灰色世界

think it three times

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)

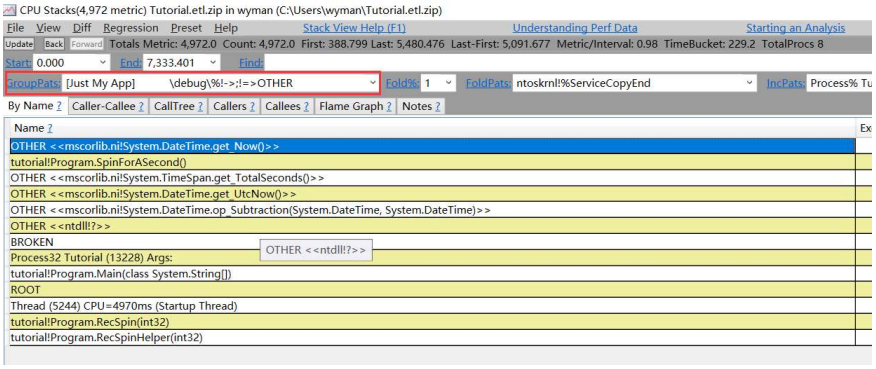
使用PerfView监测.NET程序性能（三）：分组

在上一篇博客中，我们通过Perfview帮助文件中自带的代码来简单使用了Perfview，了解了基本操作。现在来看看Perfview中的分组操作（Grouping）。分组功能都旨将记录到的各种函数调用堆栈以指定的规则进行分组，帮助你组织和找到更关心的数据。

为什么需要分组

在实际使用中，PerfView通常会收集到非常多的函数调用栈数据，包括我们关心的程序的函数调用信息，及一大堆我们不关心的其他函数调用信息，例如windows系统的底层函数。这么多有用没用的条目都列出在列表视图上，令人眼花缭乱。如何将我们不需要的数据分组归纳呢？Perfview提供分组功能。

分组功能使用类似于正则的匹配功能，将函数全名（一个函数的全名包含了程序集，命名空间，类名和函数名，例如"mscorlib.n!System.DateTime.get_Now()"）进行匹配，并替换成自定义的分组名称。例如，可以对所有在Debug目录下的程序集的函数单独显示，而其他函数则分组成“OTHER”，这样，我们就可以只看见我们程序里的函数调用。其实这就是默认的[Just My App]分组规则的作用：



通配符

那么，分组功能如何使用呢？

在使用分组之前，先看看PerfView定义的几个"通配符":

* :匹配任意数量的字符

%：匹配任意数量的数字和英文字母和点号(".")，等于.NET正则中的 [\w\d.]*

^ ： 匹配开头

| ： “或”操作

{}：代表一个分组，等于.NET正则里的小括号

分组规则

PerfView中有两种分组操作，分别是 PAT->GROUP 和 PAT=>GROUP。在这里“PAT”代表需要匹配的模式(Pattern)，“GROUP”代表你自定义的组名。 而这两种分组方式区别就在于中间的“->”和"=>"，前者表示忽略入口函数，后者则会将入口函数显示在分组中。有时我们希望知道一个分组里的函数最初是由哪个函数开始调用的，这时候就可以使用后者了。除此之外，两种分组方式没有其他不同。

我们看看具体的分组规则的使用：

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(14093)
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

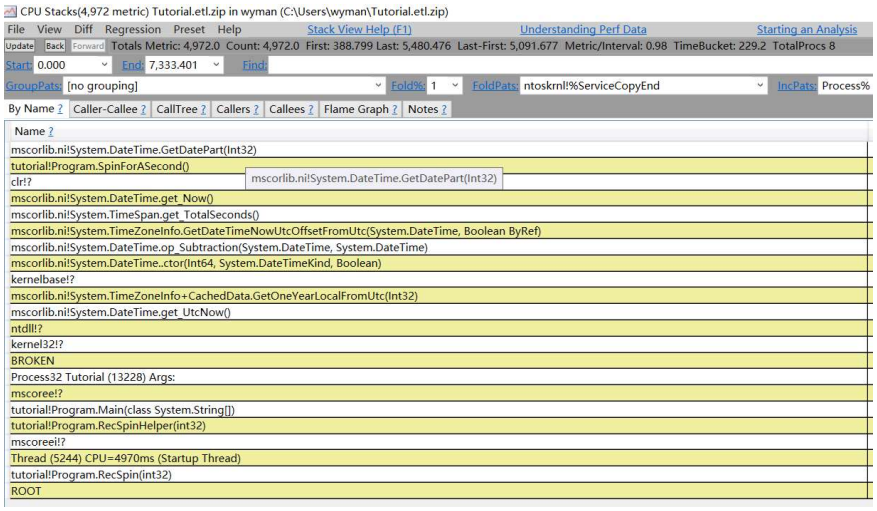
1. 行转列：SQL SERVER PIVOT与用法解释(34)
2. 使用PerfView监测.NET程序性能（二）：Perfview的使用(11)
3. [c#] 反射真的很可怕吗？(10)
4. 基于JWT的web api身份验证及跨域调用实践(8)
5. 基于WPF+XMPP的IM程序开发日志之一：开篇(6)

最新评论

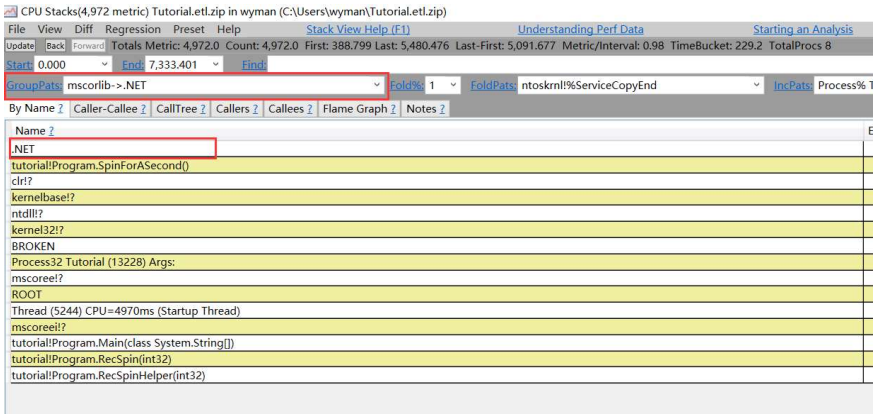
1. Re:行转列：SQL SERVER PIVOT与用法解释

1. PAT->GROUP形式

如上述，PAT->GROUP形式是简单地将一个函数的全名称中包含"PAT"字眼的条目都分到“GROUP”组中，例如，在不分组的情况下，我们收集到的函数调用数据列表是这样的：



里面有我们Tutorial.exe的函数，例如，tutorial!Program.SpinForASecond()和tutorial!Program.RecSpin()，同时也有很多.NET的内部函数，例如mscorlib.n!System.get_Now()和mscorlib.n!System.TimeSpan.get_TotalSeconds()，等等。假设我们只关心tutorial.exe自身的函数，而不希望被.NET内部函数所干扰，我们则可以设置一个分组规则“mscorlib->.NET”，这样，所有包含“mscorlib”字眼的方法全名称的条目都会被分组进“.NET”组，效果如下：



是不是清爽了很多？这样的分组能使我们快速地过滤掉mscorlib有关的函数，只剩下tutorial自己的函数（和一些其他函数，当然如果你愿意，也可以将其他的函数"分组"掉）

2. {*}=>$1

该形式的规则意思是： 花括号里匹配到的条目会被分组，而组名正是是花括号里的匹配到的内容， “$1”是一个占位符，对应的是花括号“{}”里的内容。假设有两个函数：tutorial!Program.SpinForASecond()和tutorial!Program.RecSpin()， 而应用的规则是“{tutorial!}->My APP $1”，则分组后，这两个函数被分进一组，并且组名为“My App tutorial!”

Perfview还支持同时设置多个规则，例如设置规则为“{tutorial!}->My APP $1;{mscorlib.ni!}->Internal $1”，这里有两个规则，一个是蓝色部分，另一个是红色部分，中间用分号(;)隔开。如果函数全名中有"tutorial!"的就分进名为“My APP tutorial!”组，而有“mscorlib.ni!”字眼的就分进“Internal mscorlib.ni!”组。

3.PAT=>GROUP

最后来看看入口点规则分组(Entry Point Grouping)。前边提到过， “PAT=>GROUP”与“PAT->GROUP”的不同在于，后者会忽略掉该组的入口函数，意味着你很难知道某个分组里的

@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i

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公...

--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

--TheClound

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...

--一羽赐命

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--> <!--<add name="ExtensionlessUrlHandler-Integrated-4.0" path="*." verb="*" ty...

--myskysoft

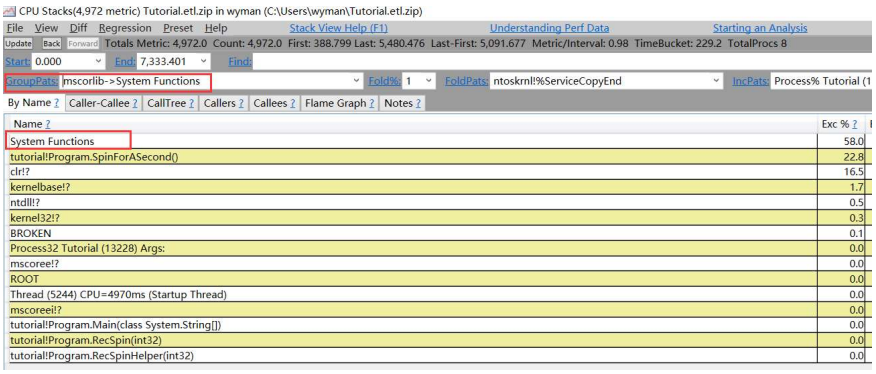
15. Re:基于JWT的web api身份验证及跨域调用实践

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。

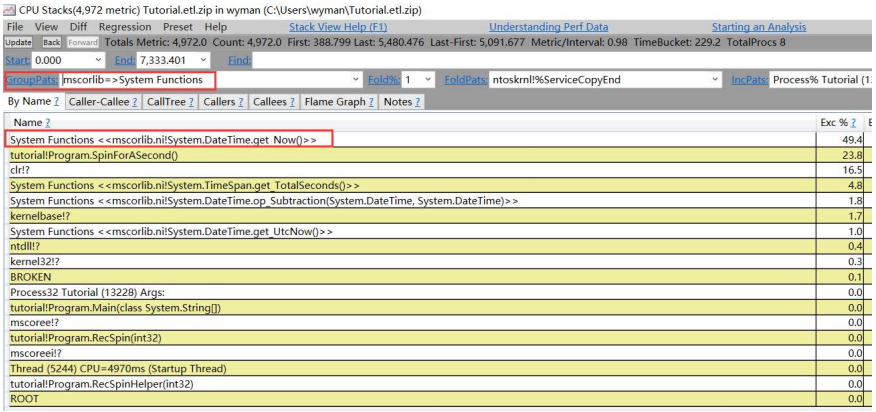
--Esofar

函数是从哪个函数执行进去的，而前者则会包含入口点函数信息。我们通过图例来看看实际效果。

下图中，使用“mscorlib->System Functions”规则来对mscorlib的函数进行分组，组名为“System Functions”,但除非你展开这个分组的明细，查找每个函数调用树，否则你不知道是什么函数调用了这组函数。



而现在使用“mscorlib=>System Functions”，看看有什么不同：



System Functions组明确指示了该组的函数的入口点是“mscorlib.ni!System.DateTime.get_Now()”函数，即DateTime.Now导致了这些函数的执行。

以上便是PerfView的分组功能。但其实这只是分组功能中的一部分。通过规则的搭配可以有更强大的效果。而最全面的说明其实是在PerfView自带的F1帮助文件。这里只作一个抛砖引玉的简要说明。因此如果需要了解更全面的分组技巧，可以去帮助文件里搜索相关主题。

系列目录

分类: 原创文章, 性能与优化

wyman25
粉丝 - 59 关注 - 15

« 上一篇： Improving .NET Application Performance and Scalability
» 下一篇： 使用PerfView监测.NET程序性能（四）：折叠，过滤和时间范围选择

posted on 2018-12-08 21:54 wyman25 阅读(1976) 评论(1) 收藏 举报

评论

默认 | **按时间** | 按支持数

#1楼 回复 引用

支持支持。重装农药第18天

支持(0) 反对(0)

2018-12-09 00:50 | 牛腩

刷新评论 刷新页面 返回顶部

发表评论 升级成为园子VIP会员

编辑 预览

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]