

عنوان: بررسی علت CPU Usage بالای برنامه در حال اجرا

نویسنده: وحید نصیری

تاریخ: ۱۳۹۰/۰۴/۳۱ ۰۰:۲۸:۰۰

آدرس: www.dotnettips.info

برچسب‌ها: Tips

فرض کنید به یک سرور مراجعه کرده‌اید و شکایت از CPU Usage مربوط به پروسه w3wp.exe یا همان IIS Worker Process است که بالای 90 درصد می‌باشد. بر روی این سرور هم هیچ چیز دیگری نصب نیست و مطابق مقررات موجود، قرار هم نیست که برنامه‌ای نصب شود. اکنون سؤال این است که چطور تشخیص می‌دهید، کدام قسمت یکی از برنامه‌های دات نتی در حال اجرا (در اینجا یکی از برنامه‌های ASP.NET هاست شده)، سبب بروز این مشکل شده است؟ کدام ترد بیشترین زمان CPU را به خود اختصاص داده است؟ چطور باید خطایابی کرد؟

اولین کاری که در این موارد توصیه می‌شود مراجعه به برنامه‌ی معروف [process explorer](#) و بررسی برگه‌ی threads آن است. در اینجا حتی می‌توان call stacks مرتبط با یک ترد را هم مشاهده کرد. اما ... این برگه در مورد پروسه‌ها و تردهای دات نتی، اطلاعات چندانی را در اختیار ما قرار نمی‌دهد.

خوشبختانه امکان دیباگ پروسه‌های دات نتی در حال اجرا توسط کتابخانه‌ی MdbgCore.dll پیش بینی شده است. این فایل را در یکی از مسیرهای ذیل می‌توانید پیدا کنید:

```
C:\Program Files\Microsoft SDKs\Windows\vXYZ\bin\MdbgCore.dll
C:\Program Files\Microsoft SDKs\Windows\vXYZ\bin\NETFX 4.0 Tools\MdbgCore.dll
```

در ادامه می‌خواهیم توسط امکانات این کتابخانه، به stack trace تردهای در حال اجرای یک برنامه دات نتی دسترسی پیدا کرده و سپس نام متدهای مرتبط را نمایش دهیم:

```
using System;
using System.Collections;
using System.Diagnostics;
using Microsoft.Samples.Debugging.MdbgEngine;

namespace CpuAnalyzer
{
    class Program
    {
        static void Main(string[] args)
        {
            var engine = new MDbgEngine();

            var processesByName = Process.GetProcessesByName("MyApp");
            if (processesByName.Length == 0)
                throw new InvalidOperationException("specified process not found.");
            var processId = processesByName[0].Id;

            var process = engine.Attach(processId);
            process.Go().WaitOne();

            foreach (MDbgThread thread in (IEnumerable)process.Threads)
            {
                foreach (MDbgFrame frame in thread.Frames)
                {
                    if (frame == null || frame.Function == null) continue;
                    Console.WriteLine(frame.Function.FullName);
                }
            }

            process.Detach().WaitOne();
        }
    }
}
```

در اینجا در ابتدا نیاز است تا pid یا process-id مرتبط با برنامه در حال اجرا یافت شود. سپس از این pid جهت اتصال (engine.Attach) به پروسه مورد نظر استفاده خواهیم کرد. در ادامه کلیه تردهای این پروسه در حال دیباگ لیست شده و سپس

MDbgFrame های این ترد بررسی می‌شوند و نام متدهای مرتبط در کنسول نمایش داده خواهند شد. خوب در مرحله بعد شاید بد نباشد که این متدها را بر اساس CPU usage آن‌ها مرتب کنیم. به این ترتیب بهتر می‌توان تشخیص داد که کدام متد مشکل ساز بوده است. برای این منظور باید به API ویندوز و تابع GetThreadTimes مراجعه کرد و اولین پارامتر ورودی آن، همان thread.CorThread.Handle باشد. هر کدام که جمع KernelTime + UserTime بیشتری داشت، همان است که مشکل درست کرده است.

```
[DllImport("kernel32.dll", CharSet = CharSet.Auto, SetLastError = true)]
public static extern bool GetThreadTimes(IntPtr handle, out long creation, out long exit, out long
kernel, out long user);
```

این مورد را به عنوان تمرین بررسی کرده و ادامه دهید! همچنین بهتر است جهت دستیابی به اطلاعاتی معتبر، اولین حلقه برنامه فوق، حداقل 10 بار اجرا شود تا اطلاعات آماری بهتری را بتوان ارائه داد. البته در این حالت نکته‌ی زیر باید رعایت شود:

```
for (int i = 0; i < 10; i++)
{
    foreach (MDbgThread thread in (IEnumerable)process.Threads)
    {
        //...
    }
    process.Go();
    Thread.Sleep(1000);
    process.AsyncStop().WaitOne();
}
```

در کل این مثال جای کار زیاد دارد. برای مثال طراحی یک رابط کاربری برای آن و نمایش جزئیات بیشتر. به همین منظور حداقل سه پروژه مشابه را می‌توان نام برد:

[Managed Stack Explorer](#)

[CLR Stack Explorer](#)

[CPU-analyzer](#)

نظرات خوانندگان

نویسنده: Nima

تاریخ: ۱۳۹۰/۰۵/۰۱ ۰۸:۲۰:۱۵

سلام آقای نصیری

بخاطر آموزش بسیار مفیدتون ممنون.یه مطلب جالب بگم بارها شده دقیقا چیزی که باهاش به مشکل خوردم رو شما در پست بعدی وبلاگتون آموزش دادین D:

یه سوال از خدمت شما داشتم اینکه چند وقته بطور اتفاقی CUP Usage سرور ما برای 2 3 بار در روز به 100 درصد میرسه و این حالت حدود 5 دقیقه ادامه داره. برنامه های ما به صورت Service Oriented نوشته شده و سرویسهای مختلفی (WCF) روی IIS هاست شدن. اگر بخوام کد بالا زمانی که CPU Usage مثلا از 80% بیشتر بشه فعال بشه چکار باید بکنم؟ آیا لازمه یک ویندوز سرویس بنویسم؟
با تشکر

نویسنده: وحید نصیری

تاریخ: ۱۳۹۰/۰۵/۰۱ ۰۸:۲۷:۰۴

سلام، نه. تمام سرویسها و تمام برنامهها هم در نهایت یک پروسه در ویندوز هستند. شما در کد بالا بجای MyApp ، نام پروسه دات نتی مورد نظر را قرار دهید (حتی اگر پروسه مثل w3wp باشد (یعنی یک سری برنامه دات نتی رو هاست کرده) هم کار می کند). سپس می تونید شروع به دیباگ کنید. البته قسمت اندازه گیری KernelTime + UserTime را که ذکر شد خودتون باید لحاظ کنید. به این ترتیب (به کمک مرتب سازی بر اساس KernelTime + UserTime) می تونید نام متدی که الان سبب بروز این CPU usage بالا شده رو دقیقا تشخیص بدید.

نویسنده: Nima

تاریخ: ۱۳۹۰/۰۵/۰۱ ۰۸:۳۱:۱۸

ممنون آقای نصیری

از کجا میشه درصد استفاده از CPU رو متوجه شد؟ وقتی CPU Usage ما به 100 درصد میرسه نه میشه به سرور کانکت شد و نه اینکه وقتی کانکتیم میذاره برنامه ای رو اجرا کنیم. بخاطر همین مجبوریم وقتی درصد از حدی بالاتر رفت بلافاصله یه سری اطلاعات را ثبت کنیم.

نویسنده: وحید نصیری

تاریخ: ۱۳۹۰/۰۵/۰۱ ۰۸:۴۵:۲۱

در یک چنین حالتی بله. می تونید سرویس ویندوز ان اتی بنویسید برای بررسی خودکار. مطلب فوق یک بررسی دستی است. یک تایمر داخل این سرویس تعریف کنید سپس به کمک WMI ([+](#)) هر از چند لحظه یکبار، وضعیت CPU usage کل سیستم را بررسی کنید. سپس اگر بالا بود می شود کل پروسه های ویندوز را لیست کرد و نکته فوق را (بر اساس مرتب سازی اطلاعات با توجه به KernelTime + UserTime) به آنها اعمال کرد. نهایتا یک گزارش متنی یا یک ایمیل می شود از این برنامه گرفت.

نویسنده: masud

تاریخ: ۱۳۹۰/۰۵/۰۱ ۱۲:۵۲:۴۰

با سلام خدمت آقای نصیری

چند وقتی است روی یکی از سرورهی ما مش out of memory به وجود آمده
مشخصات سرور ویندوز 2003 سی دو بیتی رم بالای 4 گیگابایت که با سویچ PAE/ در فایل بوت ... این حافظه به ویندوز معرفی شده در

ضمن اینکه Lock Page in memory برای یوزری که سرویس IIS را Start کرده تنظیم شده در این حالت باز ما خطا را میگیرم چیزی که به ذخن من میرسه ما در برنامه از کریستال ریپورت استفاده کردیم که داخل اون یک library ار نوع com وجود داره که کار

تبدیل تاریخ میلادی به شمسی رو انجام میدم فکر میکنم به خاطر اینکه این کتابخانه به شکل unmanage code است سوالی که دارم اینه چه طور می توانم مطمئن بشم مشکل از این کتابخانه است و یا خیر

لازم میدونم اشاره کنم تعداد کاربرای سایت بالا و همچنین این سرور مختص iis است و به هیچ عنوان SQL روی اون نصب نمی باشد

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۵/۰۱ ۱۶:۳۲

سوال شما مرتبط با بحث CPU Usage نیست...

- در سرور 32 بیتی شما نمی تونید استفاده مناسبی از سخت افزاری که تهیه کردید بکنید. نیاز است به سرور 64 بیتی نقل مکان کنید. یعنی هزینه کردید اما ... از آن استفاده بهینه ای نمی کنید.

- تابع SQL ، برای تبدیل تاریخ میلادی به شمسی وجود دارد (بگردید در انجمن ها هست). از همان تابع در کوئری های خودتون قبل از ارسال به گزارشات استفاده کنید. یعنی دیتا از همان اول تاریخ شمسی داشته باشد. به این صورت نیازی به استفاده از افزونه دیگری نیست.

- IIS یک قسمت دارد به نام Application pool . اولاً به ازای هر برنامه بهتر است یک Application pool جدا درست کنید. این حالت دقیقاً مثل اجرای هر سایت در یک پروسه جدا است. مثلاً IE8 رو دیدید که هر tab آن در یک پروسه جدا اجرا می شود، یا مثلاً مرورگر کروم هم به همین صورت است. Application pool در IIS هم دقیقاً همین معنا را دارد. عادت متداول، استفاده از یک Application pool به ازای تمام سایت ها است که غلط است. یعنی زمانیکه این توانایی هست چرا استفاده نمی کنید؟ ثانیاً در همین Application pool می تونید تنظیم کنید که اگر میزان مصرف حافظه برنامه منتسب، مثلاً به 600 مگ رسید لطفاً خودت به صورت خودکار اون رو recycle کن تا مشکلات کمبود حافظه رخ ندهد.

- از برنامه های dotTrace و Ants memory profiler هم می تونید استفاده کنید. این ها قابلیت دیباگ پروسه های مرتبط با IIS رو هم دارند و دقیقاً گزارشات آماری پیشرفته ای می تونند از وضعیت مصرف حافظه قسمت های مختلف برنامه در اختیار شما قرار بدن: (+) و (+)

نویسنده: Nima
تاریخ: ۱۳۹۰/۰۵/۰۱ ۱۵:۲۴

سلام مجدد آقای نصیری

من پیرو این مطلب مفید شما یک برنامه نوشتم اما با مشکلاتی مواجه شدم که در اینجا سوال پرسیدم :

<http://stackoverflow.com/questions/6799007/check-cpu-usage-and-log-processes-that-have-up-to-30-cpu-usage-in-windows>

قضیه اینه که CPU من 4 هسته داره و من فکر میکنم چیزی که WMI به من میدم مال یکی از هسته هاست. یعنی اصلاً نمیخونه عددی هر 100 میلی ثانیه میخونم با عددی که در Task Manager میبینم. درضمن من فکر کردم که پراسسی که ماکزیمم استفاده از CPU رو داره رو اطلاعاتش رو بگیرم ولی متأسفانه هر چی گشتم چیز مناسب و بدرد بخوری پیدا نکردم. ممنون میشم من رو کمی راهنمایی کنید.

باتشکر

نویسنده: Hossein Raziee
تاریخ: ۱۳۹۰/۰۵/۰۱ ۱۶:۰۵

با درود

جناب نصیری برام سوالی مطرح شد. شما مخالف ثبت تاریخ به صورت میلادی در دیتابیس هستید؟ و یا در این مورد خاص که دوستمون گفتند این موضوع رو پیشنهاد میکنید؟

از این که سوال ارتباطی با موضوع پست نداره پوزش میخوام ولی برام جالب بود بدونم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۵/۰۱ ۱۶:۴۵:۵۲

استفاده از تاریخ میلادی در دیتابیس خوبه. مثلا آمارگیری از تاریخ تا تاریخ یا اعمال بسیاری از امکانات توکار بانک‌های اطلاعاتی. اما یکجا مشکل ساز می‌شود و آن هم گروه بندی بر اساس ماه‌های شمسی است. مثلا گزارش جمع حقوق کارکنان را بر اساس ماه‌های شمسی یک سال تهیه کنید. این گزارش 12 سطر دارد (به ازای هر ماه) و 2 ستون (نام ماه و جمع حقوق پرداختی). اینجا است که کوئری SQL آن اصلا شکل قشنگی پیدا نمی‌کند که هیچ (چون ماه‌های تاریخ میلادی تطابقی با ماه‌های شمسی ندارد)، بسیار هم غیربهرینه می‌شود. به همین جهت یکی از سربارهایی که می‌شود از آن چشم پوشی کرد، نگهداری تاریخ شمسی و میلادی با هم است.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۵/۰۱ ۱۷:۲۵:۵۹

- بله. هر بار در طی حلقه شما CPU Usage یک هسته رو دریافت می‌کنید. یک روش دیگر هم اینجا ذکر شده: [\(+\)](#). که باید دوبار اطلاعات دریافت شود و بعد بین آن‌ها هم یک sleep کوچک نیاز است.
- لیست تمام پروسه‌های ویندوز با این متد قابل دریافت است: Process.GetProcesses (در فضای نام System.Diagnostics) که یک سری اطلاعات در مورد TotalProcessorTime دارد.
همچنین این پروژه هم در راستای مطلب جاری است:
[How to get CPU usage of processes and threads](#)
البته این پروژه اطلاعاتی در مورد stack trace تردها (مثل روش فوق) نمی‌دهد فقط یک سری اطلاعات کلی است.

نویسنده: Nima
تاریخ: ۱۳۹۰/۰۵/۰۲ ۰۸:۲۰:۲۲

با سلام مجدد جناب نصیری
با تشکر از وقتی که گذاشتین باید عرض کنم من این لینک (Stackoverflow) رو موقع سرچ دیدم ولی انگار این نسخه از دات نت که کد رو بر اساسش نوشتن با نسخه 2010 تفاوت‌های زیادی داره چون اصلا کدها رو نمیشناسه .

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۵/۰۲ ۰۹:۰۳:۲۴

جهت کامپایل سورس "How to get CPU usage of processes and threads" ، در پروژه CpuUsageAPI ، پوشه properties آن خالی است. یک سری فایل نیست. مهم هم نیستند. این‌ها را از پروژه حذف کنید. تست شد با دات نت 4 هم کامپایل می‌شود.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۶/۲۳ ۲۱:۱۴:۱۵

نگارش بعدی ASP.NET امکانات بیشتری روی کنترل CPU usage دارد:
[CPU Throttling for ASP.NET Asynchronous Scenarios in vNext](#)