

# A Superior Alternative to In-Memory Databases and Key-Value Stores

Technology Brief



---

## Introduction

Businesses often use in-memory databases or key-value stores (caching layers) when applications require extremely high performance. However, in-memory databases incur a high total cost of ownership and have hard scalability limits, incurring reliability problems and restart delays when memory limits are exceeded. In-memory key-value stores share these limitations and introduce architectural complexity and network latency as well.

This paper explains why InterSystems IRIS™ data platform is a superior alternative to in-memory databases and key-value stores for high-performance SQL and NoSQL applications.

## Taking Performance and Efficiency to the Next Level

InterSystems IRIS is the only persistent database that can match or beat the performance of in-memory databases and caching layers for concurrent data ingestion and analytics processing. It can process incoming transactions, persist the data to disk, and index it for analytics in under one microsecond on commercially available hardware without introducing network latency.

**The superior ingest performance of InterSystems IRIS results in part from its multi-dimensional data engine, which allows efficient and compact storage in a rich data structure. Using an efficient, multi-dimensional data model with sparse storage techniques instead of two-dimensional tables, random data access and updates are accomplished with very high performance, fewer resources and less disk capacity. It also provides in-memory, in-process APIs in addition to traditional TCP/IP access APIs to optimize ingest performance.**

InterSystems has developed a unique technology, Enterprise Cache Protocol (ECP), that further optimizes performance and efficiency. It coordinates the flow of data across a multi-server environment, from ingestion through consumption. It enables full access to all the data in the environment – via SQL, C++/C#, Java, Python, Node.js, and other common languages – without replicating or broadcasting data across the network.

ECP lets the servers in a distributed system function as both application and data servers. Data and compute resources can be scaled independently based on workload type (i.e., transaction processing or analytic queries) and can dynamically access remote databases as if they were local. Only a small percentage of the system's servers need to hold primary ownership of the data. If analytic requirements increase, application servers can be added instantly. Likewise, if disk throughput becomes a bottleneck, more data servers can be added. The data is repartitioned, while applications retain an unchanging logical view.

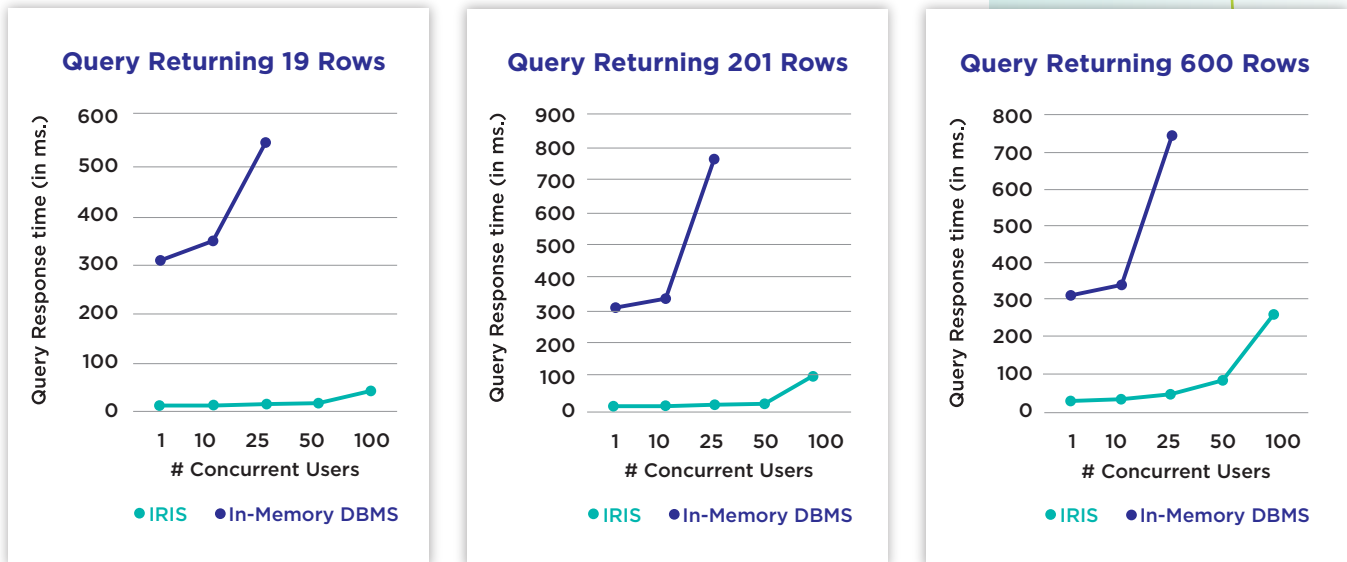
Each node in the distributed system can operate on data that resides in its own disk system or on data transferred to it from another data server. When a client requests data, the application server will try to satisfy the request from its local cache. If the data is not local, the application server will request it from the remote data server; the data is then cached on the local application server and is available to all applications running on that server. ECP automatically manages cache consistency and coherency across the network.

As a result, InterSystems IRIS enables complex analytic queries on very large data sets without replicating data. This includes the ability to perform joins that can access data distributed on disparate nodes or shards, with extremely high performance and no broadcasting of data.

Using ECP is transparent and requires no application changes or special techniques. Applications simply treat the entire database as if it were local.


In competitive tests run at a leading global investment bank using its data and queries, InterSystems IRIS consistently outperformed a leading commercial in-memory database, analyzing almost 10 times the data (320 GB vs. 33 GB) using less hardware (four virtual machines, eight cores, and 96 GB RAM vs. eight virtual machines, 16 cores, and 256 GB RAM).

### InterSystems IRIS Delivers Lower Overall Response Time


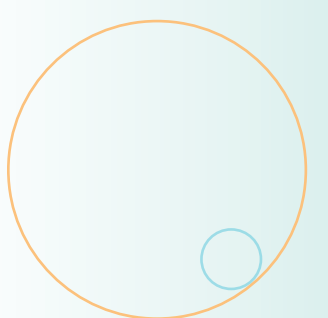


**Figure 1: IRIS outperforms leading in-memory DBMS querying 10 times the data.**

Note that the absence of a data point in the graphs indicates that the query failed to return a result.



**WITH IN-MEMORY  
DATABASES, WHEN  
THE AVAILABLE  
MEMORY IS EXHAUSTED,  
PROCESSING STOPS.**



---

## Raising Reliability Through a Permanent Data Store

Embedded within InterSystems IRIS is a permanent data store, and it is always current. InterSystems IRIS automatically maintains a current representation of all data on disk in a format optimized for rapid random access.

By contrast, in-memory databases have no permanent data store. As a result, all of the data must fit in the available memory, with enough memory available to ingest new data and process analytic workloads. The available memory can be exhausted due to unexpected increases in data volume or query volume (or both). Queries — especially large analytic queries — consume memory during execution, and to produce the results. When the available memory is exhausted, processing stops.

For mission-critical applications, such as trading applications in financial services firms, dropped or delayed transactions and service outages can be catastrophic. With in-memory databases, the contents of memory are periodically written to checkpoint files, and subsequent data is stored in write-ahead log (WAL) files. Rebuilding the in-process state after an outage, which requires ingesting and processing the checkpoint file and the WAL files, can take hours to complete before the database is back online.

With InterSystems IRIS, recovery is immediate. Thanks to its persistent database, data is not lost when a server is turned off or crashes. The application simply accesses the data from another server or from disk and continues processing, eliminating the need for any database recovery or rebuilding of database state.

## Boosting Scalability Through Intelligent Buffering

Because InterSystems IRIS does not have the hard scalability limits of in-memory databases, it is not constrained by the total amount of available memory. It uses intelligent buffer management to keep the most frequently used data in memory while rapidly accessing less-frequently used data from disk on demand and frees memory as needed by purging the data that is less frequently accessed. By contrast, an in-memory database must maintain all data in working memory, including data that may never be accessed again.

With InterSystems IRIS, if a piece of data on a one-machine system is not in the cache, it is simply retrieved from disk. In a distributed environment, if data is not in the local cache, an InterSystems IRIS-based application will automatically try to retrieve it from the cache of the data node that owns it. If the data is not in cache there, it is retrieved from disk. If the available memory is completely consumed, intelligent buffering purges the least recently used data to clear memory for new data or processing tasks.

**Since it is not memory-limited, an InterSystems IRIS-based system can handle unplanned spikes in ingest rates and analytic workloads and can scale to handle petabytes of data. In-memory databases cannot.**

## Reducing Total Cost of Ownership

Since memory is more expensive than disk, operating InterSystems IRIS-based applications results in reduced hardware costs and lower total cost of ownership compared with in-memory approaches. Many in-memory systems keep redundant copies of data on separate machines to safeguard against the effects of having a computer crash, further increasing costs.

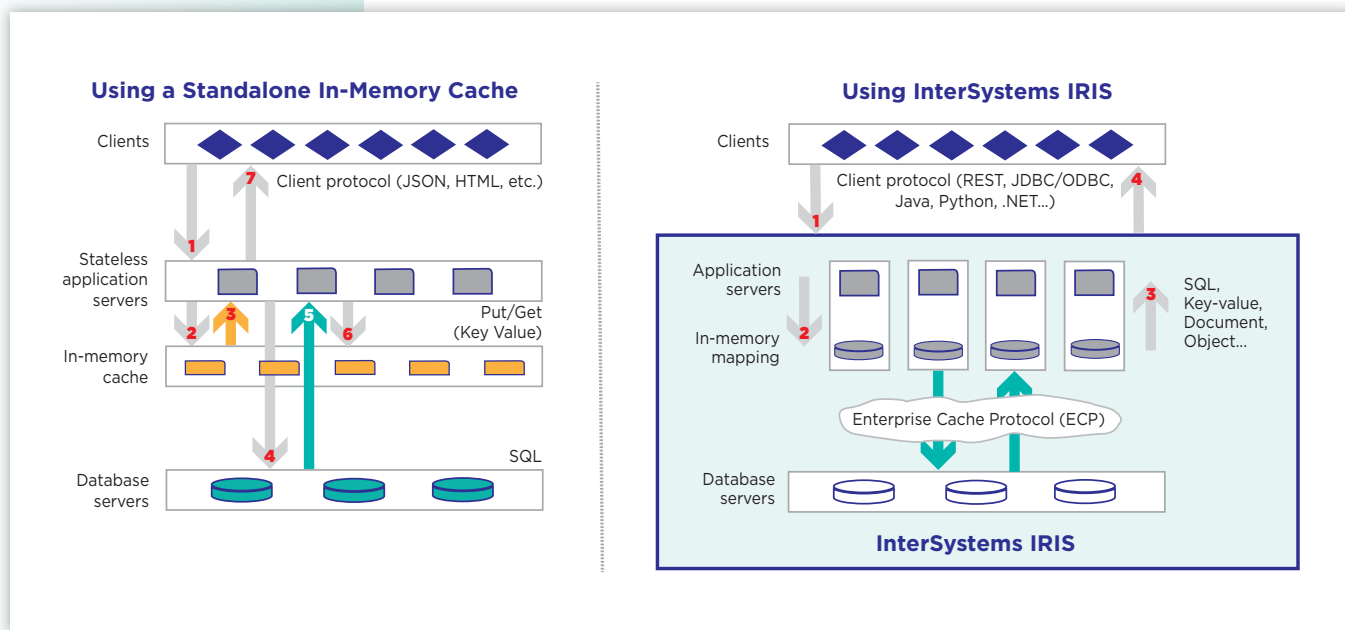
## In-Memory Key-Value Stores

Some organizations handle high-performance applications by operating an in-memory key-value store as a standalone caching layer between the storage engine and the application server. However, this approach is rapidly losing appeal for several reasons.

**Architectural complexity.** The application must manage redundant representations of the data at the various layers, as well as the integration and synchronization with the cache and the database. For example, the application code might first perform a lookup to determine whether the required data is in the caching layer. If it is not, the application will perform a SQL query to access the data from the database, execute the application logic, write the result to the caching layer, and synchronize it with the database (See Figure 2).

**Increased CPU costs.** There is an inherent mismatch between the caching layer (which work with strings and lists) and the application code. Therefore, the application must continually convert the data between the structures in the cache and the application layer, increasing CPU costs as well as developer effort and complexity.

**Latency.** Since requests between the application server and the caching layer are made over the network, this approach increases network traffic and introduces additional latency into the application.



**Figure 2: Comparing InterSystems IRIS with a separate in-memory caching layer, application server, and data store.**

In fact, in a recent research paper, engineers from Google and Stanford University argued that *“the time of the [remote, in-memory key-value store] has come and gone: their domain-independent APIs (e.g., PUT/GET) push complexity back to the application, leading to extra (un)marshaling overheads and network hops.”*<sup>1</sup>

InterSystems IRIS provides superior performance and efficiency compared with these remote caching layers while reducing architectural and application complexity.

## Conclusion

The primary reason for using in-memory databases and caching layers is performance. But despite their speed, they all have limitations, including hard scalability limits, reliability problems and restart delays when memory limits are exceeded, increased architectural and application complexity, and high total cost of ownership.

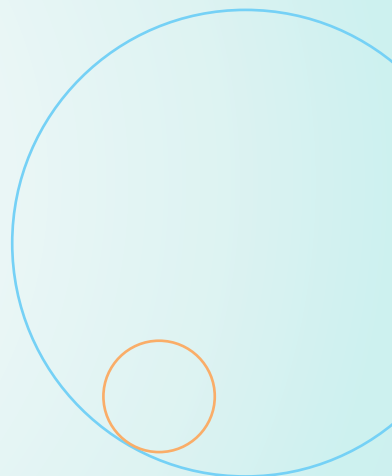
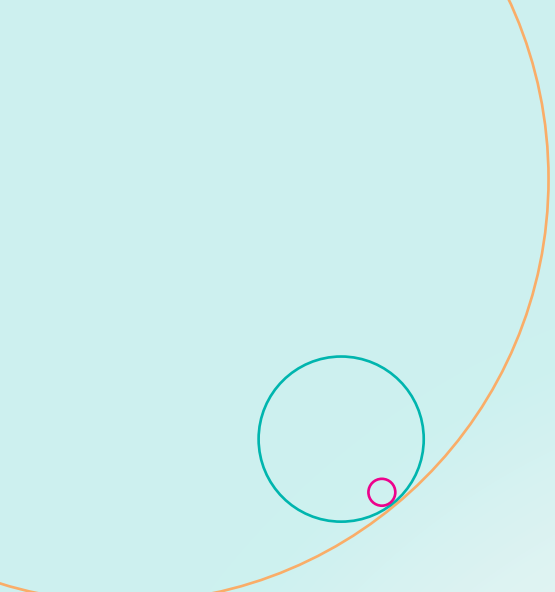
InterSystems IRIS is the only persistent database that provides performance equal to or better than that of in-memory databases and caches without any of their limitations. All of this makes InterSystems IRIS a superior alternative for mission-critical high-performance applications.

## About InterSystems

InterSystems is the information engine that powers some of the world’s most important applications. In healthcare, finance, government, and other sectors where lives and livelihoods are at stake, InterSystems has been a strategic technology provider since 1978. InterSystems is a privately held company headquartered in Cambridge, Massachusetts (USA), with offices worldwide, and its software products are used daily by millions of people in more than 80 countries.

**For more information, visit [InterSystems.com/IRIS](https://www.intersystems.com/IRIS).**

<sup>1</sup> Adya, Atul, Robert Grandl, Daniel Myers, and Henry Qin. “Fast Key-value Stores: An Idea Whose Time Has Come and Gone.” May 13, 2019. <https://doi.org/10.1145/3317550.3321434>.



The power behind what matters.

