

Evolutionary history of the RabbitMQ .NET Client towards concurrency

Goals

target

Familiarity with the RabbitMQ client

Understand how concurrency is hard with the existing client approach

Experience the process of evolving the client

Process

Understand the synchronous path

Optimize the synchronous path

Introduce a redundant asynchronous path

Process

Cleanup the mess created

Rip out the custom code and replace it with `System.Threading.Channel`

First class support for concurrency within the client

It's code
time

Recap

reminder

Find the **I/O-bound** paths in your code

Get rid of **lock contentions**

If possible use a **redundant** code path
that is **async all the way**

Recap

reminder

Gradually extend IO-bound paths

Never block threads

Only custom offload where really necessary

Recap

reminder

On .NET 5 consider **ValueTask** to benefit from pooling

Leverage existing battle tested **reliable** BCL components where you can

If you can, **contribute back to OSS** that you use in your company

Thanks

Slides, Links...

github.com/danielmarbach/AsyncRabbitMQ

Q & A

daniel.marbach@particular.net

[@danielmarbach](#)
particular.net/blog
planetgeek.ch

Thanks