

---

Question

[Uri Shmueli](#) · Jul 13, 2017

## .Net Gateway Parallel Calls – Optimal Number?

More calls in parallel to a single .Net Gateway can yield better performance, but too many could cause problems. What is the "Right" number? What do you do/recommend?

We have been using the .Net Gateway for quite a while now.

We are planning now to make even more use of it, functionality-wise, but more importantly, and the cause for my question, more heavily in the sense of concurrency.

Lately we built a mechanism that generates calls to the .Net Gateway sequentially, one at a time, but because of the planned multitude of calls, we want to move to calls in parallel, as we've seen this boosts performance.

On the other hand we've also experienced the fact that if we have "too many" processes calling in parallel then the Gateway "chokes" (that's the reason we built the "queue" mechanism to start with).

Note we also startup several .Net Gateways in parallel. Each servicing a different kind of activity. This of course also helps the parallelization and performance. But the question here concerns several Caché processes calling a single .Net Gateway at the same time.

So we'd like to have an optimum number of processes calling one Gateway in parallel.

Is there such a number? Are there some guidelines or rules of the thumb about the .Net Gateway parallelization?

Can you share your experience? What did you find worked best for you?

We're interested in hearing both what InterSystems folks have to say about this (Development, Support, SEs), and also our fellow developers out there.

Thanks

[#.NET](#) [#Caché](#)

---

Source URL: <https://community.intersystems.com/post/net-gateway-parallel-calls-%E2%80%93-optimal-number>