EBOOK

GridGain

Best Practices for Digital Transformation with In-Memory Computing

Part 1: Building an In-Memory Computing Foundation and Roadmap

Digital transformation projects are arguably the biggest initiatives in most commercial companies and public-sector organizations today. For the last few decades IT departments spent 80 percent of their budget on average maintaining existing systems and keeping the lights on. Only 20 percent was spent on new projects. In 2019, IDC estimates that 40 percent of technology spending will go toward the digital transformation of businesses.[1]

The main driver of digital transformation is optimizing the customer experience (which improves customer retention, revenues, and profits). In every industry, there are new entrants and software giants—companies like Amazon, Apple, eBay, Expedia, Google, PayPal, and Uber—who have grabbed market share by delivering a much easier and lower-cost experience. According to Harvard Business School, companies that react and become digital leaders in their industries demonstrated 50 percent greater profitability compared to those that lagged behind.[2] The full benefit is much greater. As you'll see, improving the customer experience can double revenues and more than double profits within a 10-year period.

The process has been quite a journey for those who have already transformed their digital customer experience. It required:

- Using APIs to open up existing systems and deliver new capabilities that can be consumed by anyone.
- Moving to a development process that delivers these new capabilities in days instead of months (or years).
- Adding new technologies that ingest massive amounts of data in real-time to streamline, react to, and improve each step of the customer's experience.

Optimizing the customer's digital experience requires speed, scale, real-time intelligence, and automation. Over the last decade, existing systems have been overwhelmed by 10 to 1000 times greater transactional loads, 50 times more data, and the need to deliver streamlined, responsive customer experiences. Customer-facing processes that previously took hours are now done with as little as one click, and are sometimes completed in seconds or less. These systems were not only incapable of scaling effectively to handle the growth. They also slowed down business transformation because they took too long to change.

Companies that succeeded with their digital transformations did so in part by adopting in-memory computing (IMC) technologies such as Apache® Ignite™ or GridGain®, the commercially supported version of Ignite. IMC has successfully delivered up to 1000 times lower latency by moving data into RAM, as well as unlimited horizontal scalability on lower-cost infrastructure. It is used for low-latency, high-scale transaction processing, real-time analytics, and hybrid transactional/analytical processing (HTAP) to help automate decisions and improve business outcomes. Because in-memory computing can be implemented as a set of middleware services that slide in-between your existing database and application infrastructure, you do not have to rip out and replace your existing systems.

IN THIS SERIES

This eBook series outlines the successful journey that companies have taken, and some of the more common best practices they've followed. The first eBook explains best practices for building a foundation for digital transformation. The following eBooks detail best practices for the various types of project you'll encounter, to help guide you at each step of the journey.

Consult the eBooks in this series in sequence or in random order—after all, each digital journey is different.

BEST PRACTICES FOR BUILDING YOUR DIGITAL FOUNDATION

In this eBook, you'll learn about best practices for establishing a sound and cost-effective foundation for digital transformation:

- Foundation Best Practice 1: Start with the Customer Experience
- Foundation Best Practice 2: Build a New Real-Time Layer
- Foundation Best Practice 3: Support Real-Time Processes with Real-Time Data Management and Event Processing
- Foundation Best Practice 4: Build for Speed and Scale, and for Real-time HTAP at Scale
- Foundation Best Practice 5: Prioritize Real-Time Projects Based on a Greater Plan

1 https://www.cio.com/article/3149977/digital-transformation/digital-transformation-examples.html

2 https://www.cio.com/article/3122806/it-industry/digital-laggards-must-harness-data-or-get-left-behind.html

GridGain

FOUNDATION BEST PRACTICE 1: START WITH THE CUSTOMER EXPERIENCE

The purpose of digital transformation is to improve the customer experience. Studies, like the one from Harvard Business School mentioned earlier, show that digital leaders can be 50 percent more profitable than their competitors that are slower to adopt digital technologies. But the main reason Amazon, Apple, eBay, Expedia, Google, PayPal, and Uber disrupted their respective industries and grew by so much is not just the use of newer technologies. It's the use of newer technologies to deliver a better, more convenient, and lower-cost customer experience.

The benefits of improving the customer experience often go beyond a short-term boost in profits. A loyal customer buys more each year: nearly ten times more over their lifetime as a customer than a one-time buyer. A typical retailer loses 25 percent of their customer base annually. 20 percent of their customer base—their loyal customers—contributes 80 percent of all future revenues. A company that doubles its loyal base by taking down attrition from 25 to 20 percent, for example, can double its revenues and more than double its profits over a five-to-ten-year period.

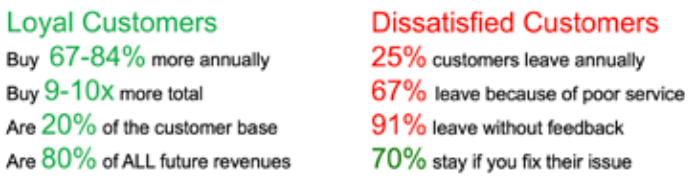

Figure 1. The Importance of Customer Loyalty

To succeed, any digital transformation must focus on improving the customer experience. At the very least, IT organizations working on digital transformation initiatives should be aligned with the executives responsible for the customer and the push to improve their experience. Ideally, the focus on the customer will underpin your digital initiative company-wide, from the CEO to every employee involved.

Define and Fill the Right Roles to Ensure a Focus on the Customer Experience

Most digital transformation initiatives center around a digital business team. At the center of the team, there is usually a Chief Digital Information Officer or IT Director. To ensure this group succeeds, it's a good idea to define a few other roles.

Chief Customer Officer

Consider a business-side executive sponsor who is responsible for improving the customer experience. Some companies have a Chief Customer Officer, while others have a Chief Digital Officer outside of IT. Whatever the name, this person needs to coordinate with IT to be sure that the right requirements and priorities are in place, and that IT has enough resources to meet its goals.

Process Architect

The process architect translates the requirements into specific process flows. This position complements the systems, data and other project-specific architects. The difference is that the process architect's mission encompasses building out different channels, applications, and APIs to make sure all the pieces fit together. This role could be supported by a business process management practitioner, for example.

API Product Managers

API product managers own the lifecycle of APIs and ensure they're designed with the API consumers in mind. You really should not expose an API to the world without having a product manager who is responsible for that API. Some companies refer to this approach as designing from the outside-in, or mapping how the consumer uses the API as an access point to the company's internal systems.

It's useful to think of each API as a product and a contract with the API consumer. In addition to creating an interface that's simple to use, the API—like any other successful product—cannot change unless it is absolutely necessary. Among other responsibilities, the API product manager helps ensure the API contract does not change.

Create an API-Centric Architecture that Enables Rapid Delivery and Customer-Friendly APIs

Existing systems are one of the biggest inhibitors of digital transformation.

One reason is that each system is a silo by design, built mostly to address a group-specific problem instead of a customer problem. Another is that systems take too long to change: often months for minor releases, and a year or more for major ones. To be viable, a digital business must evolve quickly. It must be able to deliver changes to an integrated, multi-channel, or omni-channel experience in days or weeks, not months or years.

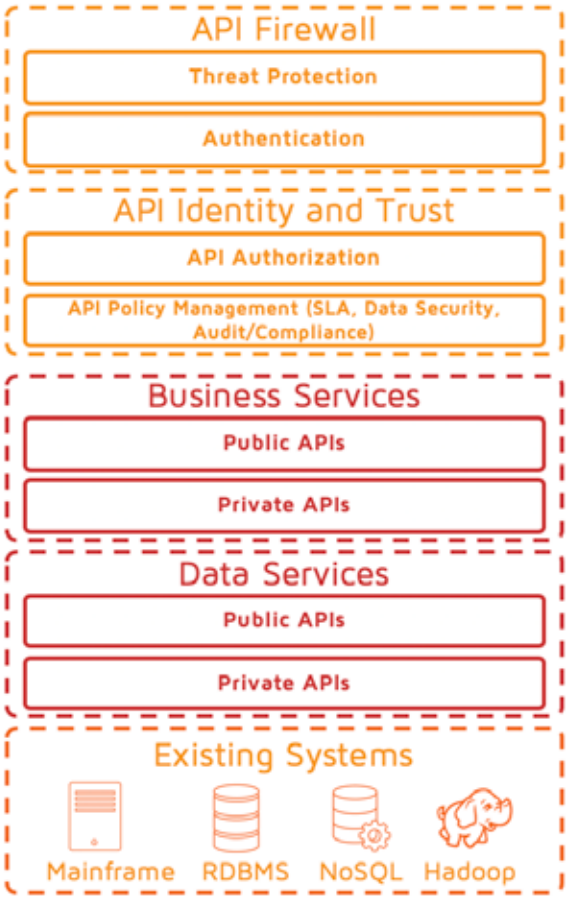

Figure 2. API-Centric Architecture

Notably, companies that can adapt this quickly use an architecture that's built from the outside-in. Several companies have a similar architecture: two layers of security on the outside, then two layers of APIs on the inside.

- The first, outermost security layer acts as an API firewall that protects against various API-level attacks, and an API authentication layer.
- The second security layer manages overall identity and trust, performing authorization for specific actions, as well as enforcing business-level API policies, including service-level agreements (SLAs), auditing, data security, or regulatory compliance.
- The business services layer provides access to top-level APIs consumed by the outside world—that is, APIs used for various interactions with customers or partners.
- The data services layer brings together related data that might be scattered across internal systems as unified APIs.

Usually, both the business and data services layers have public APIs built on top of private APIs. Public APIs are comparable to products in that they are designed to fit the needs and habits of the end user (the API's consumer) and should not change. Public APIs are often assembled from private APIs that encapsulate some specific function across systems (such as customer information).

This public/private layer decouples new API development from existing systems and allows new capabilities to be delivered in days. On a daily or weekly basis, if a new public API is needed or if logic needs to change quickly, any changes can be done by changing private APIs. If an existing system is changed or replaced, it typically impacts only a few private APIs. While that might cause a private API to be modified, and even some code within a public API, it does not cause the public API contract to change.

"WE WANT TO BE A TECH COMPANY WITH A BANKING LICENSE."

— Ralph Hamers, CEO, ING Group

Creating a Differentiated Banking Experience with In-Memory Computing

ING wanted to increase their pace of innovation, both to service changing customer needs and to move beyond traditional banking with new services and business models. Part of the challenge was that core banking services resided on a mainframe that was hard to change. They were also unable to continue to scale their infrastructure and handle increasing loads. Mobile traffic was growing 25 percent per year, and each mobile user interacted seven times more than a web user.

Using GridGain, ING added an in-memory computing layer in front of their mainframe and middleware that helped open up and share previously siloed data and functionality across channels. GridGain not only easily scaled to offload from the mainframe and handle the growth. It helped cut end-to-end latency to less than 100 milliseconds. It also helped the bank rapidly create new APIs from microservices for consumers and third parties, and be first-to-market with new services for the revised Payment Services Directive (PSD2), the Single Euro Payment Area (SEPA) initiative, and instant payments.

Watch the ING In-Memory Computing Summit presentation.

FOUNDATION BEST PRACTICE 2: BUILD A NEW REAL-TIME LAYER

Digital business happens in real-time. Unfortunately for most companies, parts of their existing business do not. Many existing processes for selling to and supporting customers take hours or days. Often, behind these processes are batch-based systems, with processes that run only hourly or nightly. This means a lot of customer data is not up to date. The question becomes: How can you provide a real-time view of a customer, or react in real-time if the data is not current?

The answer is build a new real-time layer using in-memory computing that processes real-time data and reacts in real-time. To do this, you build new applications and APIs that access a real-time version of the data within the in-memory computing layer instead of the existing systems. The in-memory computing layer handles each transaction in real-time, committing changes in memory. It then passes the transactions on to the batch-based systems, which process them on their own timeframe. The pass-through is done in such a way that it effectively allows the existing systems to think that nothing has changed. Additional integration is added to push any updates in the batch-based systems back to the real-time layer to keep both in synch. Because the in-memory computing layer has the most current version of all the data, it can now handle all reads without querying the backend systems. Offloading all reads from the existing systems helps avoid software and hardware expenditures that would otherwise be required to scale-up the backend systems.

Adding a new real-time layer is a useful strategy for more than just batch-based systems. It helps reduce latency in general. It is generally a good way to increase responsiveness and availability.

Consider that a single interaction with a customer may span many systems. The total end-to-end response time for any interaction is the sum of all the time it takes to access and combine data from all those systems. If each of ten systems delivers a 500-millisecond response time, but the requests cannot be done in parallel, the total response time adds up to over five seconds. Having all the data already in-memory eliminates this delay. With in-memory processing, these operations are executed up to 1000 times faster.

When you integrate systems, the potential uptime of the entire platform ends up being worse than the worst individual system. To illustrate, let's say you rely on ten systems, each with five-nines (99.999 percent) uptime, or five minutes a year of downtime (which is very good uptime). If you depend on all of them being up, your end-to-end reliability is four nines, or around 50 minutes of downtime a year. If one of those systems has three-nines reliability, or 525 minutes of downtime per year, you are now at 575 minutes of downtime combined. Having a single new layer that coordinates with each of the backend systems and caches the data can reduce all this downtime and deliver the reliability of a single five-nines system.

Solving a Persistent Customer Service Problem with In-Memory Computing

A leading health and fitness company had a challenge with monthly membership payments. Whenever a member paid online, the payment was submitted to a batch-based system that processed the payment overnight. As a result, it could take nearly a day to update the customer's balance. To the customer, it looked like the payment was not accepted, and they kept trying to pay. This glitch increased costs, because someone had to go in and reverse the extra payments. It also led to dissatisfied customers who would sometimes cancel their memberships.

The company used in-memory computing to fix this problem. Now whenever a payment is made, an API receives the request, updates the balance information in memory, and then passes the transaction to the back-end system. Whenever a customer checks to confirm that their payment was processed, the system looks up the account status in memory, which immediately shows the payment as received.

FOUNDATION BEST PRACTICE 3: SUPPORT REAL-TIME PROCESSES WITH REAL-TIME DATA MANAGEMENT AND EVENT PROCESSING

The real-time layer supports real-time processes. These processes have:

- Real-time data
- Real-time events
- Real-time tasks

When architects design a new real-time layer, a common trap is focusing on just one real-time aspect—for instance:

- API management, which focuses on tasks
- Messaging, which focuses on events
- In-memory computing, which focuses on data management and computing

GridGain

All these technologies are needed, as well as the best practices around them. Successful digital transformation initiatives often have architects who help design all aspects of processes, including how to react to important events. The trick is to add different tools and capabilities only when each capability is truly needed and delivers a positive return on investment.

Prioritize Real-Time Data Management

When it comes to data management, it used to be that individual systems were responsible for their own data quality. Master data management and data quality controls were mostly implemented to only help merge data for a data warehouse. Then they could build a single view of the business across disparate systems outside of the transactional applications (after the fact).

Companies usually implement master data management for operations only when data consistency becomes a major problem. In all other cases, when there are problems with the data in transactions, the transactions usually get flagged as exceptions and then fixed manually by people.

However, once you rely on real-time processes, bad data in transactions is no longer acceptable. It simply takes too long to have people manually fix exceptions. Data must be accurate and consistent from the beginning to avoid exception handling.

The best approach is to validate data at the point of entry with some combination of data quality and master data management. Several in-memory computing initiatives have started implementing real-time master data management and providing master data in memory as one of their first projects. Master data in-memory helps make sure the system can support real-time data validation. It's also a relatively low-risk, low-volume first project for in-memory computing with a fast payback for customer-facing applications.

Plan for Real-Time Event Processing and Streaming

It's important to remember that the goal of digital transformation is to improve the customer experience and to help increase customer satisfaction, retention, and spending—which in turn increases revenues and profits. The best way to monitor and improve every experience is with real-time event processing or streaming analytics.

Recall that doubling the loyal customer base can double revenues and more than double profits. Some companies achieve this by monitoring and reacting to key events around a customer's experience. For example, of those 25 percent of all retail customers who leave annually, 67 percent leave because of a poor service issue and 91 percent of those who leave give no feedback (see Figure 1). If you can identify and fix an issue, 70 percent of all dissatisfied customers stay.

In other words, if you can identify and fix all issues with customers, you can reduce attrition by over 45 percent (below 15 percent annually). The challenge for many companies is that their existing systems and processes do not make it easy to identify or respond to an issue quickly.

Streaming analytics, also known as event processing, processes streams of real-time information to help sense and respond to business events. While it is not as widely adopted as business process management or application integration technologies, event processing has been shown to help improve business outcomes for a host of use cases, including:

- Real-time customer or service monitoring to proactively sense and respond to issues
- Cross-selling or up-selling as a customer is buying other products
- Preventative maintenance by monitoring and servicing connected devices before they fail
- Real-time operational analytics where low-latency visibility is critical to making decisions fast enough

These successes make streaming analytics a critical component for digital transformation. Many of these projects also evolved into some of the first use cases for machine and deep learning. Apache Spark™ and Apache Ignite or GridGain, which companies began using together for streaming analytics in 2015, added both machine or deep learning capabilities to support the need for intelligent decision automation in these types of projects.

But for many companies, adopting streaming analytics and machine/deep learning takes time, effort, and planning. One reason is that these technologies require new sources of streaming data outside of transactional data, such as web or mobile browsing history or social media data. They also require new developer skill sets.

For that reason, it's important to identify the projects in advance that need streaming and machine/deep learning. This gives teams time to build up the skill sets and adopt the technologies. It's also important to identify and focus first on projects with a large return on investment. This helps pay for new technology infrastructure and learning that then gets reused in other projects.

Good candidates for earlier development and implementation include use cases such as customer monitoring, cross- and up-selling, preventative maintenance, or select high-value real-time analytics.

FOUNDATION BEST PRACTICE 4: BUILD FOR SPEED AND SCALE, AND FOR REAL-TIME HTAP AT SCALE

Most companies turn to in-memory computing because their digital transformation overloads their existing systems and customer channels. On average, over the last decade, the addition of new channels and APIs has created an unprecedented need for speed and scale:

- 10 to 100 times more queries and transactions as self-service and automation has increased
- 50 times more data stored and used, as companies have started to capture new types of Big Data
- 10 to 1000 times faster response times, driven by the need to streamline and automate processes, and respond in real-time to customers.

Many companies spend money on hardware and additional software licenses for each channel to increase scalability. But this usually is not the best long-term choice. In addition to being expensive, scaling vertically with hardware is limited: it cannot support 10 to 100 times greater scale and up to 1000 times greater speed (lower latency). 10-100 times growth over a few years is faster than Moore's Law.

Over time, many companies reached the same conclusion about the best long-term approach to the ever-growing need for speed and scale:

- To lower latency, add a new real-time layer in between existing systems and new projects. This moves any required data into memory as close to a customer-facing application or API as possible.
- To scale data management as data grows, scale horizontally and linearly using a "shared nothing" architecture on low-cost infrastructure rather than scaling vertically on a traditional "scale-up" architecture.
- To scale data processing as data grows, move the code over the network to the data instead of moving the data over the network to the code. Data is increasingly too big to move over the network for consumption by real-time applications

To succeed in the long run, companies decided that adding in-memory computing up front is important. First, they found it less expensive in the short term. It paid for itself because it was cheaper than buying new scale-up hardware.

Second, the need to streamline and automate processes, and to be more event-driven and responsive, led many innovators to adopt in-memory computing. Making each siloed application or channel faster was not enough. Channels needed to be integrated to deliver a seamless experience that matched what we've come to expect from the new digital players like Amazon or Uber. Each experience needed to be intelligent enough to detect an event, analyze it, and react—all in real time, during the customer interaction.

HomeAway Delivers a Seamless Personalized Experience

When HomeAway became a go-to site for vacation rentals, network bottlenecks quickly followed. To calculate daily rental rates for vacation homes in real time and for each interested traveler, HomeAway needed to run as many as 2300 batches of calculations per second. Each batch contained 200 calculations and each calculation required 250K of data. 250K multiplied by 200 calculations and then by 2300 batches is 115 GB. HomeAway needed 115 GB of data per second to run their pricing calculations at peak load times.

Moving 115 GB of data per second across a network is not feasible. HomeAway would need over 100 typical corporate networks to handle their peak loads. And even then, they might need to wait one second or more just to get the data across the network—an unacceptable delay when consumers expect total response times of one second or less.

When software architect Chris Berry of HomeAway did this math, he realized HomeAway needed an in-memory data architecture that could collocate processing with data—in other words, send the calculations to the data. The ability to collocate processing with data was one of the key reasons that HomeAway first chose Apache Ignite, the open source project GridGain is built on. With Apache Ignite, HomeAway's network bottleneck went away.

Watch the HomeAway In-Memory Computing Summit Presentation.

This was not possible in the past in part because transactions and analytics were separate. Transactional systems were not designed to support analytics. Even if they were, they did not have all the data needed to perform analytics correctly.

To get a single view of their business, companies used to extract, transform and load (ETL) data from different siloed applications and piece it back together in a single data warehouse. This meant the data would be out of date by the time it was in the data warehouse, which was good enough

at the time. As the need for real-time analytics increased, companies tried to use specialized hardware to accelerate ETL and analytics. But the expensive hardware was accelerating the wrong architecture. ETL still required the data to move across the network, which took too long. To process and respond fast enough during a transaction or interaction, the analytics need to be part of the transactional system, collocated with the operational data.

Performing analytics in-process is what Gartner calls Hybrid Transactional/Analytical Processing (HTAP). In-memory computing is widely recognized as the foundation for HTAP. It does not just add 10 to 1000 times speed and scale. In-memory computing's ability to collocate code with the data makes it possible to perform analytics and continuous learning in real-time, during a transaction or interaction. It can perform real-time analytics or machine learning—without impacting the performance of the core transactional systems—by scaling horizontally and adding more nodes to spread the load.

The early adopters of in-memory computing realized they needed to build for cost-effective speed, scale, and HTAP at scale. This ability to perform massively parallel processing (MPP) at scale in real time on a horizontally scalable low-cost architecture is what makes in-memory computing the foundation for in-process HTAP.

FOUNDATION BEST PRACTICE 5: PRIORITIZE REAL-TIME PROJECTS BASED ON A GREATER PLAN

If you use the same in-memory computing technology across applications, it not only enables you to add speed and scale to existing applications, one application at a time, as needed to keep existing business operations up and running. If you use the same platform, it also helps you succeed with your digital transformation because it:

- Opens up and combines data across existing applications for new uses, such as supporting new APIs that can be consumed by anyone
- Supports new workloads by scaling horizontally, making it easier to deliver new capabilities in days without having to change existing applications.
- Allows you to ingest massive amounts of data in real-time and combine it with existing data, during transactions and interactions, to streamline, react to, and improve each step of the customer's experience.

But a digital transformation is a journey: you cannot transform all at once. You must plan, which means choosing to tackle projects in a certain order, so that for each step of the way you've already built out the parts to make a given project succeed and to get a positive return on each investment.

The question is, which kinds of projects should you choose first, and in which order? Here are a few recommendations.

1. Review any requests for hardware and additional software licenses. Ask whether in-memory computing would make sense. Many such projects are in fact adding vertical scalability to handle increasing loads in the short term when in fact the better longer-term answer is in-memory computing.
2. Review any database purchases or re-platforming projects that rip out and replace an existing database. They're often being done to improve performance and scalability. In-memory computing may be the better answer, especially if performance and scalability issues are due to large read and data-intensive computing workloads.
3. Review all the backend sources for new API development. Determine what additional loads will be created and consider whether the use of in-memory computing or other technologies would lead to a better architecture
4. Add speed and scale to existing systems before building new APIs. Existing systems need to be able to handle the additional loads.
5. Consider grouping projects together that access the same systems. Many companies have not had sufficient control over applications from different parts of the company to enforce a common access layer and implement a single in-memory computing layer that controls all write access. This can create more work in the end trying to maintain consistency across applications. Commitment to a common layer is a lower cost and lower risk approach (if possible).
6. Order your projects by understanding your data dependencies for each project, and knowing what data needs to be in-memory for each project to succeed. If a project is dependent on other data being in-memory, either work on moving that data into memory first, or combine the projects.

NEXT STEPS: HOW TO SUCCEED WITH DIFFERENT TYPES OF PROJECTS

The focus of this eBook was on how to build the right foundation—to put in place the right people, processes, and technology. Once that foundation is in place and you've chosen a logical order for the first projects, you can start to implement those projects to help achieve the larger goal.

The next eBooks in this series are guides for each type of project you might tackle in your journey. Different project types include:

- Adding speed and scale to existing applications
- Developing new applications, channels, and APIs
- Leveraging and integrating with cloud services
- Implementing HTAP
- Building event-driven applications and streaming analytics
- Adding machine and deep learning to help automate decisions

For more information on any of these eBooks, related webinars, or customer examples, visit the In-Memory Computing Best Practices for Digital Transformation Resource Center.

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 go to 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.