

Demystifying Rocket MultiValue Connection Pools

A short adventure with and without Connection Pools



Marie

Chief Technology Officer
Major retailer, with both brick-and-mortar stores and a global online presence

She's responsible for ensuring customers can easily place online orders, especially during spikes, like Black Friday. And she needs to report back to her executive team, who are focused on not only protecting but also enhancing the organization's revenue stream.

Marie has asked two of her trusted database administrators (DBAs), Elise and Jon, to prepare proposals on the best strategy for handling both expected and unexpected spikes in demand.



Elise, who is more familiar with web applications and is used to working with APIs that offer access to a wide range of business services and applications, recently read a [blog post](#) where she learned all about Connection Pools and their performance benefits, including the ability to scale up and down with demand. She believes that since it's tough to accurately predict the volume of requests and the MultiValue application needs to dynamically scale up and down, that Connection Pools are the answer from both a cost and performance perspective.



Jon, a database expert who knows the ins and outs of accessing data, is proposing a solution that requires direct access to files and MultiValue (MV) logic using database connections and lower-level libraries. He feels this gives him more control but also knows it will require him to write code to optimize access and deal with the unexpected spikes in demand.

To accommodate more access, Jon's proposal requires the organization to purchase more user licenses for their MultiValue database and to write a custom app to handle Marie's requirements.

Whose proposal will Marie pick?



Elise's proposal centers on creating RESTful APIs to access the MV data and business logic needed to power the organization's e-commerce solution using a new, free tool — the Rocket MultiValue Integration Server (MVIS). MVIS is a piece of middleware that sits between an application and the database server and provides connection management, monitoring, and administration services. MVIS relies on Connection Pools to configure the application, so it performs well regardless of demand loads. Her proposal highlights the fact that the organization won't have to write any code to perform low level functions like opening and closing database connections or logic to scale up and down, since it's all handled by MVIS' Connection Pools.

In her proposal she makes other valuable points that Connection Pools offer:

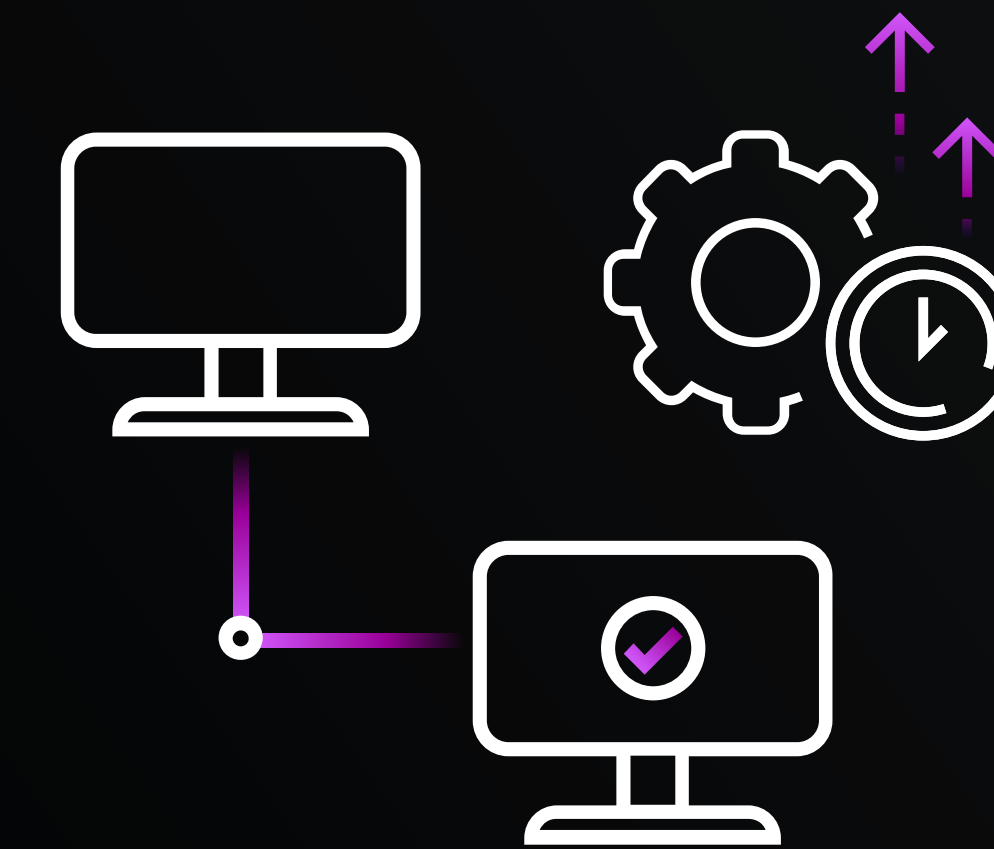
- Greater throughput for the system, since time spent opening and closing connections are removed from the equation.
- Lower individual response time for clients — **since Connection Pools re-use a connection, there's no need to wait for a new one.**
- Reduced overhead of connection creation on your data server — **letting your data server work on what matters.**
- Reduced overhead of connection creation on your clients — **letting your middle tier work on what matters.**
- Control over the number of resources dedicated to your application — **letting your data server work on what matters.**
- Protection from unexpected dangerous spikes — **limiting the number of resources dedicated to your application.**

Connection Pools are the scalability component for this organization's connectivity infrastructure.

So why did Marie implement Elise's proposal?

- 01 Overall, **Jon's** proposal takes longer to build and is more expensive and time consuming to maintain.
- 02 **Elise's** solution relies on RESTful APIs that are easier to use by the web developers writing the front-end for the e-commerce site.
- 03 Using Connection Pools provides a solution for the initial requirements that can also be configured to deal with future changes in demand.

Most of the time a traditional database connection is just waiting for activity. Connection Pools are ready for the next connection, accomplishing a lot of work in quick succession.



Connection Pools and Rocket MVIS

The Rocket MultiValue Integration Server (MVIS), a free tool for Rocket® UniVerse and Rocket® UniData® customers on maintenance, helps you extend business opportunities, expand your workforce with younger developers, and get products to market faster. MVIS allows you to easily expose both your MV data and business logic as RESTful APIs. Plus, its cloud-ready deployment options help you minimize total cost of ownership for modern MV applications.

Please note, MVIS requires UniVerse 11.3.1 or later, or UniData 8.2.1 or later running on a local or remote machine that is licensed for connection pooling. **If you have an Enterprise Edition license, did you know that you have two Connection Pools?**

Not sure how many Connection Pools you might need? We rely on scientific calculations and best practices to help you figure it out.

The future won't wait—modernize today.

Visit RocketSoftware.com

Talk to an expert

