

2019-01-05 · 約 866 字 · 預計閱讀 2 分鐘

## HttpClient 無法反應 DNS 異動的解決方式

之前筆記 [探討 HttpClient 可能的問題](#) 提到使用 HttpClient 時避免 socket 耗盡的方式就是只建立一個 HttpClient instance (透過 static or singleton) · 但這樣的方式卻會造成 DNS 紀錄出現變動被忽略進而影響系統正確運行

筆記中有提到可以透過將 HttpClient 的 `DefaultRequestHeaders.ConnectionClose` 屬性設定為 `true` · 也就是將 HTTP 的 keep-alive header 設為 `false` · 讓 socket 在每次處理完 request 即關閉 · 這幾天查資料時發現還有其他做法可以使用 · 一併紀錄一下

### 前提設定

1. 使用 staticHttpClient instance 來取得 <https://blog.yowko.com/> 資料

C#

```
1 public class StaticHttpClientService
2 {
3     private static readonly HttpClient _httpClient;
4     static StaticHttpClientService()
5     {
6         var baseUri = new Uri("http://blog.yowko.com");
7         _httpClient = new HttpClient();
8         _httpClient.BaseAddress = baseUri;
9     }
10
11     public HttpClient HttpClientInstance = _httpClient;
12 }
```

2. 透過修改 hosts 檔案來模擬 DNS 修改

修改方式請參考 [在 Windows 環境將特定網址指向不同 IP](#)

3. 使用環境

- Visual Studio 2017 15.9.4
- .NET Framework 4.7.2
- BenchmarkDotNet 0.11.3

為避免外部網路干擾 · 透過修改 hosts file 將 [blog.yowko.com](https://blog.yowko.com/) 指向 127.0.0.1



## 解決方式

### 1. dispose HttpClient

每次 request 結束即 dispose 的做法存在與 TCP 協定及 OS 層實作的效能問題，應重複使用 HttpClient 並自行管理 dispose 時間

#### C#

```
1 public class StaticHttpClientService
2 {
3     private static HttpClient _httpClient;
4     private static DateTime _TTL;
5     private static void createInstance()
6     {
7         _httpClient = new HttpClient();
8         _httpClient.BaseAddress = new Uri("http://blog.yowko.com");
9         //設定 dispose HttpClient 的時間
10        _TTL = DateTime.UtcNow.AddMinutes(1);
11    }
12    static StaticHttpClientService()
13    {
14        createInstance();
15    }
16    public HttpClient HttpClientInstance
17    {
18        get
19        {
20            if (DateTime.UtcNow > _TTL)
21            {
22                _httpClient.Dispose();
23                //重新建立 HttpClient
24                createInstance();
25            }
26
27            return _httpClient;
28        }
29    }
30 }
```

### 2. 關閉 socket 連線

將 HttpClient 的 `DefaultRequestHeaders.ConnectionClose` 屬性設定為 `true`，也就是將 HTTP 的 keep-alive header 設為 `false`，讓 socket 在每次處理完 request 即關閉

#### C#

```
1 public class StaticHttpClientService
2 {
3     private static readonly HttpClient _httpClient;
4     static StaticHttpClientService()
5     {
6         var baseUri = new Uri("http://blog.yowko.com");
7         _httpClient = new HttpClient();
8         _httpClient.BaseAddress = baseUri;
```



```

9     _httpClient.DefaultRequestHeaders.ConnectionClose = true;
10  }
11
12     public HttpClient HttpClientInstance = _httpClient;
13 }

```

### 3. 設定釋放 socket 連線時間

避免每次皆關閉 socket 而造成無謂的效能損耗 - 修改 `ConnectionLeaseTimeout` 時間: 用來管理 TCP socket 保持開啟的時間, 預設為 `-1` 永遠開啟 - 修改 `DnsRefreshTimeout` 時間: 用來管理 DNS 更新間隔, 預設為 `12000` (兩分鐘)

#### C#

```

1  public class StaticHttpClientService
2  {
3      private static readonly HttpClient _httpClient;
4      static StaticHttpClientService()
5      {
6          var baseUri = new Uri("http://blog.yowko.com");
7          _httpClient = new HttpClient();
8          _httpClient.BaseAddress = baseUri;
9          //設定 1 分鐘沒有活動即關閉連線, 預設 -1 (永不關閉)
10         ServicePointManager.FindServicePoint(baseUri)
11             .ConnectionLeaseTimeout = (int)TimeSpan.FromMinutes(1).TotalMilliseconds;
12         //設定 1 分鐘更新 DNS, 預設 12000 (2 分鐘)
13         ServicePointManager.DnsRefreshTimeout = (int)TimeSpan.FromMinutes(1).TotalMilliseconds; ;
14     }
15
16     public HttpClient HttpClientInstance = _httpClient;
17 }

```

## 效能比較

### 1. 第一次

Method	Mean	Error	StdDev
Dispose	407.8 us	7.849 us	8.060 us
ConnectionClose	691.4 us	15.557 us	45.869 us
ConnectionLeaseTimeout	423.3 us	8.316 us	16.609 us

Method	Mean	Error	StdDev
Dispose	407.8 us	7.849 us	8.060 us
ConnectionClose	691.4 us	15.557 us	45.869 us
ConnectionLeaseTimeout	423.3 us	8.316 us	16.609 us



## 2. 第二次

Method	Mean	Error	StdDev
Dispose	410.3 us	6.086 us	5.395 us
ConnectionClose	732.0 us	14.589 us	21.384 us
ConnectionLeaseTimeout	415.8 us	5.225 us	4.887 us

```
Method | Mean | Error | StdDev |
-----|-----|-----|-----|
Dispose | 410.3 us | 6.086 us | 5.395 us |
ConnectionClose | 732.0 us | 14.589 us | 21.384 us |
ConnectionLeaseTimeout | 415.8 us | 5.225 us | 4.887 us |
```

## 3. 第三次

Method	Mean	Error	StdDev
Dispose	416.6 us	2.951 us	2.760 us
ConnectionClose	759.2 us	7.041 us	6.586 us
ConnectionLeaseTimeout	422.3 us	11.351 us	14.355 us

```
Method | Mean | Error | StdDev |
-----|-----|-----|-----|
Dispose | 416.6 us | 2.951 us | 2.760 us |
ConnectionClose | 759.2 us | 7.041 us | 6.586 us |
ConnectionLeaseTimeout | 422.3 us | 11.351 us | 14.355 us |
```

## 心得

以效能數據來看，雖說三者執行時間都非常快，但每次 request 都關閉 socket 執行時間是另外兩者的 1.8 倍以上，我相信這在高流量環境下是不被允許的，差距太大了

至於自行管理 HttpClient instance 及使用 ServicePointManager 兩者差距就微乎其微了

## 參考資訊

1. [探討 HttpClient 可能的問題](#)
2. [Singleton HttpClient doesn't respect DNS changes](#)
3. [system.net.http.httpclient does not respect dns update in a timely manner](#)
4. [Beware of the .NET HttpClient](#)
5. [ServicePoint.ConnectionLeaseTimeout Property](#)
6. [ServicePointManager.DnsRefreshTimeout Property](#)



文章作者： Yowko Tsai

上次更新： 2018-01-05

授權合約： 本部落格 (Yowko's Notes) 所有的文章內容(包含圖片) · 任何轉載行為 · 必須通知並獲本部落格作者 (Yowko Tsai) 的同意始得轉載,且轉載皆須註明出處與作者。



Yowko's Notes 由 Yowko Tsai 製作 · 以創用CC 姓名標示-非商業性-相同方式分享 3.0 台灣 授權條款 釋出。

標籤： [C#](#) [Benchmark](#)

下一篇 [在 ASP.NET MVC 5 中使用 ASP.NET Core Dependency Injection 與 HttpClientFactory](#)

上一篇 [\[Benchmark\] DB 物件對映至 C# class 的做法](#)

What do you think?

1条回复  
**重要更新**

👍 Upvote 🤔 Funny 😍 Love 😲 Surprised 😡 Angry 😞 Sad

登录 Disqus 后, 我们会处理您的个人数据以方便您进行身份验证并发布评论。我们还会保存您发布的评论, 世界各地的人都能立即查看并搜索这些评论。

请访问我们的隐私政策, 以便了解 Disqus 会收集您的哪些个人信息, 以及如何使用它们。同时, 对于我们的服务用户, 还需要接收服务条款这些内容。

0条评论 Honda Chen

👍 推荐 🐦 推文 📄 分享 继续 评分最高

开始讨论...

来做第一个留言的人吧!

📧 订阅 📍 在您的网站上使用 Disqus添加 Disqus添加 🚫 Do Not Sell My Data



由 [Hugo](#) 強力驅動 | 主題 - [Even More](#)

© 2016 - 2020 ❤️ Yowko Tsai

