WHITE PAPER

GridGain vs. Redis: The Advantages of In-Memory Computing Over Caching

A GridGain Systems In-Memory Computing White Paper

Over the past decade, companies such as Amazon, eBay, Expedia, and PayPal began delivering new and better customer experiences. In response, many companies started their own digital transformation initiatives and entered a new era of real-time, web-based customer experiences based on massive amounts of data. This new business model, built on new technologies and big data, brings a host of new opportunities. But it also presents major challenges — particularly in the areas of performance and scalability. Over the past decade, transaction and query volumes have increased tenfold, and the world's data has grown fifty times larger.

To survive and thrive in this new era, companies need IT infrastructure that will enable them to deliver an intelligent response in real time, so they can provide a better customer experience at each step. This infrastructure must do more than just access customer data and perform transactions. Today's systems also need to analyze huge volumes of related data to provide a better understanding of the customer experience and to support real-time decisions about what to do next. Gartner refers to this as hybrid transactional/analytical processing (HTAP). One way that companies achieve real-time responsiveness in HTAP applications is by keeping as much data as possible in memory, for faster access.

Distributed caching solutions such as Redis, which keep copies of data in memory, were effective for improving performance and offloading queries from databases. However, as interest in HTAP grew, many businesses found that caching products do not have sufficient support for analytics or decision automation. For one thing, these products do not support SQL, the most common analytics protocol. They are also not well suited for real-time big data analytics or computation. Any application that wants to use a distributed cache with a large amount of data needs to access it over a network, and moving big data takes time.

To meet the data needs of modern applications, many companies are turning to the next evolutionary step in caching technology: in-memory data grids (IMDGs) such as the IMDG in the GridGain® in-memory computing platform. Designed to support HTAP applications, GridGain combines support for ANSI-99 SQL and distributed ACID transactions with massively parallel processing (MPP) to enable real-time Big Data analytics and decision automation.

This white paper will take a detailed look at the challenges faced by companies that have either used Redis and run into its limitations or companies that are considering Redis and find it is insufficient for their needs. This paper will also discuss how the GridGain in-memory computing platform has helped companies overcome the limitations of Redis.

THE CHALLENGES OF USING REDIS

Redis (short for REmote DIctionary Server) was a major step forward when it was released in 2009. Redis allowed application developers to copy data from a backend database into a general-purpose cache distributed across the memory of multiple computers in a network. This use case expanded the local data capacity beyond the amount of RAM in a single server.

By providing this shared, reusable, local copy of the data, Redis was able to reduce query loads on backend databases and give applications faster data access. Because its data persisted in memory, Redis also provided a helpful way for web-application developers to locally store and manage applicationspecific data such as session state or shopping-cart details.

However, the Redis architecture has some limitations that pose major challenges for many companies:

Challenge #1: Dealing with Data Updates

Redis stores a copy of the data and often does not know when the data in the database gets updated. Developers need to write code and rules to try and keep the copy held in Redis current, which can be challenging. Redis can generate a lot of additional network traffic when getting new copies of the data. Data could still be out of date.

Challenge #2: Supporting SQL and Accelerating SQL Applications

Redis does not support SQL. Because it is a key-value store, it works reasonably well only for key-value data in custom applications. Companies who want to accelerate SQL applications with Redis have to write a lot of code to translate between SQL and the Redis key-value store.

Challenge #3: Scaling for HTAP

When an application needs to perform transactions, big data analytics, and decision automation together in real time, it needs to process a large amount of data without any delays. With Redis, that data must move across a network from Redis to the application running the analytics. Advanced calculations with big data can easily overwhelm the network with traffic and add significant delays to response times.

Challenge #4: Support for Streaming Analytics, Machine Learning, and Deep Learning

Most major new initiatives — including customer experience management, digital transformation, the Internet of Things (IoT), and process automation — involve some combination of the following capabilities: real-time stream processing, advanced analytics, and machine learning or deep learning. The lack of MPP support in Redis makes it the wrong platform for any real-time computations. IMDGs and in-memory computing platforms such as GridGain can address the limitations of caching products such as Redis and offer far higher levels of performance in a broad range of use cases.

HANDLING REDIS CHALLENGE #1

How GridGain Eliminates Update Challenges

How data flows between an application, the in-memory cache or IMDG and the backend database, and how the data in memory is kept up-to-date is at the heart of the difference between Redis and GridGain.

Redis uses a "cache-aside" data model in which the cache sits to the side of the data flow between an application and the backend database. With Redis, a developer has to write code inside their application to insert data into the cache and also write rules specifying when to refresh the data to ensure it is upto-date. (All of this is in addition to any code needed to translate between SQL and key-value format, as described in in the next section, "Handling Redis Challenge #2.") While Redis does offload the traditional reads from the database, some of those reads are replaced by the reads needed to periodically refresh the data. The more frequently the data changes or is refreshed, the higher the new load.

In contrast to the Redis "cache-aside" model, GridGain uses a read-and-write-through architecture in which all queries and transactions go directly through GridGain. Any updates or transactions first get updated in the database, and upon a successful commit to the database are automatically updated in GridGain. As a result, all of the data in the memory layer is automatically up-to-date, and not a separate copy that needs to be kept in synch. This design completely offloads reads from the backend database, reduces network traffic, and eliminates the need for extra coding to keep the data in memory in synch with the database. Because GridGain contains the operational dataset, rather than a copy, companies can use GridGain as the in-memory "system of record" for other workloads.

HANDLING REDIS CHALLENGE #2

How GridGain Supports SQL and SQL Apps

For companies with SQL-based applications, Redis is especially challenging. Redis is a key-value store so developers must write code to translate any data coming from the database into a key-value format to put into Redis. For an existing SQL-based application, a developer must add code to translate between SQL and key-values, to access Redis and to keep Redis in synch with the underlying SQL database.

In contrast, GridGain supports both key-value and SQL access with no custom coding needed to support SQL applications. GridGain slides easily between existing SQL applications and their backend relational databases. Developers can simply replace their application's JDBC or ODBC drivers with GridGain's native drivers. GridGain takes care of connecting to the backend database. GridGain creates tables based on the database schema and does a one-time load to synch with the database. GridGain then looks just like the backend database to the application, but with faster data access. It provides full support for ANSI-99 SQL, including DDL and DML statements, distributed joins and distributed ACID transactions.

GridGain's strong SQL support does not come at the expense of other types of data access. It also supports NoSQL databases, such as Apache Cassandra™ and MongoDB®. Developers can code against GridGain's unified API in a host of languages and use GridGain as both a SQL and key-value store.

This flexibility of data access ensures that companies with many types of applications and databases can take advantage of GridGain's horizontal scalability and real-time performance to build a common data access layer for all of their big-data initiatives.

HANDLING REDIS CHALLENGE #3

How GridGain Scales for HTAP

The real-time performance and scalability required for HTAP is a major challenge for Redis. Companies now need to deliver real-time, personalized experiences to customers when they are looking for products, services or support online. The transactions, analytics, computations, or machine learning processes needed to deliver these experiences require large amounts of data to be consumed in real time. Even if a pricing or personalization calculation requires only 250K of data, those 250K increments add up quickly when that calculation must be made

thousands of times per second. The result might end up being gigabytes of data per second coming from a terabytes size database. At volumes such as these, the network becomes the bottleneck. It is simply not possible to move this amount of data and deliver a real-time experience.

For real-time big data analytics and any other big data processing, GridGain's MPP capabilities collocate processing with data so the data does not need to travel over the network. GridGain distributes the code to the relevant data on each node, processes the data locally (running the processing in memory with the data), and only sends the results over the network. Types of processing that can be collocated include distributed SQL joins; any user-defined Java, .NET, or C++ code; and the built-in machine learning and deep learning algorithms of the GridGain Continuous Learning Framework.

In order for collocated processing to work optimally, data needs to be distributed such that all related data needed for processing resides on the same node. GridGain can automatically partition data based on data affinity within a datacenter, across a WAN, or across any combination of nodes running on premise or in public and private clouds. This capability enables GridGain to maximize performance based on the specific processing needs as well as the overall network topology and traffic of a company's IT infrastructure.

For high volume transactions and big data ingestion and processing, GridGain also supports native persistence. This allows GridGain to be used as a distributed SQL and key-value in-memory database in addition to its IMDG and MPP capabilities. The native persistence in GridGain is a distributed ACID- and SQL-compliant disk store for storing data and indexes on any combination of SSD, Flash, 3D XPoint, an other types of non-volatile storages. With native persistence enabled, RAM can hold 0-100% of the data and indexes, with all data kept in nonvolatile storage and accessible at any time. As an in-memory database (IMDB), GridGain has demonstrated an unmatched combination of high read and write performance, high availability, and a lower cost of ownership for HTAP applications.

While Redis can be used as a database, its persistence is mostly used to back up RAM and Flash. Redis does not support SQL or distributed ACID transactions well. It also cannot use disk to make the database larger than RAM and Flash, making it a more costly solution.

Thanks to GridGain, companies such as HomeAway® have been able to deliver HTAP applications that either weren't possible to deliver at scale, or were too costly (see "HomeAway: An HTAP Success Story").

By leveraging GridGain's MPP capabilities, these companies have nearly eliminated the need to move massive amount of data across the network for real-time big data analytics and processing. They have also scaled out more cost-effectively and reliably by dynamically adding commodity servers instead of having to scale vertically by adding increasingly larger and more expensive hardware.

HomeAway: An HTAP Success Story

When HomeAway became a go-to search site for travelers seeking vacation rentals, network bottlenecks quickly followed. To calculate daily rental rates for vacation homes in real time for interested travelers, HomeAway needed to run as many as 2,300 batches of calculations per second. Each batch contained 200 calculations, and each calculation required 250K of data. While 250K is not a lot of data, 250K multiplied by 200 and again by 2,300 is 115 gigabytes (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. A typical 10-Gigabit (10GbE, pronounced "10 gig E") network can move a maximum of 1 GB of data per second with some luck and no collisions. HomeAway would need over 100 such networks to handle their peak loads. If they turned to newer 100-Gigabit networks, which are ten times as fast, they would still need over 10 such networks. Even then, they would need to wait one second just to get the data — an unacceptable delay in today's world, where the total end-to-end response time should be less than one second.

When Software Architect Chris Berry of HomeAway did this math, he realized that distributed-caching products such as Redis could not solve HomeAway's bottleneck problem. The answer was clear. Instead of a caching product, HomeAway needed an inmemory data solution that could collocate processing with data, sending the calculations to the data so the data could stay where it was.

The ability to collocate processing with data was one of the key reasons that HomeAway chose Apache Ignite, the open source project on which GridGain is built. With Ignite, HomeAway no longer needed to expand their network capacity one-hundredfold to handle their peak loads. Their network bottleneck went away.

HANDLING REDIS CHALLENGE #4

How GridGain Supports Streaming Analytics, Machine Learning, and Deep Learning

Ultimately, the types of projects for which Redis is well suited are limited by its very specific purpose. Redis is a cache which is designed only for caching. It was not designed to automate real-time decisions, process big data, process data streams or perform machine or deep learning. For these use cases, companies need an architecture that provides collocated processing and is also optimized for ingesting and processing data streams in real time, and for continuous learning.

Streaming Analytics: GridGain's in-memory streaming support enables companies to ingest, store, process, analyze, and publish large volumes of streaming data. It allows them to do so with low latency, unlimited horizontal scalability, and high availability. GridGain is integrated and used with major streaming technologies including Apache Camel™, Apache Flink™, Apache Flume™, Apache Kafka™, Apache RocketMQ™, Apache Spark™, Apache Storm™, Java® Message Service (JMS), MQTT, Twitter® and ZeroMQ™. Clients can subscribe to continuous queries that execute and identify important events as streams are processed. GridGain has easily been able to ingest, process, and publish millions of events per second on a moderately-sized cluster.

Developers using Spark for streaming analytics also get the broadest support for Spark from GridGain compared to any other in-memory computing solution. GridGain supports Spark RDDs, DataFrames, and the Hadoop Distributed File System (HDFS), all of which are used extensively by Spark developers. GridGain also accelerates SparkSQL and other data-intensive operations using MPP. All of this support enables companies to rely on GridGain to manage both their streaming data and their data-at-rest.

GridGain also provides any necessary data to Spark in memory and in real time. This enables developers to use Spark for what it does best: processing real-time streaming data. Redis only supports Apache Spark RDDs.

Machine Learning and Deep Learning: Artificial intelligence projects involving machine learning (ML) and deep learning (DL) are on the rise, enabled by high-speed processing of big data. Performing ML and DL requires a data-management system that can perform fast computations against massive amounts of data. The GridGain Continuous Learning Framework supports MPP-style machine learning and deep learning with real-time performance against petabytes of data. GridGain's extensible MPP and streaming capabilities make it extremely well suited to support continuous learning that can lead to the best outcomes for real-time decision automation.

Download the GridGain and Redis Feature Comparison

GridGain and Redis both include functionality for caching data in memory and can partition data and be scaled out across distributed clusters. But Redis has 4 challenges: dealing with data updates; supporting SQL; scaling for HTAP; and support for streaming analytics, machine learning nd deep learning. Many of GridGain's core features distinguish it from Redis and help address these challenges, including:

Native ANSI-99 SQL Support

ACID Transactions Support

Integration with RDBMSs, NoSQL Databases, and Hadoop

Slides in-between SQL-based Applications and RDBMSs with No Custom Coding

Cross-language Support for Collocated Processing (MPP)

Comprehensive In-Memory Computing Platform for HTAP

Apache Spark Support for DataFrames, RDDs, and HDFS

Support for Machine Learning and Deep Learning

Built on Apache Ignite, a leading open source project

Download an in-depth feature comparison that shows how GridGain and Redis compare across 25 categories including: use cases, 3rd party database support, native persistence, distributed SQL and queries, memory architecture and optimization, distributed architecture, data rebalancing, in-memory streaming and integration with Apache Spark, security and audit, configuration and grid management, supported platforms and built-in integrations, cloud and virtualization support.

https://www.gridgain.com/resources/product-comparisons/redis-comparison

GridGain

GRIDGAIN VS. REDIS: THE ADVANTAGES OF IN-MEMORY COMPUTING OVER CACHING

End users expect personalized, real-time responsiveness in their interactions with companies. Distributed-caching products such as Redis help with the individual performance bottlenecks of existing applications and their databases. But Redis cannot help with new initiatives where real-time analytics or continuous machine learning or deep learning are needed to make recommendations or automate decisions. Redis cannot support these requirements for HTAP, so it cannot help improve the customer experience the way the GridGain in-memory computing platform can. Feature by feature, the comparison is clear: GridGain provides the next-generation in-memory architecture, the real-time performance and horizontal scalability, and the required capabilities for HTAP that Redis lacks. With built-in support for SQL, ACID transactions, MPP, streaming, machine learning, deep learning, and much more, GridGain empowers companies to deliver real-time transactions and analytics across all of their high-volume HTAP applications.

Contact GridGain Systems

To learn more about the GridGain in-memory computing platform, please email our sales team at sales@gridgain.com or call us at +1 (650) 241-2281 (US) or +44 (0) 208 610 0666 (Europe).

About GridGain Systems

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