



[Insider Series] Announcing the MongoDB CWAL Rust driver project



WE STILL HAVE SOMETHING IN STORE TO SATISFY THE RUST DEVELOPERS AMONG YOU

The Black Friday deals are over and 2019 is coming to an end, but we still have something in store to satisfy the Rust developers among you. Today, we are announcing [MongoDB CWAL](#), a community-supported MongoDB Rust driver for those who can't wait any longer (CWAL). The project is also available on [crates.io as mongodb_cwal](#).

If you ever played the game Starcraft, you may have heard of the "operation cwal" cheat code that makes everything build much faster. In the same way, MongoDB CWAL is like a Rust cheat code to get a production-ready Rust driver without having to wait for the official one to come out.

Our Rust driver is a fork of the [unsupported Rust driver prototype developed](#) by MongoDB labs. The irony of this prototype project is that while it is not officially supported, it remains one of the best Rust drivers available. In other words, if you need MongoDB support in Rust, chances are you'd be using it, which is exactly what we did.

It worked fine initially, and we thought it would get improved over time, except it didn't. Even worse, when [inquiring about the original Rust driver project status](#), we learned that it would not be maintained because all efforts are being put on a new official Rust driver. This means that all of us who need to use MongoDB can either wait, or keep using the current driver that won't be updated at all, even if community members submit patches. This prompted us to fork the project in order to make MongoDB a viable option for Rust.

To put things in context, about a year ago, we had about 1000 Wayk Den concurrent connections. We are now peaking at 6,500 connections. With an increased load on our servers, small problems can quickly become critical issues that need to be addressed quickly.

About nine months ago, we were pulling our hair trying to figure out why MongoDB crashed after reaching over 4000 (!) database connections, which was definitely not normal. We found out that the connection pool was leaking all database connections, so the only outcome possible was the one we were experiencing.

[We issued a fix](#), and the number of database connections has been within the acceptable range since then. However, our fix was never merged, and others had the unfortunate experience of [discovering the same problem months later](#).

Fast forward a month ago, where we had to investigate and fix much deeper issues with the driver. The first one was a slow but steady memory leak of 2 MB/hour (or 336 MB/week), causing our microservices to eventually run out of memory and restart. The second one was the lack of proper replica set support, which prevented us from upgrading our database to one that would scale beyond just a single instance.

We were in dire need of a solution, and one thing we didn't have is time. It just had to work, and it had to work now. All of our resources were put to solving the problem, and while it was a team effort, we were truly lucky to have one of our talented interns save our bacon. Difficult situations can oftentimes shine the light on raw talent - in this case, Benoit. He tries to remain modest, but he still fixed both issues in a matter of days, without prior knowledge of MongoDB.

If you are a Rust developer and use MongoDB, please switch to [MongoDB CWAL](#) today. We will be glad to accept community contributions and work with other people who need good quality MongoDB support in their applications.

Happy coding!