WHITE PAPER

GridGain

Adding Speed and Horizontal Scale to SQL Server

In-Memory Computing Options for Microsoft SQL Server Deployments

Relational database management systems (RDBMS) such as Microsoft SQL Server helped standardize the way we store and manage data and develop applications. But applications and their underlying RDBMSs have been pushed beyond their architectural limits by new business needs, and new software layers.

Part of the challenge is about scalability. Over the last decade, the growth of digital business, the internet of things (IoT), and other technologies has on average helped grow query and transaction loads 10-1000x, and the amount of data collected by 50 times

Speed is also a major challenge. 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 applications and databases. In addition, the amount of data needed for different types of analytics and other types of computing—including machine and deep learning—is becoming too big to move across the network fast enough.

The third challenge is with the applications themselves, not databases. Today's initiatives—such as application changes that improve the end-to-end customer experience—must deliver new capabilities in days or weeks. Most existing applications are inflexible to change, however. It takes months just to deliver minor changes. Most applications also do not support many of the newer technologies such as streaming analytics or machine and deep learning, which are required to help streamline, automate, and improve real-time processes.

This white paper explains what your options are for adding speed, scale and agility to end-to-end IT infrastructure—from Microsoft, third-party vendors and open source. It also explains how to evolve your architecture over time. This allows you to not just increase speed and scale, but to increase IT infrastructure flexibility and support digital business and other new initiatives by adding 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 the latest incarnations of Microsoft SQL Server have been able to 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 1000x. They have resulted in 50x more data about customers, products, and interactions. They also shrunk the expected end-to-end response times from days or hours to steconds or less. It means the roundtrip latency required from a mobile device calling an API that accesses applications over the network (that in turn access SQL Server over the network) 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 10-1000x 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.

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 require

GridGain

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 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

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, and one of the most common in-memory computing platforms is Apache® Ignite™—along with GridGain®, its commercial supported version.

With Apache Ignite, companies could adopt in-memory computing incrementally, one project at a time. Most companies started by implementing Ignite as 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.

ADDING SPEED AND SCALE TO SQL SERVER

Adopting HTAP and in-memory computing is the right long-term approach for adding speed, scale, and agility when building a real-time business. But there are other shorter- and longer-term options for helping with speed and scale. Companies that rely on SQL Server do have some options from Microsoft, and more from third-party vendors and the open source community, to address the challenges with speed and scale. This white paper provides a comparison of all these options, as well as guidelines for how to move towards in-memory computing over time.

What follows first is a comparison of the main options available to add speed and scale to SQL Server—including SQL Server Always On Availability Groups, and SQL Server In-Memory OLTP—along with recommendations for when to choose each option.

Scaling Reads (Somewhat) Horizontally with Availability Groups

While Oracle built Parallel Server in the 90s (which eventually became Oracle RAC) to add some horizontal read and write scalability, Microsoft mostly focused on improving vertical write and storage scalability. Today, SQL Server can still only be scaled vertically as a single database instance for transactions, unless you shard data and add extra logic to split up queries and join the results.

However, SQL Server does offer Always On Availability Groups to help scale read workloads horizontally while providing failover for increased reliability and availability. In this mode you can have one to eight secondary servers as part of a cluster that are read-only instances populated via replication. Queries can automatically be routed to and load-balanced across the secondary servers. You can also use distributed availability groups to have a second availability group avail-

able in a different region, with the second group's primary acting as a listener to the first group's primary.

In each group, the secondary servers add the changes from each primary server's Write-Ahead Log (WAL) to its own WAL, and then converts from the WAL to the database, similar to how other CDC-based replication works. You can make the updates to the secondary server WAL be asynchronous or synchronous. It is important to note this will not be immediately consistent. Even with synchronous commits across the WALs, you will still see delays between the primary and secondary instances that lead to eventual consistency. Despite the eventual consistency, SQL Server availability groups do offer automatic failover from a primary to a secondary instance within an availability group, or manual failover to a distributed availability group with no data loss.

Availability groups and distributed availability groups, which are part of Enterprise Edition, are recommended to help increase read scalability and availability. While this can be used to add up to eight times read scalability locally for read workloads, it is limited to eight secondary instances in an availability group. Also, it does not solve write scalability issues. While reads may be offloaded to the secondary instances, writes as well as the data storage can still only scale vertically.

Scaling Vertically with Hardware

The most common way to scale SQL Server is vertically, with hardware. The reason is that writes and database instances cannot be automatically scaled across several instances. You can shard data manually and then add logic into applications to perform operations on each sharded instances, and then merge the results, as with any other database. But that requires a lot of additional work to effectively create a distributed database on your own.

If additional scalability is needed to support a packaged application or an application that is not easy to change, then you will probably need to scale vertically with hardware. There are several hardware vendors that provide companies a choice. But in general, adding horizontal scale at another layer, or making changes to the application are better long-term solutions.

Lowering Latency with RAM: Microsoft SQL Server In-Memory OLTP

If you need to lower latency or increase analytical query speeds, Microsoft provides Microsoft SQL Server In-Memory OLTP. Available in any edition of SQL Server 2016 SP1 or later (SP2 or later recommended), In-Memory OLTP lowers latency by allowing you to move any table into memory. Each in-memory table is made durable with a transaction log, and you can back up each table to disk.

Unlike with Oracle, it is not something you turn on automatically. It requires either implementing it as in-memory up-front, or converting the table and migrating the data. There is a wizard-driven Memory Optimization Advisor to help with migrations. It must be run once for each table. But it will fail to automatically migrate on the first incompatibility, of which there are many. For example, you cannot have foreign keys between in-memory and regular disk-based tables. You can only use AFTER triggers, which means you cannot use triggers for referential integrity during transactions. Certain data types are not supported. You cannot perform cross-database queries that include in-memory tables. You also cannot use CDC or replication for in-memory tables. Availability groups will work.

SQL Server In-Memory OLTP is a reasonable option for lowering latency for specific applications. But if the latency in-between the database and the consuming application is higher than the total latency needed, it is better to fix latency outside of SQL Server.

ADDING SPEED AND SCALE WITH IN-MEMORY COMPUTING

The other main option for adding speed and scale is to use in-memory computing. 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 and new data, such as streaming data from 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 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.

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 in-between an application and Microsoft SQL Server 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.

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

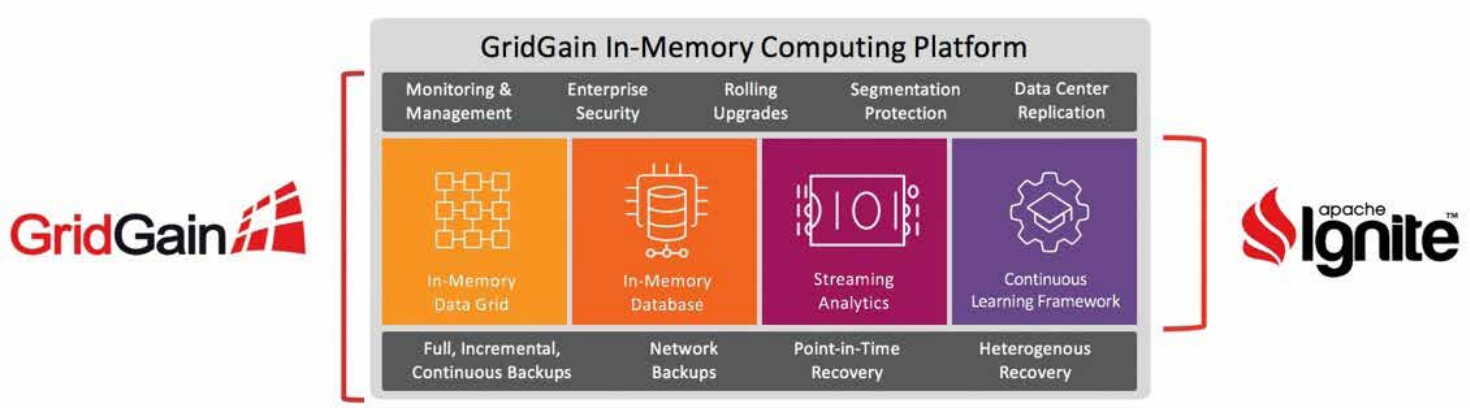

Figure 1. Apache Ignite and the GridGain In-Memory Computing Platform

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

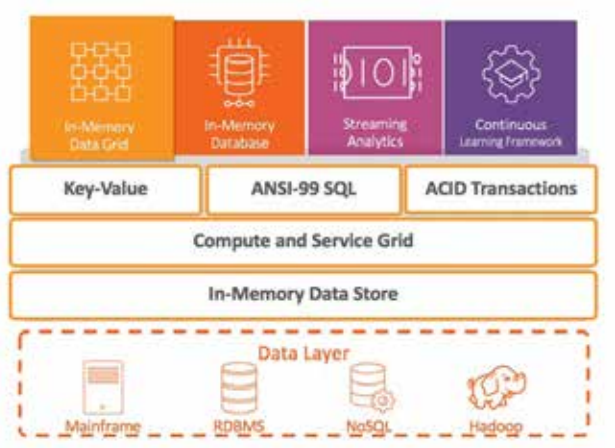

Figure 2. GridGain as an In-Memory Data Grid (IMDG)

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 underlying 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 percent 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.

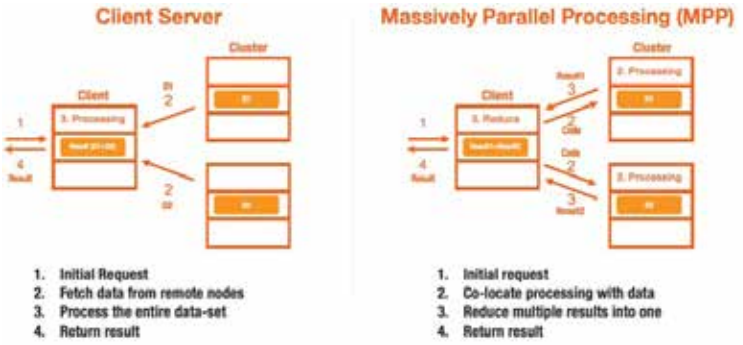

Figure 4. GridGain Compute Grid – Client Server vs. Collocated Processing

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.

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.

Figure 5. GridGain for Stream Ingestion, Processing and Analytics

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

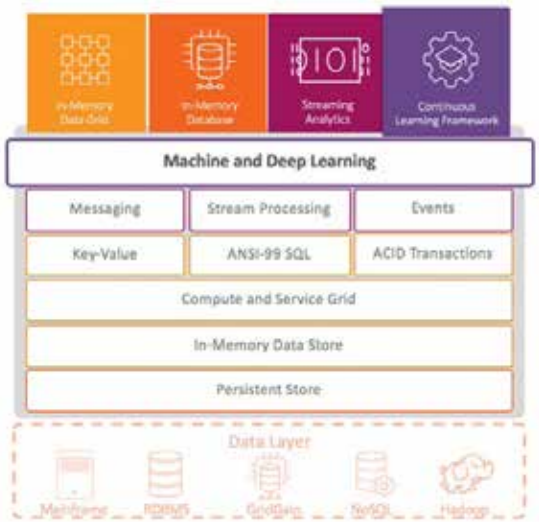

Figure 6. GridGain for Machine and Deep Learning

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. There are some Microsoft options for adding speed and scale to SQL Server—including SQL Server Always On Availability Groups and SQL Server In-Memory OLTP—and each has its place. But when the speed and scale extends beyond really needs to be addressed beyond database layer, the best long term approach is in-memory computing. Not only does it add speed and scale. It unlocks data, enabling 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 for 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.