

زمانیکه اولین نگارش ASP.NET حدود 10 سال قبل منتشر شد، تنها سیستم عاملی که از آن پشتیبانی می‌کرد، ویندوز سرور 2000 بود، تنها پروسه‌ای اجرایی آن aspnet_wp نام داشت و تنها معماری پشتیبانی شده هم X86 بود. به پروسه‌ای aspnet_wp محدودیت مصرف حافظه‌ای اعمال شده بود که در حین آغاز آن بر اساس مقدار قابل تغییر `processModel memoryLimit` محاسبه و اعمال می‌شد (تعریف شده در فایل ماشین کانفیگ). این عدد به صورت درصدی از ظرفیت RAM فیزیکی سیستم، قابل تعریف و به صورت پیش فرض به 60 درصد تنظیم شده بود. به این ترتیب این پروسه مجاز نبود تا تمام حافظه‌ی فیزیکی مهیا را مصرف کند و در صورت وجود نشستی حافظه‌ای در برنامه‌ای خاص، این پروسه امکان بازیابی مجدد حافظه را پیدا می‌کرد (recycling). همچنین یک مورد دیگر را هم باید در نظر داشت و آن هم وجود قابلیت است به نام ASP.NET Cache است که امکان ذخیره سازی مقادیر اشیاء را در حافظه‌ی مصرفی این پروسه مهیا می‌سازد. هر زمان که میزان این حافظه‌ی مصرفی به حد نزدیکی از محدودیت تعریف شده برسد، این پروسه به صورت خودکار شروع به حذف آن‌ها خواهد کرد.

محدودیت 60 درصدی تعریف شده، برای سیستم‌هایی با میزان RAM کم بسیار مفید بود اما در سیستم‌هایی با میزان RAM بیشتر، مثلاً 4 گیگ به 2.4GB حافظه مهیا (60 درصد حافظه فیزیکی سیستم) محدود می‌شد و همچنین باید در نظر داشت که میزان user mode virtual address space مهیا نیز تنها 2 گیگابایت بود. بنابراین هیچگاه استفاده مؤثری از تمام ظرفیت RAM مهیا صورت نمی‌گرفت و گاهی مشاهده می‌شد که یک برنامه تنها با مصرف 1.5GB RAM می‌توانست پیغام OutOfMemoryException را صادر کند. در این حالت مطابق بررسی‌های صورت گرفته مشخص شد که اگر مقدار `processModel memoryLimit` به حدود 800 مگابایت تنظیم شود، بهترین عملکرد را برای سیستم‌های مختلف می‌توان مشاهده کرد.

با ارائه‌ی ویندوز سرور 2003 و همچنین ارائه‌ی نسخه‌ی 1.1 دات نت فریم ورک و ASP.NET، این وضعیت تغییر کرد. پروسه‌ی جدید در اینجا w3wp نام دارد و این پروسه تعاریف مرتبط با محدودیت حافظه‌ی خود را از تنظیمات IIS دریافت می‌کند (قسمت Maximum Used Memory در برگه‌ی Recycling مربوط به خواص Application Pool مرتبط). متأسفانه این عدد به صورت پیش فرض محدودیتی ندارد و به ظاهر برنامه مجاز است تا حد امکان از حافظه‌ی مهیا استفاده کند. به همین جهت یکی از مواردی را که باید در نظر داشت، مقدار دهی Maximum Used Memory ذکر شده است. خصوصاً اینکه در نگارش 1.1، تنظیمات میزان مصرف RAM مرتبط با ASP.NET Cache نیز با برنامه یکی است.

در نگارش 2.0 دات نت فریم ورک، تنظیمات مرتبط با ASP.NET cache از تنظیمات میزان مصرفی یک برنامه‌ی ASP.NET جدا شد و این مورد توسط قسمت `cache privateBytesLimit` قابل تنظیم و مدیریت است (در فایل IIS Metabase و همچنین فایل web.config برنامه).

نکته!

اگر `process memory limit` و همچنین `cache memory limit` را تنظیم نکنید، باز به همان عدد 60 درصد سابق بازخواهیم گشت و این مورد به صورت خودکار توسط IIS محاسبه و اعمال می‌شود. البته محدودیت ذکر شده برای پروسه‌های 64 بیتی در این حالت بسیار بهتر خواهد بود. اگر هر دوی این‌ها را تنظیم کنید، عدد حداقل بکارگرفته شده، مبنای کار خواهد بود و اگر تنها یکی را تنظیم کنید، این عدد به هر دو حالت اعمال می‌گردد. برای بررسی بهتر می‌توان به مقدار `Cache.EffectivePrivateBytesLimit` و `Cache.EffectivePercentagePhysicalMemoryLimit` مراجعه کرد.

و ... اکنون بهتر می‌توانید به این سؤال پاسخ دهید که «سرور ما بیشتر از 4 گیگ رم دارد و برنامه‌ی ASP.NET من الان فقط 850 مگ رم مصرف کرده (که البته این هم نشانی از عدم dispose صحیح منابع است یا عدم تعیین تقدم و تاخر و زمان منقضی شدن، حین تعریف اشیاء کش)، اما پیغام out of memory exception را دریافت می‌کنم. چرا؟!»

بنابراین ایجاد یک [Application pool](#) جدید به ازای هر برنامه‌ی ASP.NET امری است بسیار مهم زیرا:

- به این ترتیب هر برنامه‌ی ASP.NET در پروسه‌ای ایزوله از پروسه‌ی دیگر اجرا خواهد شد (این مساله از لحاظ امنیتی هم بسیار مهم است). در اینجا هر برنامه، از پروسه‌ی w3wp.exe مجزای خاص خود استفاده خواهد کرد (شبیه به مرورگرهایی که هر tab را در یک پروسه جدید اجرا می‌کنند).
- اگر پروسه‌ای به حد بالای مصرف حافظه‌ی خود رسید با تنظیمات انجام شده در قسمت recycling مرتبط با Application pool اختصاصی آن، به صورت خودکار کار بازیابی حافظه صورت می‌گیرد و این امر بر روی سایر برنامه‌ها تاثیر نخواهد داشت (کاربران سایر برنامه‌ها مدام شکایت نمی‌کنند که سشن‌ها پرید. کش خالی شد. زیرا در حالت وجود application pool اختصاصی به ازای هر برنامه، مدیریت حافظه برنامه‌ها از هم ایزوله خواهند بود)
- کرش صورت گرفته در یک برنامه به دلیل عدم مدیریت خطاها، بر روی سایر برنامه‌ها تاثیر منفی نخواهد گذاشت. (زمانیکه ASP.NET worker process به دلیل استثنایی مدیریت نشده خاتمه یابد بلافاصله و به صورت خودکار مجدداً «وهله‌ی دیگری» از آن شروع به کار خواهد کرد؛ یعنی تمام سشن‌های قبلی از بین خواهند رفت؛ که در صورت ایزوله سازی ذکر شده، سایر برنامه‌ها در امان خواهند ماند؛ چون در پروسه ایزوله‌ی خود مشغول به کار هستند)
- با وجود application pool اختصاصی به ازای هر برنامه، می‌توان برای سایت‌های کم ترافیک و پرترافیک، زمان‌های recycling متفاوتی را اعمال کرد. به این ترتیب مدیریت حافظه‌ی بهتری قابل پیاده سازی می‌باشد. همچنین در این حالت می‌توان مشخص کرد کدام سایت از تعداد worker process بیشتر یا کمتری استفاده کند.
- کاربری که پروسه‌ی ASP.NET تحت آن اجرا می‌شود نیز همینجا تعریف می‌گردد. بنابراین به این ترتیب می‌توان به برنامه‌ای دسترسی بیشتر و یا کمتر داد، بدون تاثیر گذاری بر روی سایر برنامه‌های موجود.

نتیجه گیری:

- از IIS استفاده می‌کنید؟ آیا می‌دانید Application pool چیست؟
- آیا می‌دانید در صورت عدم مقدار دهی پارامترهای حافظه‌ی یک Application pool، به صورت پیش فرض چند درصد از حافظه‌ی فیزیکی مهیا در اختیار شما است؟

برای مطالعه بیشتر:

[CLR processModel memoryLimit](#)
[Some history on the ASP.NET cache memory limits](#)
[Managing Application Pools in IIS 7](#)

نظرات خوانندگان

نویسنده: محسن
تاریخ: ۱۳۹۰/۰۵/۰۵ ۲۳:۵۹:۱۰

مقاله ی مفیدی نوشتید. جای خالی اینجور مقالات فارسی توی اینترنت احساس میشه. خسته نباشید. دست شما درد نکنه.

نویسنده: Rab Raby
تاریخ: ۱۳۹۰/۰۵/۰۶ ۱۵:۱۹:۳۰

بسیار مهم و مفید بود مثل همیشه .

نویسنده: Amin
تاریخ: ۱۳۹۰/۰۵/۱۰ ۱۰:۰۸:۳۸

سلام آقای نصیری
ممنون از مطلب مفیدتون.
یه سوال: اگر خود این AppPool ها از لحاظ حافظه و CPU به حالتی برسند که بشه گفت به سقف چسبیدن، روشی برای رفع این مشکل وجود دارد؟ ما الان یه چنین مشکلی داریم. من مسئول این کار نیستم و زیاد در جریانش نیستم اما چون این مشکل رو دیدم می خواستم بدونم چه طور میشه این مشکل رو حل کرد.

ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۵/۱۰ ۱۰:۴۵:۰۸

- در مورد بررسی علت بالا بودن CPU Usage اینجا توضیح دادم و روش دیباگ ذکر شده. به این ترتیب می‌تونید نام متدهای مشکل ساز رو دقیقاً پیدا کنید : [\(+\)](#)
- ضمناً یکی از تنظیمات App pool ، مرتبط است با تعیین دقیقاً cpu limit مورد استفاده: [\(+\)](#) البته این تنظیمات مرتبط است به IIS 7 ولی در IIS 6 هم وجود دارد و فرقی نمی‌کند. یعنی به صورت خلاصه می‌تونید تعیین کنید که به سقف نرسند. (در مورد تنظیمات حافظه هم به همین صورت)

نویسنده: Amin
تاریخ: ۱۳۹۰/۰۵/۱۱ ۰۸:۵۵:۵۹

ممنون از راهنماییتون.

نویسنده: Sniper_528
تاریخ: ۱۳۹۰/۰۵/۱۳ ۲۲:۰۸:۰۸

کاش برای SQL Server هم چیزی مثل recycling وجود داشت یعنی هر وقت میزان استفاده اون از RAM به یه حدی می رسید recycle میشد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۵/۱۴ ۰۰:۵۷:۰۳

نه. این خوب نیست؛ چون کش اس کیوال سرور execution plan های زیادی داخل هست و خیلی مسایل دیگر (یعنی این مصرف صحیح حافظه هست نه نشتی حافظه).

در کل می‌شود برای اس کیوال سرور محدودیت حافظه گذاشت؛ در موردش قبلاً مطلب نوشتم در سایت هست : [\(+\)](#)
ضمناً یک سری دستور برای خالی کردن این کش‌ها هم هست: [\(+\)](#) ؛ ولی باز هم توصیه نمی‌شود چون این‌ها نشتی حافظه نیست.

نویسنده: Sniper_528
تاریخ: ۱۳۹۰/۰۵/۱۷ ۰۵:۱۳:۲۹

ممنون

من در سرورم با رم 2 گیگ، IIS، DNS Server و SQL Server رو با هم دارم max memory رو چی پیشنهاد می کنید برای اینها؟
سایت هم بازدید روزانه حدود 400 تا رو داره.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۵/۱۷ ۰۸:۱۳:۳۸

GB 1.2

نویسنده: Sniper_528
تاریخ: ۱۳۹۰/۰۵/۱۷ ۱۴:۱۸:۲۳

در حال حاضر 1.4 گیگ از رم اشغاله که 200 مگ مربوط به اس کیو ال میشه
پس گزاشتمش رو 500 مگ

نویسنده: Nima
تاریخ: ۱۳۹۰/۰۵/۲۲ ۱۳:۱۱:۰۰

با سلام آقای نصیری

در مورد سشن ها چطور؟ آیا محدودیتی برای حجم سشن ها هم هست؟ آیا این محدودیت قابل برداشتن هست؟ فضای سشن ها رو IIS مدیریت میکنه یا Asp.Net ؟ اگر مقدار حافظه مورد نیاز سشن زیاد باشه چه اتفاقی میفته؟ با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۵/۲۲ ۱۳:۳۸:۴۴

بستگی داره Session state به چه صورتی تنظیم شده باشد. می شود آن را طوری تنظیم کرد که در اس کیوال سرور هم حتی ذخیره شود. حالت InProc آن یعنی همان توضیحات فوق و تمام تنظیمات app pool به آن اعمال می شود. اطلاعات بیشتر:

[Session State](#)

نویسنده: میلاد حسینی
تاریخ: ۱۳۹۱/۰۶/۲۱ ۱۳:۲۵

با سلام؛

مشکلی که من دارم نمیدانم مربوط میشود به مدیریت حافظه یا موضوعی دیگر
من یک وب سایت کوچک دارم که با تکنولوژی های زیر ایجاد شده:

ASP.Net MVC 4 , Entity Framework 4 , SQL CE

آن را بر روی یک ویندوز سرور 2012 نسخه دیتاستر نصب کردم

سرور : 2GB Ram و CPU Dual Core 1.8

غیر از این سایت هیچ سایت دیگری بر روی این سرور میزبانی نشده است.

صفحات با سرعت نسبتاً خوبی باز میشوند، اما به هر شکلی iis را تنظیم میکنم، اگر پس از 2 یا 3 دقیقه درخواستی به سمت سرور ارسال نگردد، برنامه از حافظه خارج میشود. اگر درخواستی برای مشاهده صفحه به سرور ارسال شود 15 تا 20 ثانیه طول میکشد تا دوباره کامپایل انجام شود و صفحه درخواستی نمایش یابد.

تصویر تنظیمات Application Pool

General	
.NET Framework Version	v4.0
Enable 32-Bit Applications	False
Managed Pipeline Mode	Integrated
Name	Hand
Queue Length	1000
Start Automatically	True
Start Mode	OnDemand
CPU	
Limit (1/1000 of %)	1000
Limit Action	NoAction
Limit Interval (minutes)	5
Processor Affinity Enabled	True
Processor Affinity Mask	4294967295
Processor Affinity Mask (64-bit option)	4294967295
Process Model	
Generate Process Model Event Log Entry	
Idle Time-out Reached	False
Identity	LocalSystem
Idle Time-out (minutes)	0
Load User Profile	False
Maximum Worker Processes	4
Ping Enabled	True
Ping Maximum Response Time (seconds)	90
Ping Period (seconds)	30
Shutdown Time Limit (seconds)	90
Startup Time Limit (seconds)	90
Process Orphaning	
Enabled	False
Executable	
Executable Parameters	
Rapid-Fail Protection	
"Service Unavailable" Response Type	HttpLevel
Enabled	True
Failure Interval (minutes)	5
Maximum Failures	5
Shutdown Executable	
Shutdown Executable Parameters	
Recycling	
Disable Overlapped Recycle	False
Disable Recycling for Configuration Change:	False
Generate Recycle Event Log Entry	
Private Memory Limit (KB)	0
Regular Time Interval (minutes)	0
Request Limit	0
Specific Times	
Virtual Memory Limit (KB)	TimeSpan[] Array
	0

پ.ن: لطفاً اگر امکان دارد بهترین تنظیمات را برای سروری که فقط به یک سایت می‌خواهد سرویس دهد عنوان کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳:۵۷ ۱۳۹۱/۰۶/۲۱

به فرض اینکه تنظیمات specific times فوق که در اینجا مشخص نیست صحیح است (مثلاً تنظیم شده به 2 بامداد)، [این مطلب](#) بیشتر مرتبط است به کار شما.

نویسنده: داود
تاریخ: ۱۳:۱۹ ۱۳۹۲/۰۸/۱۴

سلام؛ ما به سایت داریم که در روز حدود 1000 تا کاربر دارد. در قسمت admin، آپلود فایل هم زیاد داریم.
RAM وب سرور هم 8GB هست.

WinServer 2008 32bit - CPU: Xeon 5160 3GHz

IIS 7

تقریباً روزی یکی دوبار شاید هم هر دو سه روز به بار نیاز به recycle داشته باشیم.

پیشنهاد می‌کنید Virtual Memory Usage و Private Memory Usage چند باشد تا کمترین نیاز رو برای recycle داشته باشیم؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۵۲ ۱۳۹۲/۰۸/۱۴

- سرور 32 بیتی نمی‌تونه از حداکثر میزان RAM سرور شما (بیشتر از 2GB) نهایت استفاده رو انجام بده. [تمهیداتی](#) هم در این زمینه هست ولی ... بهتره به یک سرور 64 بیتی کوچ کنید. بدون *این تمهیدات*، میزان حافظه مهابی جهت یک پروسه 32 بیتی به اندازه address space آن یعنی 2GB محدود است.

- همچنین باید [کش کردن](#) اطلاعات رو فعال کنید و اجازه بدید IIS بجای برنامه این مسایل رو راساً مدیریت کنه؛ یا از یک کش سرور مجزا استفاده کنید.