BW license with no discounts. But when purchased for full use, for both SAP and non-SAP technologies, the pricing is per GB of data. While there is a low cost option, HANA One on AWS, which is roughly $0.99/hr for every 60GB up to 244GB, it is very expensive for larger deployments. In 2014 one estimate was $5.9M for Hana Enterprise over 4 years, and $2.7M for a 1TB Hana Cloud deployment.

Given the pricing, it might not come as a surprise that SAP HANA is not typically used for large deployments. But the current deployments and limits on scalability indicate HANA is not well suited for managing big data. The largest deployment cited in 2016 was 135TB. When you bring your own licenses and run on Amazon, the maximum sizes available are either 12TB on a single instance, or 50TB when scaled horizontally. While SAP cites 10x data compression, real-world examples have cited 2.5-3.3x compression, which is change: more in line with other leading columnar databases.

Beyond high price and limitations in the cloud, HANA has other limitations with scalability. HANA can be scaled horizontally, and tables partitioned manually based on any key, including foreign keys. But it does require manual rebalancing. Also, while HANA can execute SQL, C++ and JavaScript locally on the servers, it is not a general-purpose compute grid for massively parallel processing (MPP) that collocates other code with the data and aggregates the results. It does not support Java or .NET MPP, for example.

HANA also does not support most other in-memory computing use cases outside of an IMDB as well as other in-memory computing technologies. It cannot act as an in-memory data grid (IMDG), which means it is not a good choice for providing real-time access to data that isn't stored in HANA. It also means HANA is not as good for real-time analytics involving SAP and non-SAP data unless it is acceptable for the non-SAP data to go through an Extract-Transform-Load (ETL) process and be slightly out of date. For streaming analytics, while HANA does have its own streaming analytics capabilities, it not support streaming analytics with Apache Spark. SAP Vora, which is integrated with HANA, can accelerate SQL queries against data in Hadoop. These results can be more easily joined with data in HANA.

HANA does support running a long list of machine learning algorithms in place with the data. There is integration with R that allows R to be invoked, though it is not supported out of the box. You will need to install and configure the environment. You can use HANA to score TensorFlow models after TensorFlow has been used externally for training and building the models. You can also build other algorithms and deploy them. But HANA is not as well suited for machine or deep learning model training and building, partly because it is so expensive, and partly because it lacks the integration to do in-place TensorFlow model training and building.

ADDING SPEED AND SCALE BEYOND SAP APPLICATIONS

The main option for adding speed and scale outside of SAP is not HANA. It is to use a more general-purpose in-memory computing platform that provide some combination of IMDG, IMDG, streaming analytics and machine and deep learning.

The end goal of in-memory computing is to move data into memory for speed, and to use a combination of a shared-nothing architecture and MPP for linear, horizontal scale for all data-intensive workloads. To perform HTAP you need both existing data in relational databases, including SAP HANA and new data like streaming Web interactions or devices or social data that helps you understand customer preferences and relationships.

The most common first step is the use of in-memory computing as an IMDG to accelerate existing applications, for two reasons. First, an IMDG adds in-memory speed and horizontal scalability that is more cost-effective in the longer term than scaling up with expensive hardware.

Do the math. Add up all the expected read and write scalability needs for the next three to five years. Then figure out your long-term options. Most companies discover the following: they can either spend the money now on expensive hardware, and then have to implement an IMDG in the future, or add the IMDG now and slowly grow it to the same size in the future (assuming no other uses).

Second, an IMDG unlocks existing data for new uses, such as for real-time analytics, HTAP, streaming analytics, or machine and deep learning. To support all these projects requires other capabilities, namely IMDB support that is able to store and manage new types of data alongside existing data, streaming analytics support including integration with other streaming technologies like Apache Kafka and Spark, and machine and deep learning support.

Again, do the math. Identify the projects that can be achieved with existing data accessible in memory and add up the ROI over those three to five years. That is money you are losing without an IMDG as part of a broader in-memory computing platform. The ROI on those additional projects should be added to help decide between different in-memory computing technologies and other options.

GridGain

How an IMDG Adds Speed and Scale, and Unlocks Data

An IMDG adds speed and scale by sitting in-between applications and databases, in the path of all reads and writes. It stores all data in memory and keeps the data up to date by supporting a read-through/write-through cache pattern. It receives all writes, writes to memory, and then passes it on to the database as a transaction. If the database transaction succeeds, the IMDG commits to memory as well. Because this keeps all data in the IMDG in sync with the database, the IMDG can then handle all reads directly. This lowers latency for reads because the data is accessed directly from RAM, not a disk-based database. An IMDG also adds scale by offloading all read workloads from the database. Most IMDGs can scale horizontally on commoditized hardware to handle increased read loads without putting additional loads on the database. This is much less expensive than buying specialized database hardware.

The easiest way to slide an IMDG in-between an application and an existing relational database (RDBMS) is for the IMDG to support SQL. If it does not support SQL, then you will need to write new code that replaces your SQL with a key-value API, and more code for the IMDG to access SQL Server.

Most IMDGs also provide some form of massively parallel processing (MPP) where they can divide up data into smaller sets across nodes, and collocate code with the data (like Hadoop). MPP allows horizontal scalability of both the data and computing, similar to the way MapReduce or Spark work. If the data is partitioned so that the computing has all the data it needs on each node, then the data does not need to be fetched over the network. This approach helps eliminate one of the most common performance bottlenecks in big data analytics and general Big Data computing.

Most databases do not support MPP. If moving data over the network is part of the performance issue, scaling the database will not solve the problem. Also, part of the reason for adding an IMDG is to unlock data that is in the existing database, to be able to use it in new projects, including HTAP, without overloading the database or requiring a hardware upgrade. Most databases also do not support real-time analytics or high performance computing at scale. They do not directly support Spark or other streaming analytics technologies. They also do not support general-purpose machine or deep learning. All of these capabilities rely on MPP.

HOW TO MERGE SAP AND NON-SAP ENVIRONMENTS

Given that SAP HANA is the only choice for adding speed and scale to SAP applications, but not the best or most cost-effective choice for non-SAP applications and workloads, most companies need to build a second in-memory computing layer and integrate with SAP via APIs, or integrate data across HANA and their new in-memory computing platform to support analytics and other use cases. The use of APIs, API management, even SOA middleware is relatively well defined, and a level above HANA and other in-memory computing technologies.

Regarding data integration, there is no one rule for how to merge SAP and non-SAP-managed data. It depends more on how you will use the data, and also whether you are willing to spend the extra license cost for allowing HANA to hold non-SAP data. If you are not, then using an ETL process to extract from SAP, by using SAP Business Objects for example, or performing live SQL queries for real-time access, are two common approaches.

Investing in both levels of integration not only are more cost-effective. If you layer properly, providing access via APIs and in-memory computing will give you the flexibility to deliver new capabilities much faster.

GridGain

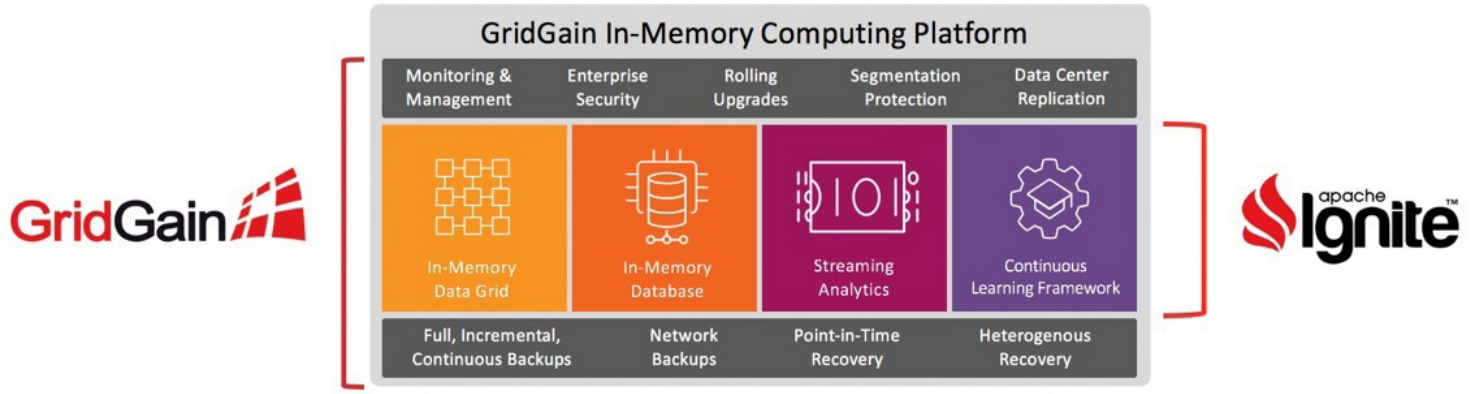

Figure 1. Apache Ignite and the GridGain In-Memory Computing Platform

APACHE IGNITE AND THE GRIDGAIN IN-MEMORY COMPUTING PLATFORM

GridGain is the leading in-memory computing platform for real-time business. It is the only enterprise-grade, commercially supported version of the Apache® Ignite™ (Ignite) open source project. GridGain includes enterprise-grade security, deployment, management, and monitoring capabilities which are not in Ignite, plus global support and services for business-critical systems. GridGain Systems contributed the code that became Ignite to the Apache Software Foundation and continues to be the project's lead contributor.

GridGain and Ignite are used by tens of thousands of companies worldwide to add in-memory speed and unlimited horizontal scalability to existing applications, and then add HTAP to support new initiatives to improve the customer experience and business outcomes. With GridGain, companies have:

- Improved speed and scalability by sliding GridGain in-between existing applications and databases as an IMDG with no rip-and-replace of the applications or databases
- Improved transactional throughput and data ingestion by leveraging GridGain as a distributed IMDB
- Improved the customer experience or business outcomes by adding HTAP that leverages real-time analytics, streaming analytics and continuous learning

GridGain customers have been able to create a new shared in-memory data foundation. This single system of record for transactions and analytics enables real-time visibility and action for their business. With each project, they have unlocked more information for use by other applications on a platform with real-time performance at peak loads and always-on availability. As a result, they can develop new projects faster, are more flexible to change, and are more responsive in ways that have improved their experiences and business outcomes.

ADDING SPEED AND SCALABILITY TO EXISTING APPLICATIONS WITH AN IMDG

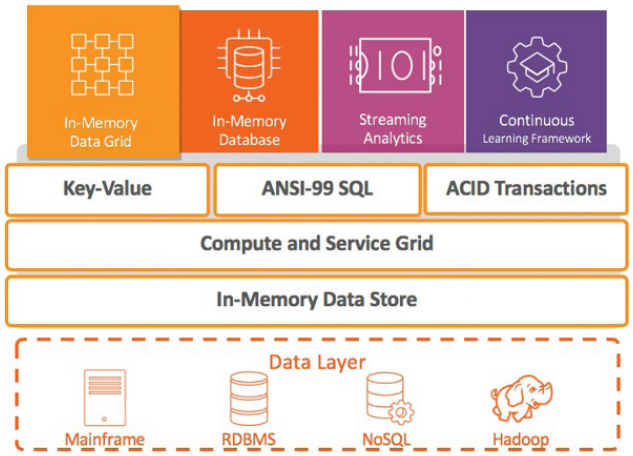

Figure 2. GridGain as an In-Memory Data Grid (IMDG)

One of the core GridGain capabilities and most common use cases is as an IMDG. GridGain 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 without major architectural changes.

This is because GridGain supports ANSI-99 SQL and ACID transactions. GridGain can sit on top of leading RDBMSs including IBM DB2®, Microsoft SQL Server®, MySQL®, Oracle® and Postgres as well as NoSQL databases such as Apache Cassandra™ and MongoDB®. GridGain generates the application domain model based on the schema definition of the underlying database, loads the data, and then acts as the new data platform for the application. GridGain handles all reads and coordinates transactions with the under-

lying database in a way that ensures data consistency in the database and GridGain. By utilizing RAM in place of a disk-based database, GridGain lowers latency by orders of magnitude compared to traditional disk-based databases.

STORING DATA FOR HIGH VOLUME, LOW LATENCY TRANSACTIONS AND DATA INGESTION WITH AN IMDB

A GridGain 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.

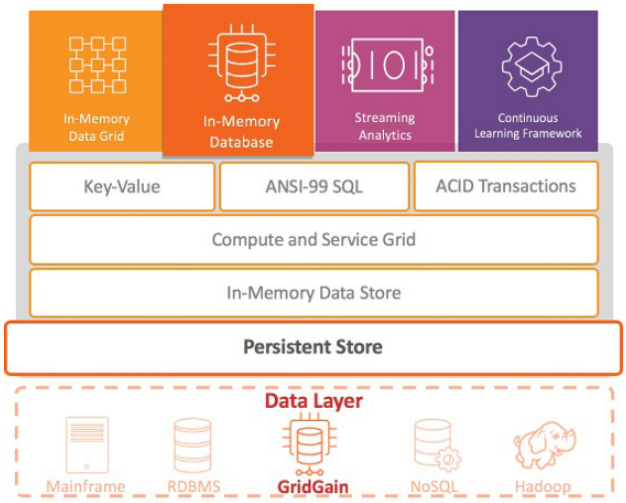

Figure 3. GridGain as an IMDB

The GridGain IMDB combines distributed, horizontally scalable ANSI-99 SQL and ACID transactions with the GridGain Persistent Store. It supports all SQL, DDL and DML commands including SELECT, UPDATE, INSERT, MERGE and DELETE queries and CREATE and DROP table. GridGain 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 the GridGain Persistent Store. 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 GridGain platform using standard SQL commands through JDBC or ODBC or natively developed APIs across other languages as well.

ADDING REAL-TIME ANALYTICS AND HTAP WITH MASSIVELY PARALLEL PROCESSING (MPP)

Once GridGain is put in place, all of the data stored in existing databases or in GridGain is now available in memory for any other use. Additional workloads are easily supported by GridGain with unlimited linear horizontal scalability for real-time analytics and HTAP.

GridGain accomplishes this by implementing a general purpose in-memory compute grid for massively parallel processing (MPP). GridGain optimizes overall performance by distributing data across a cluster of nodes, and acting as a compute grid that sends the processing to the data. This collocates data and processing across the cluster. Collocation enables parallel, in-memory processing of CPU-intensive or other resource-intensive tasks without having to fetch data over the network.

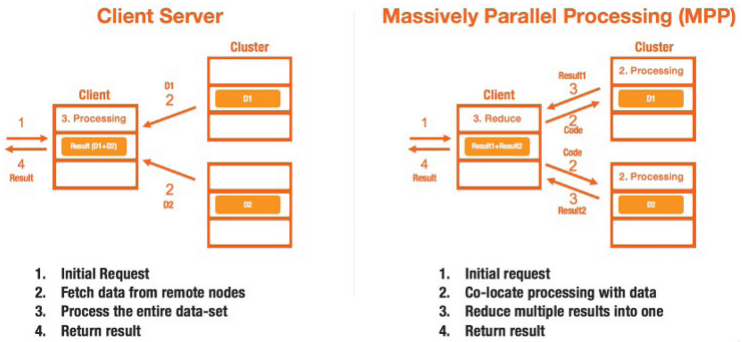

Figure 4. GridGain Compute Grid – Client Server vs Collocated Processing (MPP)

The GridGain Compute Grid is a general-purpose framework that developers can use to add their own computations for any combination of transactions, analytics, stream processing, or machine learning. Companies have used GridGain's MPP capabilities for traditional High-Performance Computing (HPC) applications as well as a host of real-time HTAP applications.

GridGain has implemented all of its built-in computing on the GridGain Compute Grid, including GridGain distributed SQL as well as the GridGain Continuous Learning Framework for machine and deep learning. Developers can write their own real-time analytics or processing in multiple languages, including Java, .NET and C++, and then deploy their code using the Compute Grid.

Collocation is driven by user-defined data affinity, such as declaring foreign keys in SQL DDL (data definition language) when defining schema. Collocation helps ensure all data needed for processing data on each node is stored locally

either as the data master or copy. This helps eliminate the network as a bottleneck by removing the need to move large data sets over the network to applications or analytics.

ADDING DEEPER INSIGHTS AND AUTOMATION WITH STREAMING ANALYTICS AND CONTINUOUS LEARNING

The capabilities GridGain supports are not just limited to real-time analytics that support transactions. GridGain is also used by the largest companies in the world to improve the customer experiences or business outcomes using streaming analytics and machine and deep learning. These companies have been able to incrementally adopt these technologies using GridGain to ingest, process, store and publish streaming data for large-scale, mission critical business applications.

Figure 5. GridGain for Stream Ingestion, Processing and Analytics

GridGain is used by several of the largest banks in the world for trade processing, settlement and compliance. Telecommunications companies use it to deliver call services over telephone networks and the Internet. Retail and e-commerce vendors rely on it to deliver an improved real-time experience. And leading cloud infrastructure and SaaS vendors use it 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.

GridGain is integrated and used with major streaming technologies including Apache Camel™, Kafka™, Spark™ and Storm™, Java Message Service (JMS) and MQTT to ingest, process and publish streaming data. Once loaded into the cluster, companies can leverage GridGain's built-in MPP-style libraries for concurrent data processing, including concurrent SQL queries and continuous learning. Clients can then subscribe to continuous queries which execute and identify important events as streams are processed.

GridGain also provides the broadest in-memory computing integration with Apache Spark. The integration includes native support for Spark DataFrames, a GridGain RDD API for reading in and writing data to GridGain as mutable Spark RDDs, optimized SQL, and an in-memory implementation of HDFS with the GridGain File System (GGFS). The integration allows Spark to:

- Access all the in-memory data in GridGain, not just data streams
- Share data and state across all Spark jobs
- Take advantage of all GridGain's in-memory processing including continuous learning to train models in near real-time to improve outcomes for in-process HTAP applications

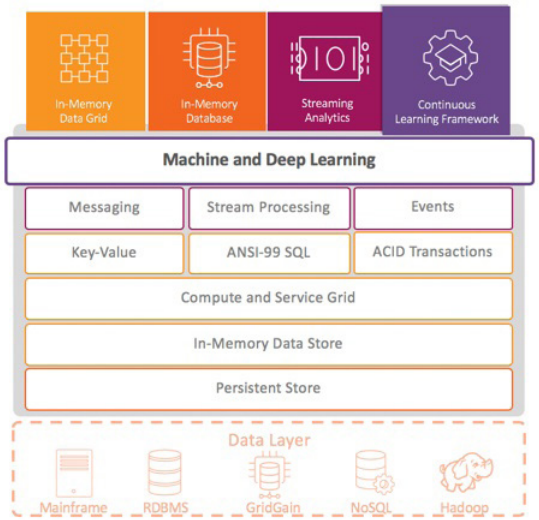

Figure 6. GridGain for Machine and Deep Learning

GridGain also provides the GridGain Continuous Learning Framework. It enables companies to automate decisions by adding machine and deep learning with real-time performance on petabytes of data. GridGain accomplishes this by running machine and deep learning in RAM and in place on each machine without having to move data over the network.

GridGain provides several standard machine learning algorithms optimized for MPP-style processing including linear and multi-linear regression, k-means clustering, decision trees, k-NN classification and regression. It also includes a multilayer perceptron and TensorFlow integration for deep learning. Developers can develop and deploy their own algorithms across any cluster as well as using the compute grid. The result is continuous learning that can be incrementally retrained at any time against the latest data to improve every decision and outcome.

SUMMARY

Applications and their underlying RDBMSs have been pushed beyond their architectural limits by new business needs, and new software layers. Companies have to add speed, scale, agility and new capabilities to support digital transformation and other business critical initiatives. If you are an SAP customer, SAP HANA is the only long-term choice for adding Speed and Scale to SAP. But HANA is not a good choice for non-SAP applications due to cost and partly because its primary function is as an IMDB. For non-SAP environments, the best long term approach is an in-memory computing platform. Not only does it add speed and scale. It provides an incremental approach to adding speed and scale one project at a time that unlocks data across systems and enables companies to be much more agile.

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 at www.gridgain.com/contact and we will contact you.

About GridGain Systems

GridGain Systems is revolutionizing real-time data access and processing with the GridGain in-memory computing platform built on Apache® Ignite™. GridGain and Apache Ignite are used by tens of thousands of global enterprises in financial services, fintech, software, e-commerce, retail, online business services, healthcare, telecom and other major sectors, with a client list that includes ING, Raymond James, American Express, Societe Generale, Finastra, IHS Markit, ServiceNow, Marketo, RingCentral, American Airlines, Agilent, and UnitedHealthcare. GridGain delivers unprecedented speed and massive scalability to both legacy and greenfield applications. Deployed on a distributed cluster of commodity servers, GridGain software can reside between the application and data layers (RDBMS, NoSQL and Apache® Hadoop®), requiring no rip-and-replace of the existing databases, or it can be deployed as an in-memory transactional SQL database. GridGain is the most comprehensive in-memory computing platform for high-volume ACID transactions, real-time analytics, web-scale applications, continuous learning and hybrid transactional/analytical processing (HTAP). For more information on GridGain products and services, visit www.gridgain.com.