

# نگاهی به ویندوز سرور ۲۰۰۳

## IIS 6.0

بهر روز نوعی پور

اشاره

یکی از مهمترین قسمت‌های بهبود یافته در سیستم عامل ویندوز سرور ۲۰۰۳، بخش سرویس اینترنت آن است که عمدتاً در برگرفته وب سرور این سیستم عامل شبکه‌ای است. IIS 6.0 در ویندوز ۲۰۰۳ تقریباً از نو طراحی شده و تغییرات بسیار زیادی به خود دیده است. نسخه اخیر IIS بسیار امن تر و پایدارتر است و کارایی بهتری دارد. IIS 6.0 به فناوری دات نت مایکروسافت مجهز است و از ASP.NET، سرویس‌های وب و فناوری XML پشتیبانی کاملی به عمل می‌آورد. این مقاله می‌تواند هم برای سرپرستان شبکه (Administrators) و هم برنامه‌نویسان ASP.NET سودمند باشد.

### معماری جدید IIS

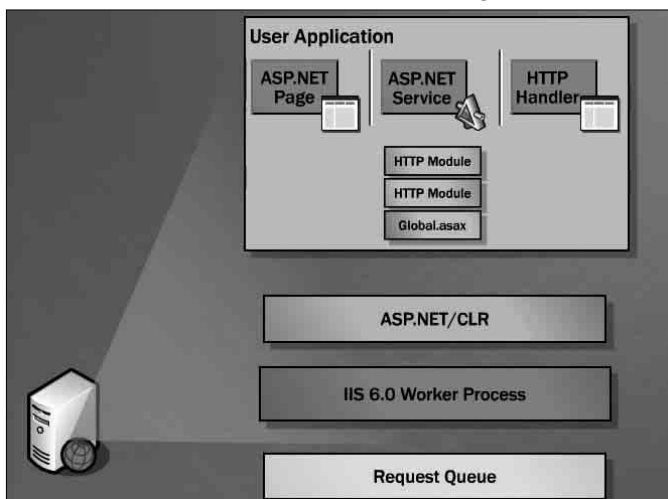
ساختار داخلی IIS در نسخه ششم آن به کلی تغییر کرده است. در این نسخه، IIS به طور عمده به دو بخش تقسیم می‌شود: بخش Kernel (که جزئی از هسته سیستم عامل است) و بخش Application Pools (که دربرگیرنده نرم‌افزارهای در حال اجرا و تحت وب می‌باشد). هنگامی که درخواست مشاهده یک صفحه وب به IIS 6.0 می‌رسد کل فرآیند را می‌توان به زبان ساده این‌طور توضیح داد: ابتدا لایه Network Stack در بخش کرنل درخواست مربوطه را به عنصر HTTP.SYS که یکی از مهمترین قسمت‌های IIS است هدایت می‌کند. در آنجا IIS بررسی می‌کند ببیند آیا صفحه وب مورد نظر قبلاً روی Cache سیستم قرار گرفته است یا خیر. اگر این صفحه در داخل Cache موجود باشد، اصولاً فرآیند جلوتر از این نمی‌رود و همان‌جا خیلی سریع، پاسخ به Network Stack ارسال می‌شود. اما اگر URL صفحه مورد نظر در cache پیدا نشود، این درخواست در لایه Request Queue از بخش کرنل منتظر می‌ماند تا IIS آن را به یک Application Pool مناسب هدایت کند. در آنجا تازه داستان این مقاله آغاز می‌شود!

شکل ۲ نمای ساده‌ای از معماری یک App. Pool را نشان می‌دهد. لایه زیر بنای این بخش را چیزی به نام Worker Process تشکیل می‌دهد. به زبان ساده این‌طور می‌توان توضیح داد که IIS 6.0 از مکانیزم فرآیندهای ویندوز (۳) برای ایزوله کردن فضای اجرای برنامه‌های کاربردی تحت وب در حافظه و روی پردازنده استفاده می‌کند. این فرآیندها را Worker Process می‌نامند (۴). این لایه سه کار مهم انجام می‌دهد: نخست این‌که بررسی می‌کند ببیند آیا لازم است برای مشاهده یک صفحه خاص، باید عملیات Authentication

IIS 6.0 در ویندوز سرور ۲۰۰۳ بخشی از Application Server Role محسوب می‌شود (۱) اگر پیش از این تجربه کار با IIS 5.0 را داشته‌اید، پس از نصب App. Server تغییرات بسیار زیادی را در ساختار، امکانات و توانایی‌های IIS مشاهده خواهید کرد و اگر به تازگی به جمع برنامه‌نویسان ASP.NET پیوسته‌اید، از قابلیت‌های متعدد و سودمندی که در این سرویس مهم سیستم عامل تعبیه شده است، راضی خواهید بود. شکل ۱ نمای از App. Server در ویندوز ۲۰۰۳ را نمایش می‌دهد. برای باز کردن کنسول مدیریتی App. Server کافی است پنجره Manage Your Server را از شاخه Administrative Tools در فهرست نرم‌افزارهای نصب شده روی سیستم عامل، باز کنید و گزینه Manage this application server را انتخاب کنید. البته کنسول مدیریتی IIS به طور مستقیم نیز (مانند نسخه‌های گذشته) در دسترس قرار دارد.

App. Server به معنایی که دقیقاً از این اصطلاح برداشت می‌شود چیز جدیدی در سیستم عامل ویندوز محسوب می‌شود. با در نظر گرفتن همه قابلیت‌ها و امکانات این مجموعه، از جمله سرویس Assembly Cache در پلاگرم دات نت، سرویس IIS 6.0 و سرویس Componentها (۲)، می‌توان App. Server ویندوز ۲۰۰۳ را در مقایسه با سایر محصولات رقیب در بازار، یک App. Server واقعی و جدی به حساب آورد که در کنار نرم‌افزار بانک اطلاعاتی SQL Server مجموعه نسبتاً کامل و قدرتمندی را فراهم می‌سازد که برای پیاده‌سازی انواع برنامه‌های کاربردی تحت وب، از ساده‌ترین تا پیچیده‌ترین آن‌ها آماده است. خبرهای خوب زیادی برای برنامه‌نویسان ASP سابق و ASP.NET جدید وجود دارد، از جمله این‌که IIS 6.0 واقعا محکم‌تر و سریع‌تر از نسخه قبلی است و در این مقاله به طور اجمالی نشان خواهم داد چرا.

شکل ۲- معماری ساده شده Application Pool



شکل ۱- نمای از Application Server در ویندوز ۲۰۰۳



(احراز هویت) و Authorization (تعیین اختیارات کاربر) انجام شود یا خیر.

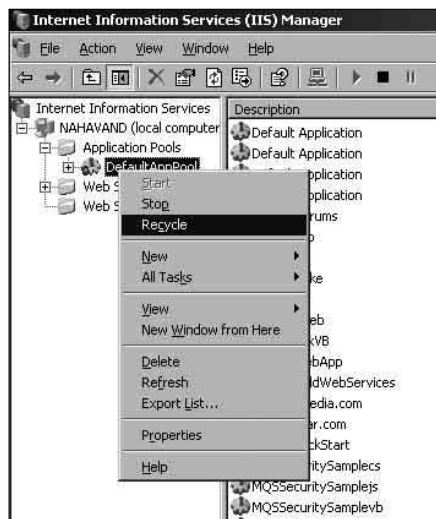
سپس کنترل می‌کند که نوع صفحه درخواست شده استاتیک است یا دینامیک. اگر صفحه HTML استاتیک درخواست شده باشد، خودش آن را بارگذاری و برای کاربر متقاضی ارسال می‌کند. اما اگر صفحه مورد تقاضا دینامیک باشد، بررسی می‌کند که نوع صفحه مورد تقاضا چیست و متناسب با آن، سرویس میزبان مربوطه را احضار و ادامه رسیدگی به تقاضا را به او واگذار می‌کند. این میزبان ممکن است ASP, ISAPI Filter, ASP.NET یا یک web service باشد. اگر صفحه تقاضا شده از نوع ASP.NET باشد، رسیدگی به آن به لایه بعدی معماری IIS 6.0، یعنی سرویس میزبانی ASP.NET واگذار می‌شود. این میزبان جزوی از لایه CLR در پلاتفرم دات نت است. (۵)

### استحکام و پایایی (Reliability) در IIS 6.0

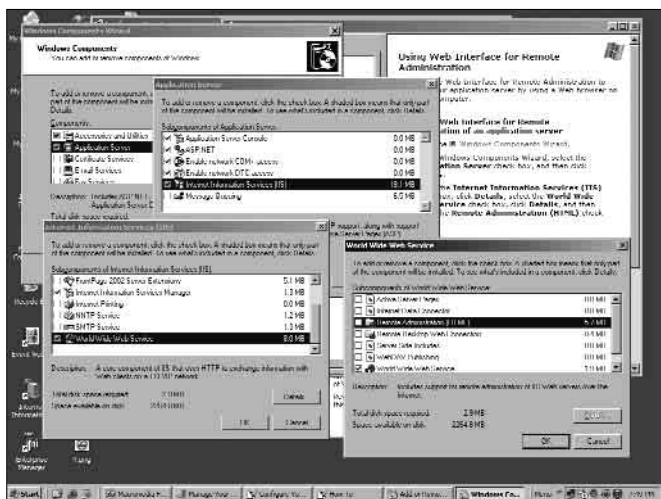
اگر به معماری IIS 6.0 و ساختار منطقی آن دقت کنید، دلیل این ادعا که IIS جدید از نسخه قبلی آن پایدارتر است کاملاً روشن است. در این معماری نوعی تقسیم کار میان IIS و چارچوب دات نت (۶) صورت گرفته است. IIS عمدتاً مسوول ارتباط دادن موتور ASP.NET با پروتکل شبکه‌ای TCP/IP است، در حالی که هسته پردازش ASP.NET در پلاتفرم دات نت کاری با این فرآیندهای شبکه‌ای ندارد. به تعبیر کارشناسان نرم افزار، IIS مسوول فراهم ساختن Connectivity است حال آن‌که ASP.NET مدل برنامه‌نویسی (۷) را مشخص می‌کند. بنابراین برخلاف آن‌چه که در IIS 5.0 وجود داشت، IIS 6.0 و نه ASP.NET است که فرآیندها و رسیدگی به تقاضاهای مشاهده صفحات دینامیک را اداره می‌کند (۸). در IIS 6.0، موتور ASP بیش از حد درگیر عملیات رسیدگی به تقاضاها و فرآیندها بود و همین مساله ایجاد دردسر می‌کرد.

بخش جالب و قدرتمند IIS 6.0 همانا سیستم Worker Process آن است. دانستن نحوه عمل این لایه از App. Pool به درک چگونگی کارکرد سرویس میزبانی صفحات وب در ویندوز ۲۰۰۳ کمک بسیار زیادی می‌کند. در معماری IIS 6.0 «نحوه عمل سیستم Worker Process» را «مدل فرآیند» (۹) نامیده‌اند. مدل فرآیند در IIS 6.0 از مکانیزم جدیدی استفاده می‌کند که علت اصلی استحکام و پایایی آن است. این مکانیزم اصطلاحاً Recycling نامیده می‌شود (۱۰). هنگامی که یک برنامه کاربردی تحت وب در حال اجراست، ممکن است به دلایل مختلفی روند کار آن مختل شود. مثلاً ممکن است حافظه کم بیاید یا «نشت حافظه» (۱۱) رخ دهد یا حتی برنامه ناگهان از سرویس‌دهی باز بماند (۱۲). در این لحظه،

### شکل ۴- می‌توان به صورت دستی مکانیزم Recycling را احضار کرد.



مکانیزم Recycling وارد صحنه می‌شود و بدون این‌که بازدیدکننده سایت بویی ببرد، برنامه معیوب را از حافظه اخراج و مجدداً نسخه تازه‌نفسی از آن را در حافظه بارگذاری می‌کند. در طول این عملیات، درخواست‌های بازدیدکنندگان سایت موقتاً در لایه Request Queue تلنبار می‌شوند و به نوبت در صف قرار می‌گیرند. اما این لحظه آن قدر کوتاه است که بازدیدکننده

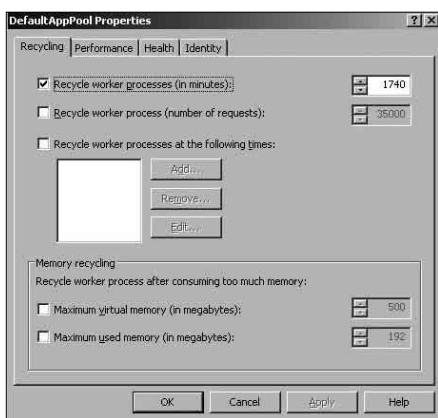


شکل ۳- نصب سفارشی اجزای IIS 6.0

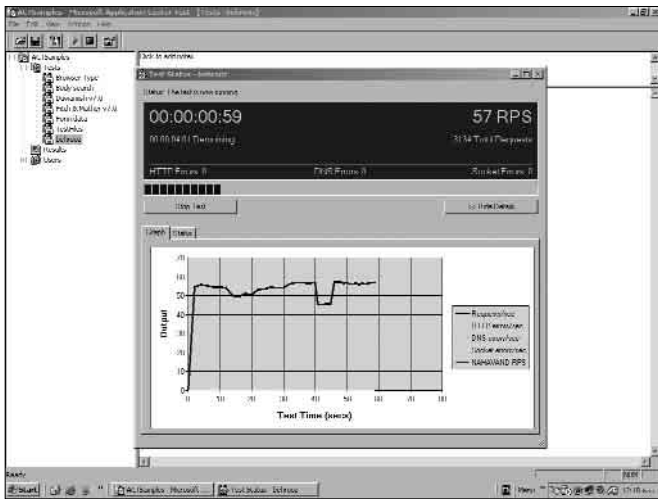
چیزی متوجه نمی‌شود. اگر برنامه‌نویس سایت یا مدیر شبکه کنجکاو باشد بداند چرا برنامه دچار مشکل شد و IIS عملیات Recycling را اجرا کرد، می‌تواند سیستم Recycling را در حالت orphan قرار دهد. در این صورت IIS پس از بارگذاری یک نسخه تازه از برنامه در حافظه، نسخه معیوب را همچنان نگه می‌دارد تا برای کنکاش و اشکال زدایی (۱۳) در دسترس باشد اما هیچ تقاضای جدیدی از سوی Request Queue برای آن ارسال نخواهد شد.

اما مکانیزم Recycling در IIS 6.0 چنان‌که ممکن است تصور کنید، چیز بدی که بهتر است از آن دوری شود، نیست. معماری ASP.NET و IIS 6.0 به گونه‌ای است که با استفاده از یک روش برنامه‌نویسی به خصوص که بعدتر به آن اشاره خواهیم کرد، می‌توانید کلیه مقادیر و متغیرهایی را که برنامه تا پیش از Recycling در حال کار با آن‌ها بوده است، نگه دارید. بنابراین برنامه کاربردی تحت وب شما Reset نمی‌شود، بلکه مجدداً بارگذاری می‌شود و به گردش در می‌آید. یک نتیجه مهم که در این جا به دست می‌آید آن است که دیگر نیازی نیست که مدیران و سرپرستان شبکه، سیستم IIS و حتی ویندوز سرور ۲۰۰۳ را به خاطر مشکلاتی که برای سایت‌های وب پیش می‌آید Restart کنند. IIS 6.0 طوری طراحی شده است که می‌توان هر گروه از سایت‌های وب و برنامه‌های کاربردی تحت وب را در یک App. Pool قرار داد. محدودیت چندانی برای تعداد App. Pool‌ها وجود ندارد. هر بار که یک برنامه کاربردی یا یک سایت وب در یک App. Pool دچار مشکل می‌شود می‌توان آن App. Pool را دوباره به گردش درآورد (شکل ۴). مختل شدن روند کار یک App. Pool هیچ تاثیری روی دیگران ندارد. این نکته احتمالاً برای صاحبان و گردانندگان شرکت‌های میزبانی وب بسیار مهم است. زیرا اگر سایت یک مشتری دچار مشکل شود (مثلاً هک شود یا حافظه کم بیآورد یا دچار اختلال شود و از کار


### شکل ۵- تنظیم و برنامه ریزی Recycling



بازماند) سایر سایت‌هایی که روی همان سرور در حال اجرا هستند متاثر نخواهند شد و نیازی هم به Restart کردن سرویس IIS یا سیستم‌عامل نیست. وقتی IIS را reset می‌کنیم، تمام درخواست‌های رسیده از سوی بازدیدکنندگان حذف و نادیده گرفته می‌شوند و تمام ارتباطات TCP/IP با



شکل ۶- آزمایش راندمان یک سایت روی IIS توسط برنامه ACT

یکی دیگر از ویژگی‌های مهم IIS6.0 افزایش امنیت نرم‌افزار و سیستم عامل در آن است. ASP.NET نسخه ۱/۱ و پلاتنوم دات‌نت، مستقل از IIS6.0 پاره‌ای از ویژگی‌های امنیتی جدید مانند Code Access Security را فراهم می‌آورد (۱۸). اما IIS 6.0 خود به تنهایی دارای قابلیت‌های امنیتی بهبود یافته‌ای است. سازندگان نسخه اخیر IIS هنگام طراحی معماری آن، مساله چگونگی برخورد با buffer overflow را مدنظر قرار داده‌اند و تاثیر این موضوع در استحکام IIS6.0 محسوس است. «مدل فرآیند» نیز که پیش از این شرح آن رفت، به دلیل بالا بردن پایایی و استحکام نرم‌افزار و قدرت پاسخگویی آن و نیز مدیریت بهتر حافظه، به امنیت IIS 6.0 افزوده است. در این نسخه از IIS هر فرآیند دارای یک ID منحصر به فرد است که انجام هر خطی از سورس کد برنامه تنها به واسطه کنترل این شماره شناسایی امکان پذیر است. بنابراین احتمال اجرای کدهای خطرناک و غیرمجاز بسیار کاهش می‌یابد. علاوه بر این، طراحان IIS 6.0 و ASP.NET، موتور پردازش کدهای برنامه را در لایه‌ای از سیستم عامل قرار داده‌اند که اصولاً ناتوان از اجرای برخی از فرمان‌های سطح پایین در سیستم عامل است تا به این ترتیب امکان اجرای فرمان‌های مخرب توسط هکرها کاهش یابد. 

### پی‌نوشت:

- (۱) برای کسب اطلاعات بیشتر درباره Server Roles در ویندوز ۲۰۰۳ به مقاله‌ای در همین رابطه که در شماره ۳۸ ماهنامه شبکه به چاپ رسیده است مراجعه کنید.
- (۲) Component Services شامل NET Utilities، سرویس COM+ و DCOM و سرویس تراکنش‌ها (Transactions) می‌باشد.
- (۳) Windows Processes
- (۴) هر Worker Process می‌تواند به صورت Multi-Thread باشد. هر Thread دسته‌ای از سیکل‌های کاری پردازنده کامپیوتر است که دستورات زبان ماشین یک برنامه نرم‌افزاری خاص را اجرا می‌کند. ریزپردازنده، Threadهای ده‌ها نرم‌افزار در حال اجرا را پشت سر هم قطار می‌کند و به طور چرخشی به همه آن‌ها رسیدگی می‌کند. به همین دلیل ویندوز یک سیستم عامل Multitask است و می‌تواند همزمان چند کار انجام دهد.
- (۵) برای مطالعه شرح کاملی از CLR و NET Framework به مقاله‌ای در همین رابطه که در شماره ۳۷ ماهنامه شبکه چاپ شده است مراجعه کنید.
- (۶) NET Framework
- (۷) Programming Model
- (۸) شاید این تشبیه جالب باشد: معماری IIS6.0 به یک نانوائی شبیه است. آن‌هایی که در صف ایستاده‌اند متقاضیان مشاهده یک صفحه وب هستند. کسی که

IIS قطع می‌شوند. ولی مکانیزم Recycling فقط فرآیند مربوط به همان App. Pool موردنظر را از نو به گردش در می‌آورد و حتی تقاضاهای رسیده برای مشاهده صفحات آن سایت نیز در صف قرار می‌گیرد تا در اسرع وقت رسیدگی شوند. مکانیزم recycling همزمان در دو حالت اتوماتیک و دستی کار می‌کند. IIS در دوره‌های زمانی معین سیگنال‌هایی به هر فرآیند می‌فرستد و انتظار دارد که در بازه زمانی معینی پاسخ ببیند. اگر پاسخی از جانب یک فرآیند نیاید، IIS می‌فهمد که آن فرآیند مشکل دارد و آن را دوباره به گردش در می‌آورد. Administrator می‌تواند با مراجعه به IIS خصوصیات مختلف recycling را تنظیم کند. با استفاده از این تنظیمات، می‌توان شروطی را برای IIS تعیین کرد که در صورت تحقق آن شروط، مکانیزم recycling خود به خود به کار می‌افتد. مثلاً می‌توان به IIS گفت اگر میزان استفاده یک سایت وب یا برنامه کاربردی تحت وب از ۲۰ مگابایت فراتر رفت، آن سایت یا برنامه را از نو بارگذاری کند. در واقع مکانیزم recycling نه تنها به صورت دستی on demand در دسترس است، بلکه قابل برنامه‌ریزی (۱۴) نیز هست. مثلاً می‌توان به IIS گفت هر دو ساعت یک بار فلان App. Pool را از نو به گردش در بیاور یا در صورتی که تعداد تقاضاهای رسیده به یک سایت X فراتر رفت، عمل recycling را انجام بده. (شکل ۵) بنابراین انعطاف‌پذیری بسیار زیاد در استفاده از «مدل فرآیند» در IIS6.0 وجود دارد. به همین دلیل بسیاری از سرپرستان شبکه‌ها و سایت‌ها تمایل به استفاده از مکانیزم recycling دارند تا نسبت به سلامتی و برپا بودن سایت‌ها و برنامه‌های خود اطمینان حاصل کنند. از این رو مایکروسافت به برنامه‌نویسان ASP.NET توصیه می‌کند که سایت و برنامه تحت وب خود را نسبت به همسان بودن با فرآیند recycling آزمایش کنند. اگر نتیجه آزمایش منفی بود، باید اصطلاحاً مدیریت حالت یا session state management در برنامه کاربردی اصلاح شود (۱۵).

### قابلیت‌های دیگر

یکی از مهمترین ویژگی‌های مشترک ASP.NET و IIS6.0، در سیستم cache است (۱۶) استفاده از cache در معماری جدید IIS تاثیر معجزه‌آسایی در راندمان سایت و برنامه‌های تحت وب دارد و بین ۲۰ و ۳۰ درصد کارایی و سرعت پاسخگویی نرم‌افزار را بالا می‌برد. مکانیزم cache در IIS6.0 باعث افت قابل ملاحظه بار روی پردازنده کامپیوتر می‌شود و به همین دلیل می‌توان تعداد بیشتری از سایت‌ها و سرویس‌ها را روی یک دستگاه برپا کرد. از این رو استفاده از App. Server در ویندوز ۲۰۰۳ می‌تواند به کاهش هزینه‌های یک سازمان بینجامد زیرا با مجتمع کردن چند برنامه کاربردی تحت وب روی یک سرور می‌توان از هزینه‌های نگهداری و نیز هزینه‌های سخت‌افزاری کاست (۱۷).

یک نرم‌افزار بسیار سودمند به نام Application Center Test همراه مجموعه ویژگی‌ها و ابزارهاست که عرضه می‌شود که قادر است سایت یا برنامه کاربردی شما را زیر فشار قرار دهد و محیط واقعی را شبیه‌سازی کند تا ببینید سایت شما در هر ثانیه چند تقاضا را می‌تواند پاسخ دهد.

به کمک این ابزار می‌توانید تاثیر مکانیزم cache بر راندمان و سرعت پاسخگویی سایت خود را بیازمایید. در شکل ۶ نمایی از این برنامه را ملاحظه می‌کنید که در حال آزمایش یک سایت روی دستگاه خانگی من است. این تصویر نشان می‌دهد که سرور خانگی مورد آزمایش قادر به پاسخگویی به حداکثر ۶۰ تقاضا در هر ثانیه است (که البته راندمان خوبی نیست) راندمان متوسط حدود ۲۰۰ تقاضا در هر ثانیه است. روی یک کامپیوتر خانگی سریع یا یک Laptop به شرط آن‌که برنامه نویسی سایت به خوبی انجام شده باشد، این راندمان تا ۵۰۰ هم ممکن است بالا رود. با استفاده از مکانیزم cache می‌توان این سرعت را تا ۷۰۰ افزایش داد. سخت‌افزارهایی مثل سرورهای آی بی ام یا اچ پی و مارک‌های دیگر راندمان بسیار بیشتری را ارائه می‌کنند. استفاده از سیستم cache در IIS6.0 روی آن‌ها به طور قابل ملاحظه‌ای قدرت پاسخگویی سایت را بالا می‌برد.

جلوی پیشخوان ایستاده و نان‌ها را به مشتریان می‌دهد IIS است. کسی که نان‌ها را از تنور در می‌آورد (ASP.NET Host) اهمیتی نمی‌دهد که ترتیب و توالی مردمی که در صف ایستاده‌اند، چگونه است. فروشنده نان، بسته به نوع تقاضا (HTML، ISAPI، ASP، ASP.NET) چند قالب خمیر، چند عدد نان تازه یا چند عدد نان که چند دقیقه پیش پخته شده‌اند (cache) را در اختیار مشتری قرار می‌دهد.

Process Model (۹)

(۱۰) در IIS6.0 مکانیزم Recycling چیزی جز Reloading یا بارگذاری مجدد

برنامه کاربردی تحت وب نیست.

Memory Leakage (۱۱)

Crash (۱۲)

Debugging (۱۳)

Scheduling (۱۴)

(۱۵) یک راه حل چنین است: از المان sessionstate در فایل web.config

سایت موردنظر استفاده کنید و مقدار mode را به جای Inproc در حالت

SQLServer قرار دهید. در این صورت حالت متغیرها پیش از عمل recycling در پایگاه داده‌های SQL Server ذخیره می‌شود و از دست نمی‌رود. البته باید از دستور <Serialization> ابتدای نام کلاس (شی) مورد نظر که حاوی متغیرهای دلخواه شماست استفاده کنید. با این دستور، شیئی که از روی آن کلاس ساخته شده، به هنگام ذخیره شدن در SQL Server به صورت XML در می‌آید تا قابل ذخیره شدن باشد. با این روش راندمان نرم‌افزار حدود ۳۰ درصد افت می‌کند اما می‌توان با استفاده از روش cache آن را جبران کرد.

(۱۶) برای استفاده از سیستم cache در ASP.NET باید به سند راهنمای ASP.NET

(مثلا در MSDN) مراجعه کنید و بخش مربوط به Page Directives و به طور

خاص <%@OutputCache%> را ملاحظه کنید.

(۱۷) این تکنیک در مدیریت فناوری اطلاعات اصطلاحاً consolidation نامیده می‌شود.

(۱۸) در شماره‌های آینده ماهنامه شبکه، مفصلاً درباره امنیت ویندوز ۲۰۰۳ و

پلا تفرم دات نت مطالبی خواهیم داشت.