InterSystems IRIS Basics: Scaling with Distributed Caching

Version 2024.1
2024-07-02

InterSystems Corporation 1 Memorial Drive Cambridge MA 02142 www.intersystems.com

InterSystems IRIS Basics: Scaling with Distributed Caching
InterSystems IRIS Data Platform Version 2024.1 2024-07-02

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Response Center (WRC)
Tel: +1-617-621-0700
Tel: +44 (0) 844 854 2917
Email: support@InterSystems.com

Table of Contents

InterSystems IRIS Basics: Scaling with Distributed Caching

This article explains how InterSystems IRIS® data platform can scale for user volume by using application servers for distributed caching, leveraging the Enterprise Cache Protocol (ECP).

For an online hands-on exercise that will take you through the process of creating and testing a simple distributed cache cluster, see Creating a Distributed Cache Cluster

1 The Problem: Scaling for User Volume

When users connect to your InterSystems IRIS databases via applications, they need quick and efficient access to the data. Whether your enterprise is small, large, or in between, a high number of concurrent user requests against the databases — *user volume* — can cause performance problems on the system that hosts the databases. This can potentially affect many more users, making them wait longer to receive the information they need. And in a dynamic business, user volume can grow rapidly, further impacting performance.

In particular, if a lot of users are executing many different queries, the size of those queries can outgrow the cache, meaning that they can no longer be stored in memory and instead need to read data from the disk. This inefficient process causes bottlenecks and performance problems. You can increase the memory and cache size on the system (vertical scaling), but that solution can be expensive, inflexible, and ultimately limited by the maximum capabilities of the hardware. Spreading the user workload over multiple systems (horizontal scaling) is a more flexible, efficient, and scalable solution.

2 The Solution: Distributed Caching

To improve the speed and efficiency of users' access to the data, InterSystems IRIS can use distributed caching. This technology allows InterSystems IRIS to store the database cache on multiple *application servers*. User volume can then be distributed across those servers, thus increasing the cache efficiency. The internode communication that makes this possible is enabled by ECP, the Enterprise Cache Protocol.

Using distributed caching, you can enable users who make similar queries to share a portion of the cache, which is hosted on an application server clustered with the data server where your data is hosted. The actual data remains on the data server, but caches are maintained on the application servers for faster user access. The data server takes care of keeping the cached data up-to-date on each application server in the enterprise.

With a distributed cache cluster, you can easily scale your solution by adding or removing application servers as needed. All application servers automatically maintain their own connections to the data server, and attempt to recover the connection if it drops.

You can configure application servers and their associated data servers on the individual cluster instances using their Management Portals, or deploy and configure a cluster using InterSystems Cloud Manager (ICM). For more information on ICM, see *InterSystems IRIS Demo:ICM* and the *InterSystems Cloud Manager Guide*.

Important: As of InterSystems IRIS release 2023.2, ICM is deprecated. It will be removed in future releases.

3 How Does Distributed Caching Work?

When you deploy an InterSystems IRIS distributed cache cluster, you designate one instance as the data server, and one or more instances as application servers. The instances do not need to run on the same operating system or hardware; they only need to conform to the InterSystems IRIS system requirements.

- The data server performs like a standard InterSystems IRIS server, hosting databases in namespaces and providing the data to other systems upon request.
- The application servers receive data requests from applications. When a user opens an application, instead of connecting to the data server, it connects to an application server. The user won't notice anything different. The application server fetches the necessary data from the data server and provides it to the user.
- The application server stores the data in its own cache, so that the next time any user requests the same data, the application server can provide it without needing to contact the data server again.
- The data server monitors all of the application servers to make sure that the data in their caches is up-to-date. The data server also handles data locks for the whole system.
- If the connection between an application server and the data server is lost, the application server automatically attempts to reconnect and recover any needed data.
- You can design your applications to direct users who make similar queries to the same application server. That way, the users can share a cache that includes the data they need the most. For example, in a healthcare setting, you might have clinicians running a particular set of queries and front-desk staff running different queries, using the same application and the same underlying data; those sets of users can be grouped together on separate application servers. As another example, if the cluster handles multiple applications, each application's users can be directed to their own application server(s) for maximum cache efficiency.

4 Learn More About Distributed Caching and ECP

To learn more about using distributed caching and ECP with InterSystems IRIS, see the following resources:

- Creating a Distributed Cache Cluster (online hands-on exercise)
- "Horizontally Scaling Systems for User Volume with InterSystems Distributed Caching" chapter of the *Scalability Guide*
- Sample Mirroring Architecture and Network Configurations, Redirecting Application Connections Following Failover or Disaster Recovery, Configuring Application Server Connections to a Mirror, and other distributed caching and ECP-related sections in the "Mirroring" chapter of the *High Availability Guide*