In-Memory Computing: A Key Enabler of Operational Intelligence

By Dr. William Bain, Founder & CEO and Chris Villinger, VP Business Development & Marketing

In-memory computing technology enables OI by providing a scalable, highly available platform that simultaneously tracks and analyzes events from multiple data sources and then generates immediate feedback to steer behavior.

HIGHLIGHTS

- Today's digitally interconnected applications require immediate business insights — "operational intelligence" — to enable actionable business decisions that drive results.
- In-memory data grids have evolved to include advanced, in-memory computing features.
- In-memory computing provides the key enabling technology for operational intelligence.

TABLE OF CONTENTS

ScaleOut Software
In-Memory Computing for Operational Intelligence

www.scaleoutsoftware.com

OPERATIONAL INTELLIGENCE FOR IMMEDIATE BUSINESS INSIGHTS

Today's competitive world requires businesses to make lightning fast decisions. Quick responses to changing market conditions are imperative to maintain a competitive edge. Beyond commerce, industries such as healthcare and transportation safety increasingly rely on advanced informatics capabilities to stay a step ahead. With actionable, real-time insights, physicians can provide immediate diagnostics to reduce patient recovery times. Hospitals can improve patient care at lower costs. Transportation departments can reduce traffic congestion, avoid hazardous collisions, and in extreme situations, save lives.

A companion whitepaper, "Operational Intelligence: The Next Generation of Business Intelligence" introduces the concept of operational intelligence (OI) and discusses application scenarios across several industry verticals, including e-commerce, retail, financial services, manufacturing, Internet of Things (IoT), cable television and healthcare. While business intelligence provides strategic guidance from the data warehouse, operational intelligence enables live systems to react to second-by-second changes, providing immediate recommendations and alerts. This enables live systems to capture perishable opportunities before the moment is lost. By instantly tracking and analyzing fast-changing data and then generating feedback to steer behavior, operational intelligence extracts value from these systems that otherwise would be missed.

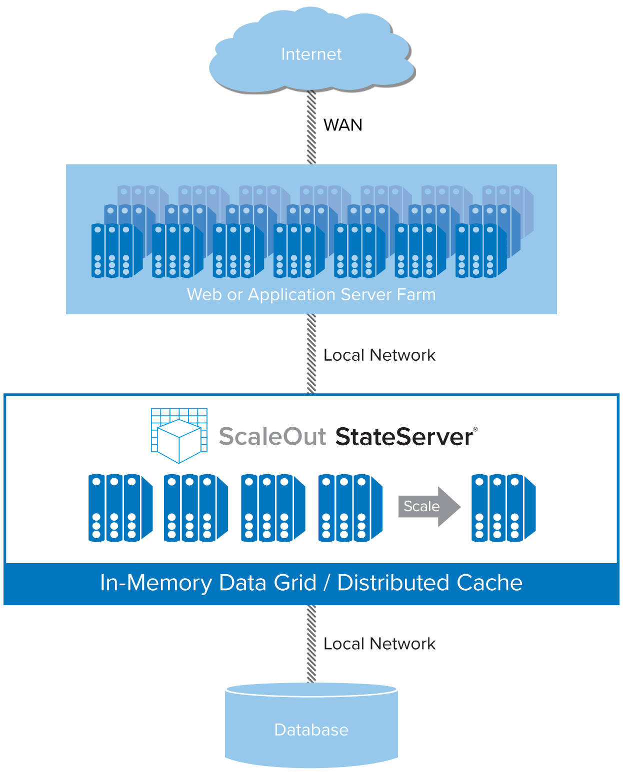
Whether suggesting a new product, rebalancing a hedge fund, monitoring a patient, or looking for an impending machine failure, operational intelligence adds exciting new value that can generate significant ROI.

In-memory computing technology enables OI by providing a scalable, highly available platform that simultaneously tracks and analyzes events from multiple data sources and then generates immediate feedback to steer behavior. By hosting business logic that delivers real-time insights, this technology extends traditional business intelligence beyond the data warehouse to realize the full potential of OI for today's real-time applications. As a result, OI enables businesses to extract valuable information from live systems that otherwise would be missed.

This paper provides an in-depth, technical explanation on how in-memory data grids (IMDGs) have evolved to deliver a sophisticated foundation for broader in-memory computing capabilities, which in turn are the key enablers of operational intelligence.

IN-MEMORY DATA GRIDS SCALE APPLICATION PERFORMANCE

An in-memory data grid (IMDG) is a distributed software architecture that stores data in memory (RAM) across a cluster of commodity servers. (IMDGs should not be confused with in-memory databases, which are designed to accelerate traditional databases using memory-based storage.) While database servers typically provide a relational view of the data, IMDGs offer an object-oriented view of data using an in-memory, key-value store, also known as a "NoSQL" store, of serialized objects. These objects are generated and managed by object-oriented business logic, and they typically are organized into collections by type. IMDGs complement traditional databases by providing a fast, scalable, highly-available, and uniformly accessible data storage layer for use by business logic within distributed applications.



An example of an IMDG deployed as a distributed cache for a web farm

The fundamental benefits of using in-memory data grids for data storage include:

- extremely fast data access (low latency) for distributed applications using RAM-based storage, which is orders of magnitude faster than disk,
- the ability to seamlessly add or remove servers for automatic scalability as data workloads increase (or shrink),
- transparent high-availability to handle failures without data loss by storing data redundantly across multiple servers,
- tight integration with business logic using object-oriented techniques and straightforward APIs that access data in multiple programming languages, including C#, C/C++, Java and REST,
- parallel query that runs across all grid servers to quickly search for stored data based on its properties, and
- integrated, WAN-based data replication across multiple data centers to enable strategies for disaster recovery.

When used for data storage, IMDGs help to scale the performance of mission-critical, distributed business applications by hosting large data sets in RAM and making fast-changing data quickly accessible to applications. IMDGs often are used for distributed caching to offload databases and other disk-based storage, for example, in storing session state or shopping carts within large web farms.

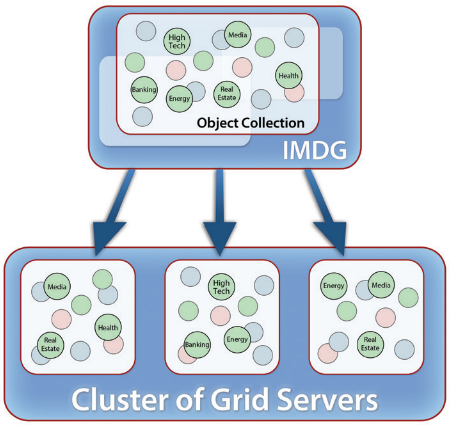
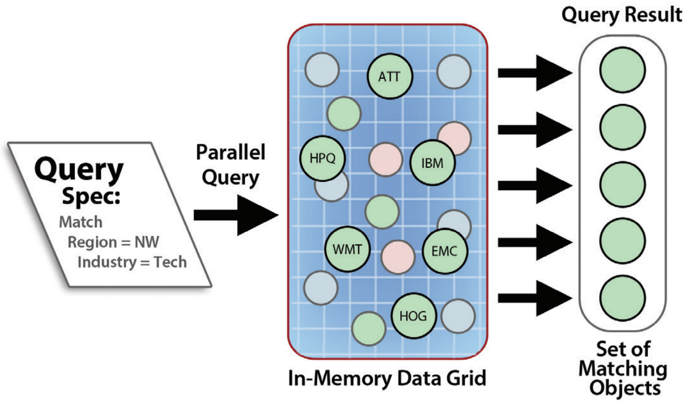


Illustration of an IMDG's mapping of an object collection (logical view) to a collection of grid servers (physical view) for in-memory storage

Architecting an IMDG to transparently manage its key mechanisms, including cluster membership, data partitioning, load-balancing, replication, heartbeating, recovery and self-healing, lets users benefit from simplified application development, fast deployment, and minimum total cost of ownership (TCO). Effective IMDG products are engineered to maximize ease of use as a fundamental design philosophy. This is accomplished by automating the grid's inner workings to the greatest extent possible.

THE NEXT STEP: INTEGRATING IN-MEMORY COMPUTING INTO IMDGS

IMDGs typically offer a distributed, data-parallel query mechanism that can quickly scan a large data set for objects whose properties match a query specification. This provides an important tool for identifying groups of related data to be reviewed or analyzed. The following diagram illustrates the use of parallel query for selecting stock history data:



An example of parallel query within an IMDG

With ever growing data set sizes and increasingly complex analyses, the use of parallel query can create performance bottlenecks by requiring large amounts of data to be retrieved simultaneously. Moreover, this approach does not take advantage of the scalable computing power inherent in the design of an IMDG, which runs on a cluster of computers. However, moving computation into the grid minimizes data in motion to eliminate these bottlenecks and enables IMDGs to serve as fast, data-parallel compute engines.

In many ways, IMDGs are next generation parallel supercomputers. Like a supercomputer, an IMDG can employ data-parallel programming to execute a single computation on a collection of objects that have been distributed across the servers in the IMDG's cluster. This code automatically runs in parallel across all servers to deliver fast results and scalable performance.

Luckily, today's IMDGs do not require a PhD in parallel supercomputing to deploy and use in production. In fact, this technology is readily and easily accessible to the enterprise developer. By leveraging the IMDG's object-oriented data model to simplify the design of data-parallel applications, grid-based APIs for data-parallel computation make it easy to take advantage of the cluster's computing power. This makes development straightforward and fast.

To implement a data-parallel computation, a developer simply writes a single method which the IMDG executes in parallel on a large collection of objects held within the grid. Compared to task-parallel techniques (such as Apache Storm), which involve several tasks, explicit communication and coordination, data-parallel programs are significantly easier to develop.

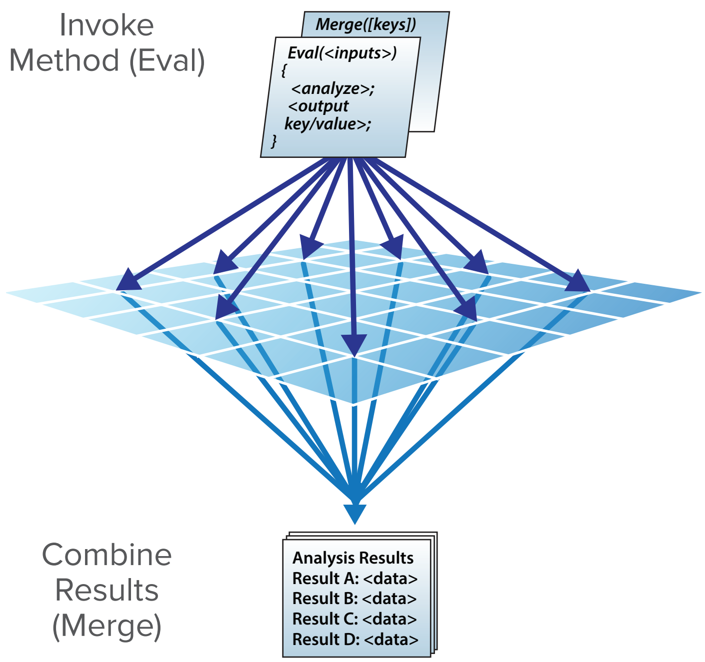
Beyond creating a fast, scalable repository to store fast-changing data, an IMDG's server cluster provides an ideal platform to execute data-parallel computations on data stored in the grid. Its CPU resources can run analytics code on in-memory data stored within each server. This capability takes advantage of data partitioning in the grid to shorten execution times, and avoid the overhead of data motion over the network. This dramatically reduces compute times and ensures scalable performance. It also leverages the IMDG's scalability; adding servers handles growing workloads without increasing compute times.

Object-oriented computing

Because IMDGs store data as objects created by business logic, an integrated compute engine follows an object-oriented model to structure data-parallel computations. Called “parallel method invocation” (PMI), this feature lets developers specify an analysis (or “eval”) method that the compute engine executes in parallel on a collection of objects stored within the IMDG. This method analyzes an object and possibly updates it using a domain-specific algorithm.

> ▶▶ *Moving computation into the grid minimizes data in motion to eliminate network bottlenecks and enables IMDGs to serve as fast, data-parallel compute engines.*

The developer also can specify a second (“merge”) method to combine results created by the parallel computation so that they can be collected for return to the requesting application. The compute engine executes the merge method in parallel on each grid server and then globally combines the results to produce a single object for return to the client application. Together, PMI’s eval and merge methods represent an object-oriented formulation of well understood techniques from parallel supercomputing. The two methods also provide a convenient basis for cleanly separating domain-specific algorithms from the specifics of the data-parallel execution platform.



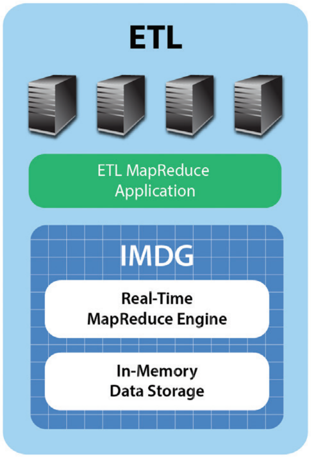
An example of parallel method invocation

Automatic code shipping and parallel execution

Sophisticated IMDGs automatically ship application code to all grid servers for execution and start execution environments (e.g., JVMs or .NET runtimes) on these servers to host and execute this code. To minimize startup times and reduce execution latency, the developer also can persist an execution environment (called an “invocation grid”) across several parallel method invocations.

Hadoop MapReduce on live data

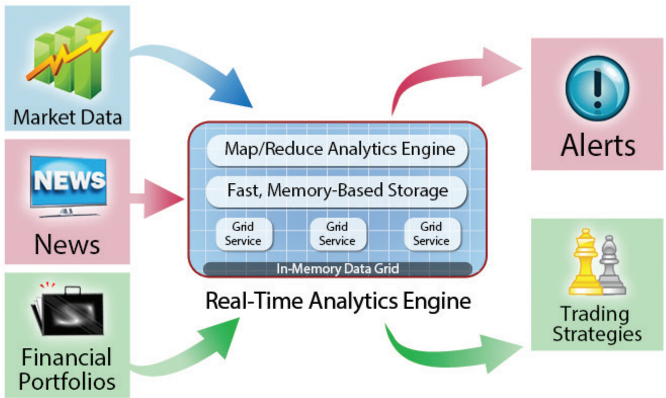
IMDG’s which incorporate a parallel computation engine, also can execute standard Hadoop MapReduce applications (i.e., applications which are fully code-compatible with Apache Hadoop), allowing these applications to access in-memory data from the grid and output to HDFS or vice versa. This enables integration with the data warehouse in myriad new ways. For example, an extract-transform-load (ETL) function can be deployed as a conventional MapReduce application within the IMDG. The application can read incoming live, streaming data from the grid's memory, transform the data as required for storage in the data warehouse, and then output the data to HDFS using standard MapReduce techniques, as illustrated in this diagram:



In-memory MapReduce for ETL

ENABLING OPERATIONAL INTELLIGENCE WITH IN-MEMORY COMPUTING

Recent advancements in in-memory data grids have paved the way for operational intelligence. With operational intelligence, companies can now analyze in real time the fast-changing data they manage within their operational systems, and enrich the data with historical information to identify patterns and trends that require immediate action. While business intelligence provides strategic guidance from the data warehouse, operational intelligence acts on the front lines, helping to add value and enhance competitiveness on a second-by-second basis.



Using an IMDG to implement real-time analytics for operational intelligence

In-memory computing eliminates the bottlenecks inherent in the techniques used for business intelligence. For example, it avoids the overheads of disk-based data storage and batch scheduling, enabling the immediate, efficient tracking and analysis of live data. Moreover, in-memory data grids incorporate high availability techniques that ensure uninterrupted processing vital to live, mission-critical environments.

> ▶▶ *With operational intelligence, companies can now analyze in real time the fast-changing data they manage within their operational systems.*

Operational intelligence requires that fast-changing data be ingested, correlated, and analyzed quickly to provide immediate feedback to a live system. The low-latency, scalability, and high availability of in-memory data grids combines with in-memory computing to enable operational intelligence for mission-critical enterprise applications.

Low latency for speed

The most important requirement for operational intelligence is low latency for fast results. In-memory computing delivers low latency by using fast, in-memory data storage and integrated, data-parallel computing. This architecture tracks and analyzes fast-changing data generated by live systems in seconds (or less), generating immediate feedback to

  | www.scaleoutsoftware.com

identify patterns, steer behavior, and capture business opportunities.

Scalability for low latency with large workloads

Scalability and low latency work hand in hand. Scalability keeps latency low as the workload grows, enabling fast results in all situations. An in-memory computing architecture which combines data-parallel computing techniques and automatic load-balancing delivers seamless scalability. By simply adding (or removing) servers, developers can maximize throughput and ensure low latency.

High availability for fault-tolerance

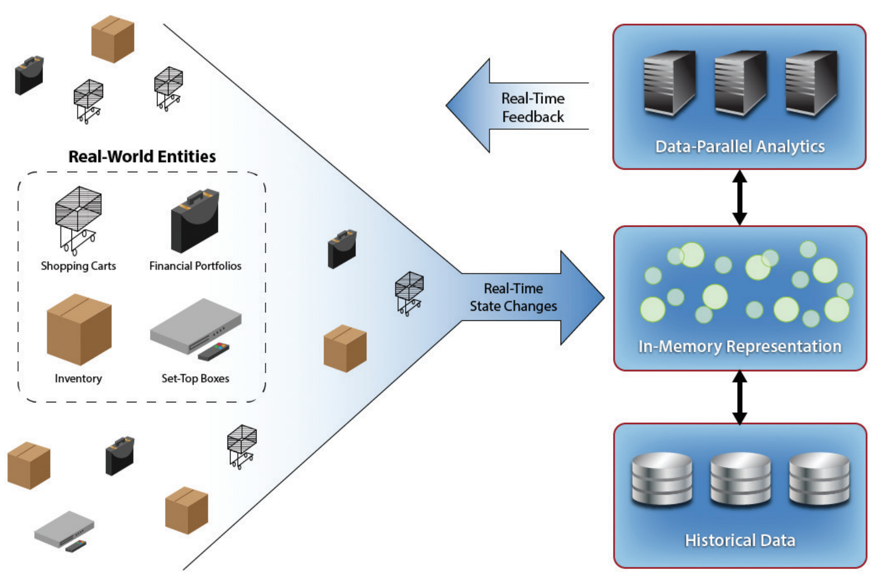
What sets operational intelligence apart is its integration into live, production systems — mission-critical systems that require 24/7 availability. Since operational systems cannot afford to lose mission-critical data, they require an in-memory computing platform that ensures data is never lost and computations always complete — even if a server fails.

Object-oriented, in-memory data storage lets developers create a linearly scalable, highly available model of real-world entities (such as financial portfolios, e-commerce shoppers, or real-world devices in the Internet of Things), which easily can be updated as changes occur. In conjunction with a data-parallel compute engine that enables fast analysis to detect emerging patterns and generate immediate feedback, these technologies make it straightforward to build an in-memory representation of real-world entities.

Unlike pure streaming and event-processing systems, operational intelligence uses in-memory computing to maintain a comprehensive model of an entire live system. This approach leads to a deeper introspection that goes beyond understanding what has happened to focus on understanding what is happening right now, and predict what will happen next. Scalable, real-time analysis of data-in-motion in conjunction with data-at-rest is what makes operational intelligence possible.

A FOCUS ON IN-MEMORY MODELS THAT TRACK LIVE SYSTEMS

The combined capabilities of distributed, in-memory computing allow mission-critical applications to tap into the power of live data and combine it with related, historical information. In order to model and steer the behavior of live systems which produce event streams in real time, users need a low-latency platform which stores, updates, and analyzes fast-changing data to maintain and enhance this model and then provide feedback within milliseconds.



Example of using in-memory computing to track and model a live system for real-time feedback

IN SUMMARY

Time-to-insight can make or break businesses trying to survive in an increasingly digital world. Operational intelligence delivers immediate insights on perishable business opportunities before the moment is lost. Integrating in-memory computing into in-memory data grids has brought many of the benefits of parallel super-computing to the enterprise, and has given rise to a new generation of advanced in-memory computing capabilities that enable operational intelligence.

ABOUT THE AUTHORS

Dr. William L. Bain is founder & CEO of ScaleOut Software, Inc., and holds a Ph.D. in electrical engineering from Rice University. Bill has contributed to advancements in parallel computing at Bell Labs research, Intel and Microsoft, and holds several patents in computer architecture and distributed computing. Bill founded and ran three start-up companies prior to founding ScaleOut Software in 2003. His previous company, Valence Research, developed and distributed Web load-balancing software, and was acquired by Microsoft Corporation. As a member of the screening committee for the Seattle-based Alliance of Angels, Bill is actively involved in entrepreneurship and the angel community. Bill is also a regular speaker at conferences around the world.

Chris Villinger is VP of Business Development & Marketing at ScaleOut Software, Inc. Chris has two decades of experience at global, high-tech multinationals, including Microsoft, Philips and Accenture, where he drove software marketing, marketing technology and content management, led global digital marketing and commercial sales networks, and systems integration consulting. He holds a B.S. in electrical engineering from Tulane University and a Bilingual International MBA from IESE Business School in Barcelona, Spain.

ABOUT SCALEOUT SOFTWARE

ScaleOut Software develops leading-edge software that delivers scalable, highly-available in-memory computing technology to a wide range of industries. ScaleOut Software's in-memory computing platform enables operational intelligence by storing, updating, and analyzing fast-changing data, so that businesses can capture perishable opportunities, before the moment is lost.

ScaleOut Software offers a full portfolio of in-memory computing products and technologies to deliver operational intelligence for your organization. Please visit www.scaleoutsoftware.com for more details on our technology and product portfolio for mission critical enterprise applications, including ScaleOut StateServer®, our award-winning, linearly scalable, highly available core in-memory data grid, and ScaleOut ComputeServer®, our compute engine for blazingly fast, data-parallel computation on live, fast-changing data. Lastly, ScaleOut hServer® offers the world's first in-memory execution platform that delivers real-time results on in-memory data without any changes to your MapReduce code.

ScaleOut Software was named a "Cool Vendor" in Gartner's "Cool Vendors in In-Memory Computing Technologies, 2015."

Gartner 2015 CoolVendor

ScaleOut Software