

Forum:[Hack Forums](#)

Thread: [WebClient Memory Leak \[C#\]](#)

draxyee - September 5, 2019, 11:25 am

Hi,

I'm using HttpClient on C# to make requests, however, in my Parallel.For loop, the RAM on the process keeps increasing until it hits a System.OutOfMemoryException. I have researched this and it is to do with the nature of HttpClient. ([stackoverflow . com/questions/27732546/httpclienthandler-httpclient-memory-leak](#))

I have then tried initiating an array of HttpClient objects, and then reusing them, but once you've made a request, you cannot change the proxy of the HttpClient (which is what I'm trying to achieve), otherwise it will throw this error:

Quote:HttpClient This instance has already started one or more requests. Properties can only be modified before sending the first request

TL;DR

- 1) If I generate a new object in each Parallel iteration it will have a memory leak.
- 2) If I hard code an array of httpclient objects it wont allow me to modify them after usage.

Does anyone know an approach I can take here?

Thanks a lot,
Draxyee

Cybertic - September 7, 2019, 1:12 pm

Hey,

well, it depends on the code you are using and what are exactly doing, it's a little bit hard to know the problem without the code.

I am not 100% sure about that but try to use Task.WhenAll

draxyee - September 8, 2019, 3:39 am

(Sep 7, 2019 01:12 PM)Cybertic Wrote: [\[CITE\]](#) Hey,

well, it depends on the code you are using and what are exactly doing, it's a little bit hard to know the problem without the code.

I am not 100% sure about that but try to use Task.WhenAll

It doesn't really matter what threading/task method you use, WebClient leaks when it's a short lived object, and I can't think of any solution atm.

0xDEAD10CC - September 8, 2019, 5:42 pm

What is wrong with creating an object pool for a collection of re-usable clients that get handed leased out to each of the threads used then? Use a semaphore or something and control number of concurrent requests if needed. That, or don't use the HttpClient issue if profiling determines that it isn't usable for your scenario. The solution(s) is/are fairly obvious.

draxyee - September 9, 2019, 3:57 pm

(Sep 8, 2019 05:42 PM)0xDEAD10CC Wrote: [\[CITE\]](#) What is wrong with creating an object pool for a collection of re-usable clients that get handed leased out to each of the threads used then? Use a semaphore or something and control number of concurrent requests if needed.

I can't reuse HttpClient due to the error I showed at the start of the thread. I need to change the proxy when I reuse the object, and after making a request with a client you can't change its properties.

Also, I can't change it from HttpClient to something else, its a library :(

0xDEAD10CC - September 9, 2019, 5:22 pm

(Sep 9, 2019 03:57 PM)draxyee Wrote: [\[CITE\]](#) I can't reuse HttpClient due to the error I showed at the start of the thread. I need to change the proxy when I reuse the object, and after making a request with a client you can't change its properties.

Also, I can't change it from HttpClient to something else, its a library :(

How many proxies are you using here and how many concurrent connections? You aren't gaining anything by just spinning up 100+ threads with all different proxies since this would exhaust the network stack anyways. You can create an HttpClient pooling mechanism to re-use the same HttpClient for a single proxy though, rather than re-creating a new instance to use the same proxy.

draxyee - September 10, 2019, 5:20 am

(Sep 9, 2019 05:22 PM)0xDEAD10CC Wrote: [\[CITE\]](#) How many proxies are you using here and how many concurrent connections?

Using hundreds of thousands of proxies & thousands of threads at the same time. We could set 100 but it wouldn't be efficient for us.

(Sep 9, 2019 05:22 PM)0xDEAD10CC Wrote: [\[CITE\]](#) You can create an HttpClient pooling mechanism to re-use the same HttpClient for a single proxy though, rather than re-creating a new instance to use the same proxy.

Yes, but we need to change proxy with each iteration of the parallel.for, which means I can't reuse the same HttpClient.

0xDEAD10CC - September 11, 2019, 7:10 pm

(Sep 10, 2019 05:20 AM)draxyee Wrote: [\[CITE\]](#) Using hundreds of thousands of proxies & thousands of threads at the same time. We could set 100 but it wouldn't be efficient for us.

You don't understand what you're doing then, and that statement is clear... You're not going to gain efficiency just by spawning new threads, and if you think that's true then you don't understand the hardware.

(Sep 10, 2019 05:20 AM)draxyee Wrote: [\[CITE\]](#) Yes, but we need to change proxy with each iteration of the parallel.for, which means I can't reuse the same HttpClient.

What is the purpose of the tool you're trying to make?

gigajew - September 11, 2019, 7:56 pm

Stackoverflow Wrote: That's definitely something you need to look out for. However, even if you could set a different proxy per request, the underlying network stack would still need to open a socket for each proxy, so you wouldn't be gaining anything over an HttpClient instance per proxy in terms of the socket exhaustion problem.

<https://stackoverflow.com/questions/49818605/dynamically-change-proxy-in-httpclient-without-hard-cpu-usage>

0xDEAD10CC - September 11, 2019, 7:59 pm

(Sep 11, 2019 07:56 PM)gigajew Wrote: [\[CITE\]](#)

<https://stackoverflow.com/questions/49818605/dynamically-change-proxy-in-httpclient-without-hard-cpu-usage>

Yeah, why did you link this?

"But what is true is that you can't set a different property per request in a thread-safe way, and that's unfortunate."

^ He is using tons of threads

riu41776 - September 12, 2019, 5:15 am

You're probably creating shitload of clients without releasing them from memory after usage.

Note that classes such as HttpClient implement IDisposable which are meant to be used using the "using" statement.

Also, HttpClient is thread-safe, while WebClient isn't.