



灰色世界

think it three times

随笔 - 87, 文章 - 0, 评论 - 115, 阅读 - 26万

导航

博客园
首页
新随笔
管理

| 2025年8月 | | | | | | |
|---------|----|----|----|----|----|----|
| 日 | 一 | 二 | 三 | 四 | 五 | 六 |
| 27 | 28 | 29 | 30 | 31 | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 1 | 2 | 3 | 4 | 5 | 6 |

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

最新随笔

1. StiReport使用
2. Asp.net core authentication
3. PO BOX地址校验
4. SQL SERVER 性能监视和优化工具
5. EasyNetQ笔记
6. javascript 对象，函数，原型和 this
7. 使用PerfView监测.NET程序性能（四）：折叠，过滤和时间范围选择
8. 使用PerfView监测.NET程序性能（三）：分组
9. Improving .NET Application Performance and Scalability
10. 使用PerfView监测.NET程序性能（二）：Perfview的使用

我的标签

javascript(9)
xmpp(2)
http头域(2)
http header(2)
ECMAScript(2)
活动对象(2)
闭包(2)
wpf线程模型(1)
vs远程调试(1)
Variable Object(1)
更多

积分与排名

积分 - 91326
排名 - 17832

随笔分类 (121)

.NET(24)
C/C++(2)
IT人生(1)
javascript(10)
MySQL(1)
SQL Server(14)
xmpp(2)
国际物流(1)
数据结构(3)
性能与优化(13)
原创文章(35)

使用PerfView监测.NET程序性能（二）：Perfview的使用

在上一篇博客中，我们了解了对Windows及应用程序进行性能分析的基础：Event Trace for Windows (ETW)。现在来看看基于ETW的性能分析工具——Perfview.exe

Perfview简介

Perfview是一个开源的CPU和内存性能分析工具，也包括一些针对.NET的分析功能，例如GC分析，JIT分析，甚至ASP.NET中的请求统计等等。Perfview是一个Windows应用程序，但也能对在Linux系统上采集的数据进行分析（参考）。Perfview免安装，而且只是一个14M的.exe文件，非常容易部署到需要进行性能分析的机器上，例如生产环境的服务器。而且在性能数据收集的过程中不需要重启应用程序或者服务器，而且收集的性能数据日志（.etl文件）可以被拷贝到其他Windows机器上，再进行分析工作，对业务的影响非常少。

Perfview已迁移到GitHub上，可以在上面下载Perfview.exe，clone库或者查看相关资料。

Perfview GitHub: <https://github.com/Microsoft/perfview>

Perfview视频教程：<https://channel9.msdn.com/Series/PerfView-Tutorial>

Vance Morrison关于Perfview的博客：

<https://blogs.microsoft.com/vancem/tag/perfview/>

Perfview使用

在简单介绍Perfview后，我们来使用Perfview进行一个小小的性能分析，来熟悉一下Perfview的基本操作。

这个实验使用的代码，就是Vance Morrison在视频教程中用到的Console程序。代码可以在Perfview自带的帮助文件中找到。

```
using System;
// using System.Collections.Generic;

class Program
{
    public static int aStatic = 0;
    // Spin is a simple compute bound program that lasts for 5 seconds
    // It is a useful test program for CPU profilers.
    static int Main(string[] args)
    {
        int numSec = 5;
        if (args.Length == 1)
            numSec = int.Parse(args[0]);

        Console.WriteLine("Spinning for {0} seconds", numSec);
        RecSpin(numSec);
        return 0;
    }

    // Spin for 'timeSec' seconds. We do only 1 second in this
    // method, doing the rest in the helper.
    static void RecSpin(int timeSec)
    {
        if (timeSec <= 0)
            return;
    }
}
```

转载的好文章(15)

随笔档案 (87)

2021年10月(1)
 2021年4月(1)
 2020年12月(1)
 2020年4月(1)
 2019年10月(1)
 2019年8月(1)
 2018年12月(3)
 2018年11月(6)
 2018年9月(1)
 2018年8月(2)
 2017年5月(1)
 2017年4月(12)
 2017年3月(2)
 2017年2月(1)
 2016年4月(1)
 2016年3月(2)
 2016年2月(1)
 2014年12月(1)
 2014年7月(2)
 2014年1月(2)
[更多](#)

阅读排行榜

1. 行转列：SQL SERVER PIVOT与用法解释(85726)
2. B树详解(32629)
3. 基于JWT的web api身份验证及跨域调用实践(29483)
4. HTTP头域列表与解释 之 request篇(1 4093)
5. [c#] 反射真的很可怕吗? (9859)
6. 使用PerfView监测.NET程序性能
(二)：Perfview的使用(9324)
7. apache不能启动：Windows无法启动Apache2.2服务，错误1067。(5459)
8. IIS7中的站点，应用程序和虚拟目录详解(4531)
9. HTTP头域列表与解释 之 response篇(4241)
10. 基于WPF+XMPP的IM程序开发日志之一：开篇(3846)

评论排行榜

1. [c#] 反射真的很可怕吗? (26)
2. 行转列：SQL SERVER PIVOT与用法解释(20)
3. 分享一个基于FileSystemWatcher的文件自动备份程序(15)
4. 基于WPF+XMPP的IM程序开发日志之一：开篇(10)
5. 基于JWT的web api身份验证及跨域调用实践(9)
6. 使用http module 对url进行重写的尝试(6)
7. asp.net程序员与php程序员，傻瓜机用户与单反机用户(6)
8. [转载]大型网站架构演变和知识体系(4)
9. HTTP头域列表与解释 之 request篇(3)
10. 基于WPF+XMPP的IM程序开发日志之三：用户头像Avatar(3)

推荐排行榜

1. 行转列：SQL SERVER PIVOT与用法解释(34)
2. 使用PerfView监测.NET程序性能
(二)：Perfview的使用(11)
3. [c#] 反射真的很可怕吗? (10)
4. 基于JWT的web api身份验证及跨域调用实践(8)
5. 基于WPF+XMPP的IM程序开发日志之一：开篇(6)

最新评论

1. Re:行转列：SQL SERVER PIVOT与用法解释

```
--timeSec;
SpinForASecond();
RecSpinHelper(timeSec);
}

// RecSpinHelper is a clone of RecSpin. It is repeated
// to simulate mutual recursion (more interesting example)
static void RecSpinHelper(int timeSec)
{
    if (timeSec <= 0)
        return;
    --timeSec;
    SpinForASecond();
    RecSpin(timeSec);
}

// SpinForASecond repeatedly calls DateTime.Now until for
// 1 second. It also does some work of its own in this
// methods so we get some exclusive time to look at.
static void SpinForASecond()
{
    DateTime start = DateTime.Now;
    for (; ; )
    {
        if ((DateTime.Now - start).TotalSeconds > 1)
            break;

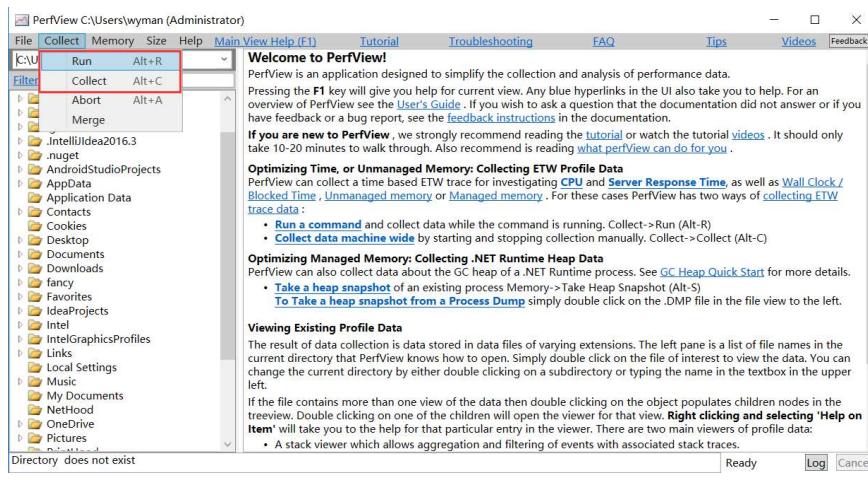
        // Do some work in this routine as well.
        for (int i = 0; i < 10; i++)
            aStatic += i;
    }
}
}
```



以上代码很简单，SpinForASecond()在一秒内不断调用DateTime.Now，而RecSpin()和RecSpinHelper()则不断地相互调用对方。这里使用循环的目的是，循环执行是一种典型的CPU密集型操作，而RecSpin()和RecSpinHelper()则是为了丰富程序的函数调用栈。

步骤一：收集程序运行数据，生成由ETW数据组成的.etl文件。

Perfview提供两种收集数据的方式，Run和Collect。“Run”是直接指定需要启动的应用程序的名称，以便启动该程序。“Collect”则是直接启动Perfview并开始收集。但不要以为“Run”方式只收集指定程序的数据。事实上无论哪种方式，Perfview都会收集系统范围内全部数据，并且收集完成后，需要选择某一个进程以进行分析。



@xhb 这里有动态列的方法。感谢楼主分享。 ...
--不懂01的ITer-Jack
2. Re:使用PerfView监测.NET程序性能
(二) : Perfview的使用
支持 支持

--winds_随风

3. Re:行转列: SQL SERVER PIVOT与用法解释

你好, 可以转载吗?

--七加一

4. Re:B树详解

关于B树的高度这里有问题, 底数T应该是非根非叶结点的最小孩子数目, 也就是M/2的上限, 得到的值还需要再进行加1, 才能得到B树的高度。我刚才用你的公式算, 感觉不太对, 后来在课本上找到了这个求高度的...

--车照123

5. Re:HTTP头域列表与解释 之 request 篇

怎么判断头部是否包含Authorization呢
--陌生人, 你好

6. Re:SQL Server 死锁概念和分析

想请教下moe_bookfolder是系统表还是自定义视图

--TheCloud

7. Re:基于JWT的web api身份验证及跨域调用实践

@ 跟着阿笨一起玩.NET真不要脸, 一个破视频要69.9块, 还到处贴链接, 几个地方都看到你了! ...

--大雄小顾

8. Re:使用PrefView监测.NET程序性能
(三) : 分组

支持支持。重装农药第18天

--牛腩

9. Re:使用PrefView监测.NET程序性能
(二) : Perfview的使用

赞

--雪峰

10. Re:ETW (Event Tracing For Windows) – what it is and useful tools 坏哥厉害

--hongkong_8

11. Re:基于JWT的web api身份验证及跨域调用实践

ASP.NET WebApi 基于JWT实现Token签名认证ASP.NET WebApi 基于分布式Session方式实现Token签名认证...
--跟着阿笨一起玩.NET

12. Re:基于JWT的web api身份验证及跨域调用实践

mark

--大漠孤阳

13. Re:基于JWT的web api身份验证及跨域调用实践

public class ApiAuthorizeAttribute :
AuthorizeAttribute { protected
override bool IsAuthorized(HttpA...
--羽赐命

14. Re:基于JWT的web api身份验证及跨域调用实践

<!--注释掉下面这个解决预请求验证失败--> <!-->
name="ExtensionlessUrlHandler-Integrated-4.0" path="*." verb="*"
ty...
--myskysoft

15. Re:基于JWT的web api身份验证及跨域调用实践

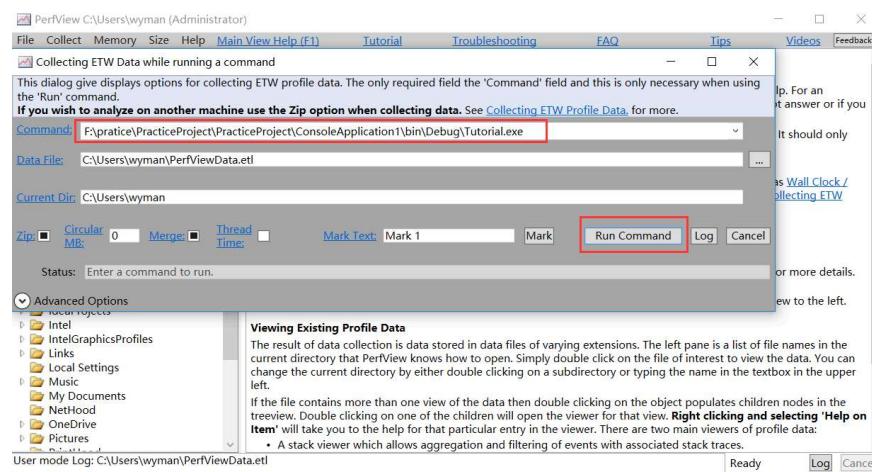
请教: 像这种跨域请求, 客户端一般应该把jwt保存在哪里, cookie不太合适了吧。

--Esofar

使用PerfView监测.NET程序性能 (二): Perfview的使用 - wyman25 - 博客园

我们以“Run”方式来收集以上代码生成的Tutorial.exe程序。

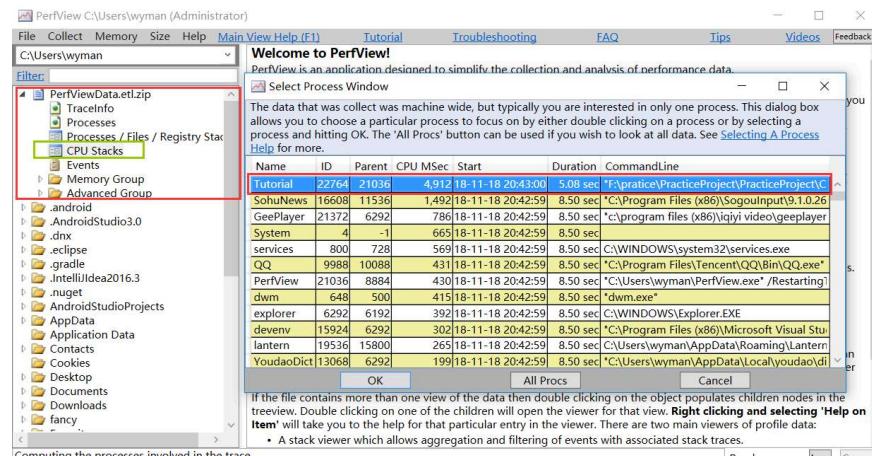
在弹出的对话框中, 填入需要启动Tutorial.exe的全文件名, 以及填入生成etl文件的文件名 (这里是PerfViewData.etl), 并点击“Run Command”:



Perfview收集和处理数据的时间比较长。在处理过程中, Perfview的右下角会闪动, 并且可以查看运行日志, 了解到当前Perfview在执行什么工作。

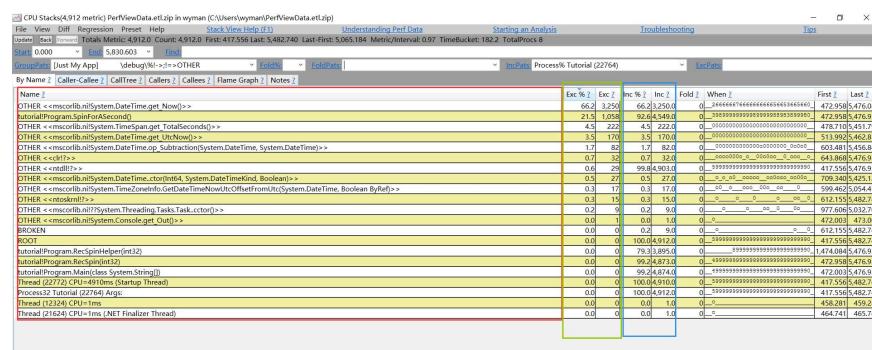
步骤二: 选择需要分析的进程

在收集完毕后, 在左边选择“PerfViewData.etl.zip”, 并在展开的选择项中双击选择“CPU Stacks”, 此时, 会弹出进程选择对话框, 选择需要进行CPU分析的进程。这里选择我们运行的Tutorial.exe进程。



步骤三: 查看执行栈视图

在双击选择了“Tutorial.exe”的进程后, 进入到程序详细的执行栈的视图中。这里记录着Tutorial.exe的函数调用树, 以及函数的执行时间。



在该视图中，你可以看到Tutorial.exe的函数调用情况，包括函数调用树（Call-Tree），某个函数的调用者（Calls）和被该函数调用的函数（Callees），另外，在视图右侧，是函数的执行时间，其中，“Exc”是指 Exclusive，是指函数自己（不包含该函数里执行的子函数）的执行时间，而“Inc”是指Inclusive,指该函数及该函数中执行的子函数的总的执行时间。

另外，这个执行时间是怎么认定的呢？答案是CPU采样。Perfview对CPU进行采样，默认每个CPU采样是1毫秒（在Prefview的高级设置中可以设置到0.125毫秒~1毫秒），每次采样中可以得到当前CPU正则执行什么代码。例如DateTime_getNow()有3250采样，则可以说明在整个程序运行中，DateTime_getNow()占用了3250毫秒的CPU时间，占整个运行时间的66.2%。通过比较各个函数的执行时间，我们就可以知道程序中哪个函数占用比较多的CPU时间。

以上便是Prefview的基本的使用步骤。Prefview提供了非常多并强大的功能，例如分组（Grouping），折叠（Folding），时间范围选择，这些在后续教程里再聊。而更强大的是，F1帮助手册里，有着非常详细的使用说明和术语解析，而且界面上几乎每个功能都有说明的ToolTip和说明的超链接如果对某个功能用法不是很清楚，可以方便地找到说明，真是业界良心。

参考资料

[How many samples are enough when using a sample based profiler in a performance Investigation](#)

[The TraceEvent Library Programmers Guide](#)

系列目录

[使用PerfView监测.NET程序性能（一）：Event Trace for Windows](#)

[使用PerfView监测.NET程序性能（二）：Perfview的使用](#)

[使用PerfView监测.NET程序性能（三）：分组](#)

[使用PerfView监测.NET程序性能（四）：折叠，过滤和时间范围选择](#)

分类: [原创文章](#), [性能与优化](#)

[好文要顶](#) [关注我](#) [收藏该文](#) [微信分享](#)



wyman25

粉丝 - 59 关注 - 15

11

0

[+加关注](#)

[升级成为会员](#)

[« 上一篇：【转载】Configure the max limit for concurrent TCP connections](#)

[» 下一篇：Improving .NET Application Performance and Scalability](#)

posted on 2018-11-23 19:14 wyman25 阅读(9325) 评论(2) 收藏 举报

评论

[默认](#) | [按时间](#) | [按支持数](#)

#1楼 回复 引用

赞

支持(0) 反对(0)

2018-11-23 21:32 | 云卷云舒2021

#2楼 回复 引用

支持 支持

支持(0) 反对(0)

2021-03-09 16:58 | winds_随风

[发表评论](#) [升级成为园子VIP会员](#)[编辑](#) [预览](#)[B](#) [🔗](#) [ⓘ](#) [“”](#) [☒](#)

支持 Markdown

 自动补全[提交评论](#) [退出](#) [订阅评论](#) [我的博客](#)

[Ctrl+Enter快捷键提交]

编辑推荐：

- 下划线字段在golang结构体中的应用
- SQL Server也能玩正则表达式？
- CUDA 编程初探
- 《C#高级GDI+实战：从零开发一个流程图》增加贝塞尔曲线
- AES 加密模式演进：从 ECB、CBC 到 GCM 的 C# 深度实践

阅读排行：

- 在本地部署Qwen大语言模型全过程总结
- 十年大厂员工终明白：MySQL性能优化的尽头，是对B+树的极致理解
- Coze工作流实战：一键生成历史人物一镜到底爆款短视频
- Open JDK 和 Oracle JDK傻傻分不清楚
- 记一次OOM

历史上的今天：

2011-11-23 我所知的javascript之prototype

博客园 © 2004-2025



浙公网安备 33010602011771号 浙ICP备2021040463号-3