Article
Benjamin De Boe · Sep 19, 2017 4m read

Horizontal Scalability with InterSystems IRIS

Last week, we announced the **InterSystems IRIS Data Platform**, our new and comprehensive platform for all your data endeavours, whether transactional, analytics or both. We've included many of the features our customers know and loved from Caché and Ensemble, but in this article we'll shed a little more light on one of the new capabilities of the platform: **SQL Sharding**, a powerful new feature in our scalability story.

Should you have exactly 4 minutes and 41 seconds, take a look at this neat video on scalability. If you can't find your headphones and don't trust our soothing voiceover will please your co-workers, just read on!

Scaling up and out

Whether it's processing millions of stock trades a day or treating tens of thousands of patients a day, a data platform supporting those businesses should be able to cope with those large scales transparently. *Transparently* means that developers and business users shouldn't worry about those numbers and can concentrate on their core business and applications, with the platform taking care of the scale aspect.

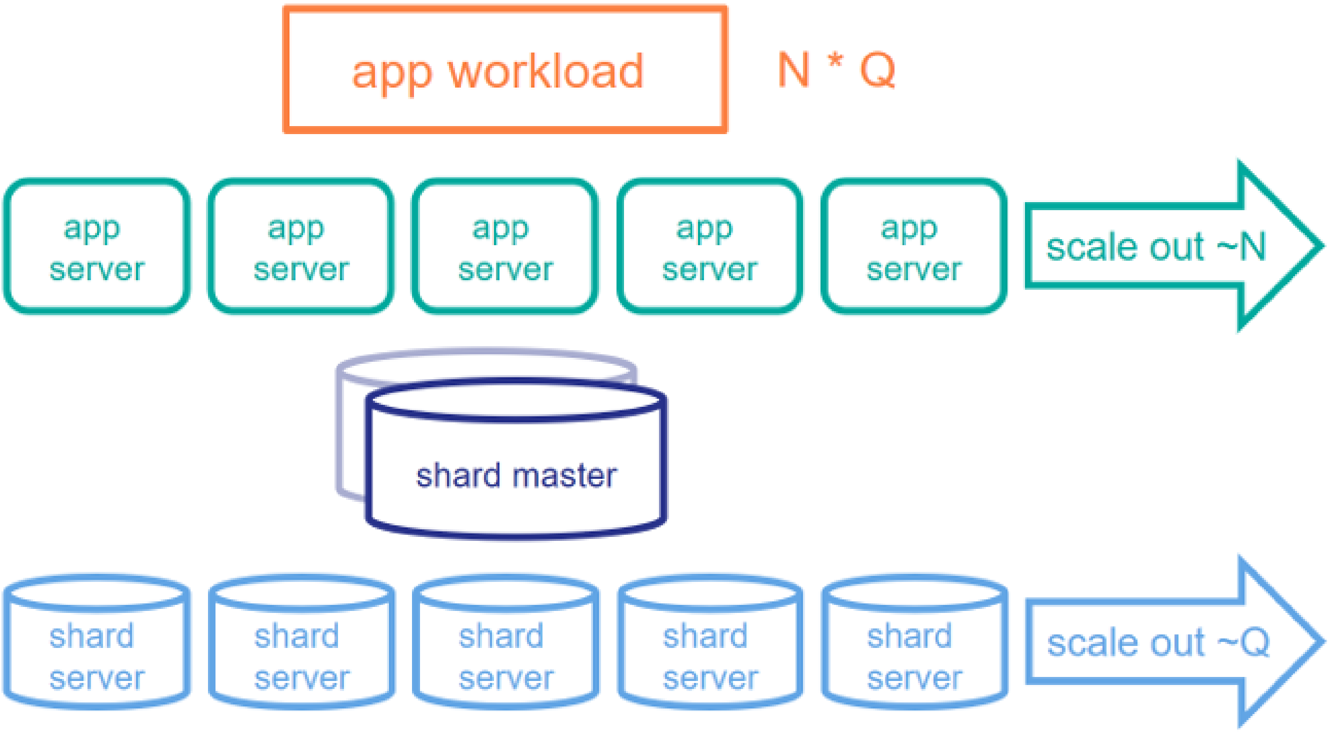
For years, Caché has supported **vertical scalability**, where advancements in hardware are taken advantage of transparently by the software, efficiently leveraging very high core counts and vast amounts of RAM. This is called scaling *up*, and while a good upfront sizing effort can get you a perfectly balanced system, there's an inherent limit to what you can achieve on a single system in a cost-effective way.

In comes **horizontal scalability**, where the workload is spread over a number of separate servers working in a cluster, rather than a single one. Caché has supported ECP Application Servers as a means to *scale out* for a while already, but InterSystems IRIS now also adds SQL sharding.

What's new?

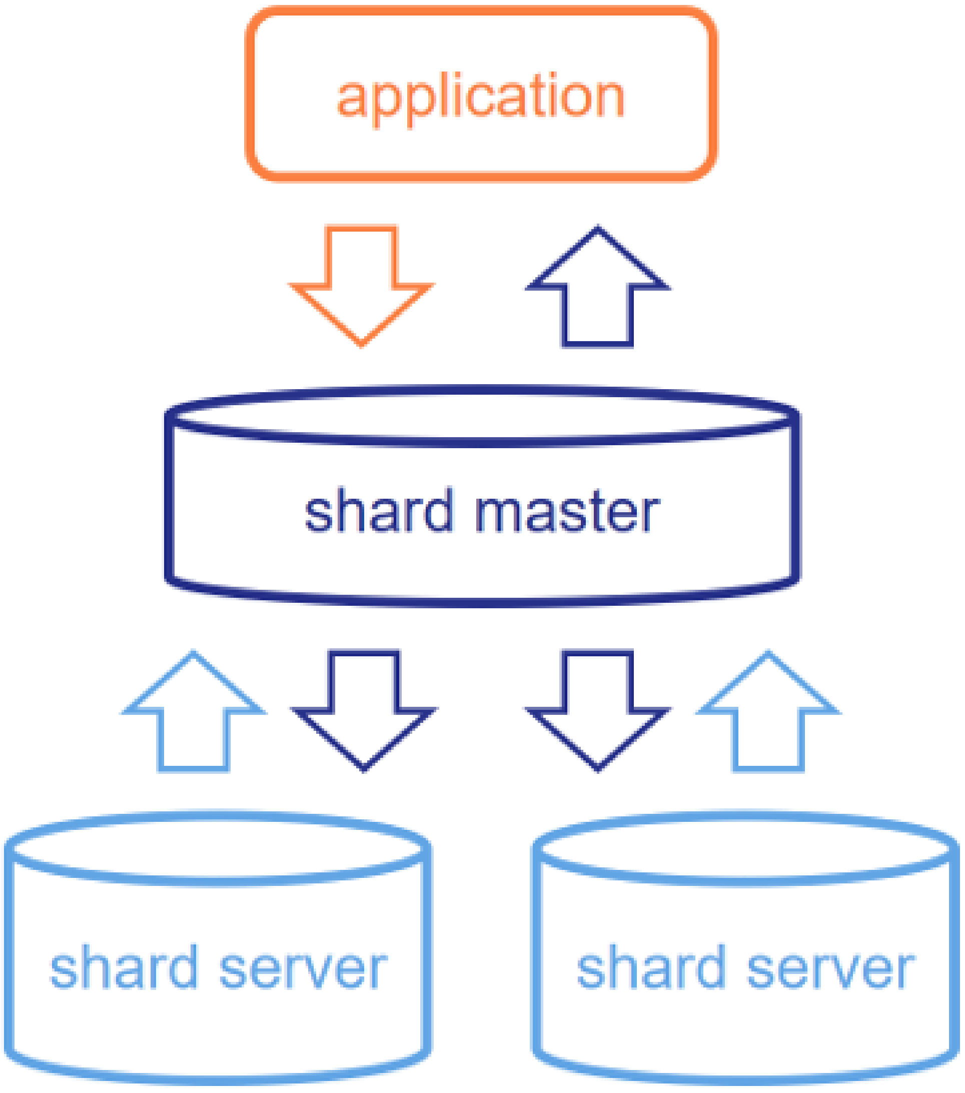
So what's the difference between ECP Application Servers and the new sharding capability? In order to understand how they differ, let's take a closer look at workloads. A workload may consist of tens of thousands of small devices continuously writing small batches of data to the database, or just a handful of analysts issuing analytical queries each spanning GBs of data at a time. Which one has the largest scale? Hard to tell, just like it's hard to say whether a fishing rod or a beer keg is largest. Workloads have more than one dimension and therefore scaling to support them needs a little more subtlety too.

In a rough simplification, let's consider the following components in an application workload: N represents the *user workload* and Q the *query size*. In our earlier examples, the first workload has a high N but low Q and the latter low N but high Q. ECP Application Servers are very good at helping support a large N, as they allow partitioning the application users across different servers. However, it doesn't necessarily help as much if the dataset gets very large and the working set doesn't fit in a single machine's memory. Sharding addresses large Q, allowing you to partition the dataset across servers, with work also being pushed down to those *shard servers* as much as possible.



SQL Sharding

So what does sharding really do? It's a SQL capability that will split the data in a *sharded table* into disjoint sets of rows that are stored on the shard servers. When connecting to the *shard master*, you'll still see this table as if it were a single table that contains all the data, but queries against it are split into shard-local queries that are sent to all shard servers. There, the shard servers calculate the results based on the data they have stored locally and send their results back to the shard master. The shard master aggregates these results, performs any relevant combination logic and returns the results back to the application.



While this system is trivial for a simple SELECT * FROM table, there's a lot of smart logic under the hood that ensures that you can use (almost) any SQL query and a maximum amount of work gets pushed to the shards to maximize parallelism. The *shard key*, which defines which rows go where, is where you anticipate typical query patterns. Most importantly, if you can ensure that tables often JOINed together are sharded along the same keys, the JOINs can be fully resolved at the shard level, giving you the high performance you're looking for.

Of course this is only a teaser and there is much more to explore, but the essence is what's pictured above: SQL sharding is a new recipe in the book of highly scalable dishes you can cook up with InterSystems IRIS. It's complementary to ECP Application Servers and focuses on challenging dataset sizes, making it a good fit for many analytical use cases. Like ECP app servers, it's entirely transparent to the application has a few more creative architectural variations for very specific scenarios.

Where can I learn more?

Recordings from the following Global Summit 2017 sessions on the topic are available on http://learning.intersystems.com:

- What's Lurking in Your Data Lake, a technical overview of scalability & sharding in particular
- We Want More! Solving Scalability, an overview of relevant use cases demanding for a highly scalable platform

See also this resource guide on InterSystems IRIS on learning.intersystems.com for more on the other capabilities of the new platform. If you'd like to give sharding a try on your particular use case, check out http://www.intersystems.com/iris and fill out the form at the bottom to apply for our early adopter program, or watch out for the field test version due later this year.

#Artificial Intelligence (AI) #Analytics #Distributed Data Management #ECP #Machine Learning (ML) #Sharding #SQL #InterSystems IRIS

Source URL:https://community.intersystems.com/post/horizontal-scalability-intersystems-iris