

“Stop guessing, weet waar je probleem niet zit!”

HET PERFORMANCE CENTER VAN UNISYS ZOEKT DE GRENZEN OP

In het performance center van Unisys worden applicaties tot hun uiterste limieten belast. Al binnen een half uur worden de eerste bottlenecks geïdentificeerd, vervolgens begint de zoektocht naar de oorzaak waarna de performance geoptimaliseerd wordt en worden zoveel mogelijk tweaks toegepast om een applicatie optimaal te laten draaien.

Lees over de testwijzen en ervaringen van de heren die zijn besmet met het performancevirus.

W e zitten in het performance center van Unisys in Schiphol-Rijk waar achter glas de ES7000's en andere zware jongens rustig draaien. Deze machines worden normaal gesproken hard aan het werk gezet. Het performance center heeft ten doel om klanten te helpen bij het valideren en optimaliseren van de prestaties van hun applicaties. Vanuit heel Europa en ver daar buiten komen ontwikkelaars naar dit performance center om exact te weten waartoe hun applicaties in staat zijn en waar eventueel problemen zitten.

Henk van der Valk is manager van het performance center en heeft sinds jaren een grote voorkeur voor tunen en optimaliseren. Van der Valk: “In het performance center gaan we samen met klanten hun applicaties testen. Dit zijn in het algemeen mission-critical applicaties die nog niet in productie zijn. Eigenlijk komt het er op neer dat onze klanten binnen hun organisatie van deze applicaties een bepaalde responsetijd of workload willen garanderen. Wij valideren of dit haalbaar is. Om daar zeker van te zijn en risico's uit te sluiten kunnen wij hen helpen om applicaties te testen en de bottlenecks te verhelpen. We hebben in de afgelopen jaren vele soorten applicaties getest. Wat vaak blijkt is dat men de limieten van de eigen applicatie niet kent. We doen onder andere performancetests van applicaties die door softwarehuizen zijn gemaakt waarvan de klant een garantie wil wat betreft de responsetijd. Een dergelijk applicatie is bijvoorbeeld in staat om bij duizend gebruikers een responsetijd onder de seconde te bieden. Maar wat gebeurt er als het aantal gebruikers groeit naar tweeduizend? De meeste softwarehuizen hebben niet de middelen om dit te testen en kunnen zeker geen rapport tonen met daarin de limieten van de applicatie. We bouwen dan een testomgeving voor een de klant. Vaak blijkt dat negen van de tien keer de applicatie niet doet wat hij zou moeten doen. In de afgelopen zeven jaar is het slechts één keer voorgekomen dat een applicatie beter presteerde dan de specificaties. Dat was een .NET-toepassing voor het verwerken van creditcardgegevens, geschreven door twee dames uit Turkije.”

Limieten opzoeken

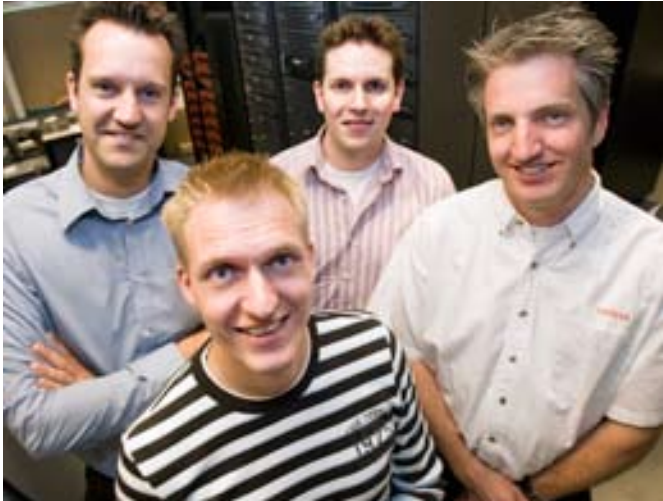
Wat betreft technologie, besturingssysteem of database kan alles getest worden in het performance center. Er staan verschillende zware systemen van 2-4 CPU's tot en met 64 core machines met 256GB aan RAM. Er zijn vele terabytes aan storage beschikbaar. Henk van der Valk: “Afhankelijk van de technologie van de applicatie, het besturingssysteem en de database wordt een team samengesteld. Zo'n team bestaat uit vijf tot twaalf experts.

Daarna wordt in nauw overleg met de klant een realistisch testplan gemaakt op basis van business requirements. Er moeten bijvoorbeeld met een applicatie een x aantal gebruikers tegelijk werken met een bepaalde responsetijd en een gegeven transactie-load. Meestal draaien we tests die een eindscenario nabootsen tot bijvoorbeeld het jaar 2012. We gaan dan die applicatie testen waarbij wordt aangenomen dat de hoeveelheid data per jaar met 30% groeit, dat er drie bedrijven worden overgenomen en dat er drie keer zoveel mensen met de applicatie werken. Dat scenario gaan we testen. We laten dan zien dat de doelstelling van 2012 haalbaar is. Mocht het niet lukken om die doelstelling te halen, dan helpen we bij het opsporen en verhelpen van de bottlenecks of geven advies hoe die zijn op te lossen. We zoeken hier echt de limieten van een applicatie op. Als een applicatie niet omvalt, hebben we ons werk niet goed gedaan. We testen net zolang tot we de grens van de applicatie hebben bereikt. Een gemiddelde test duurt ongeveer twee tot vier weken. Vaak zitten we op de tweede dag al de problemen op te lossen.”

Schaalbaarheid

Jeroen van Weeren is ruim vijf jaar in dienst bij Unisys en heeft verschillende functies bekleed op de gebieden infrastructuur, development en architectuur. Van Weeren: “Omdat hier het performance center is, word je automatisch als ontwikkelaar opgeleid en gestuurd om software te schrijven die schaalbaar is. In het .NET-team houden we ons vooral met businessapplicaties bezig. De focus ligt op alle lagen van een applicatie en ook op de performance van de database. Vaak wordt er gedacht, gooi er meer ijzer tegen aan en dat lost het performanceprobleem zich wel op. Het leuke is dat als we hier een benchmark draaien, we dat dan ook kunnen doen. Geef die applicatie maar eens tweemaal zoveel processoren en memory en kijk wat er gebeurt. Eigenlijk begint het dan pas.”

Henk Van der Valk: “Vaak wordt er bij performanceproblemen maar wat geroepen. Tijdens mijn presentatie op de SQL PASS was er een discussie over RAID 10 tegen RAID 5. Al snel riep iedereen dat RAID 5 langzamer is. Aan de hand van een testopstelling kon de throughput van beide getest worden. En dan blijkt dat RAID 10 inderdaad iets sneller schrijft en een throughput had van ongeveer 9 MB per seconde. Maar voor de grap had ik de cache van het SAN uitgezet. Op het moment dat je de cache weer aanzet heb je een throughput van wel 100 MB per seconde. Het is dus heel belangrijk wat het referentiekader is als je naar getallen kijkt.”



Afbeelding 1. V.l.n.r. Martijn van Schie, Jeroen van Weeren, Jeroen Kooiman en Henk van der Valk (Foto: Hans Oostrum fotografie)

De methodiek die Unisys bij het opsporen van performanceproblemen toepast is eigenlijk heel simpel. Het team begint op de lijst alles af te strepen wat juist niet het probleem is. Met andere woorden 'stop guessing'. Een performanceprobleem kan zich overal bevinden. Van der Valk: "Stap een is eigenlijk om alles te controleren of het werkt zoals het behoort te werken en per onderdeel een waardeoordeel te geven. Aan de hand van een kader kijken we of iets goed of slecht presteert. Al snel komen de bottlenecks dan vanzelf naar voren. Als ik op deze manier niet binnen een half uur een bottleneck kan vinden, heb ik zelf een probleem."

Tools

Jeroen Van Weeren: "We gebruiken diverse tools om de throughput van de verschillende onderdelen te meten. Er komt echter wel veel ervaring kijken bij het interpreteren van meetgegevens. Neem nu bijvoorbeeld de performancecounters van Windows zelf. De waarden van de verschillende counters variëren sterk. Bij de ene heb je een getal in de tienduizend en de andere keer een getal onder de tien. Wat is een goede waarde?"

Henk van der Valk vult zijn collega aan over de verschillende tools die gebruikt worden (zie kader red.). "Een aardig voorbeeld waar we tegen aanliepen bij het behalen van een nieuw ETL benchmark record met SQL Server 2008 is dat alle 64 CPU's op 100% stonden. Er ging (maar) 400 MByte per seconde door heen, terwijl je met één CPU 10 MByte haalt, een eenvoudig sommetje geeft 640 MByte bij 64 CPU's. De doelstelling was om meer dan 600 MB per seconde te halen. Maar als de CPU's op 100% staan, haal je dat niet. Een tool als Kernrate helpt dan om te achterhalen wat de processoren aan het doen zijn. Wat bleek is dat 40% van alle CPU-tijd werd gebruikt om de performancecounters van SQL Server bij te houden." Deze bottleneck is door de SQLServer developers in Redmond inmiddels verholpen.

Serius belasten

Martijn van Schie is werkzaam als consultant en developer in het .NET-team. "Het komt veelvuldig voor dat de performance niet het probleem is, maar dat er functionele problemen optreden onder load. Vanzelfsprekend heeft de klant zelf een functionele test gedaan, weliswaar met één gebruiker. Met die ene gebruiker loopt alles netjes, bijvoorbeeld bij een offertesysteem komt er mooi een pdfje uit. Met tien gebruikers echter blijkt dat de applicatie niet threadsafe is en gegevens van verschillende offertes in één pdf werden samengevoegd. Dit mag natuurlijk nooit gebeuren. Wat toch wel vaak voorkomt zijn problemen met de garbage collector. Dit komt voornamelijk door het niet goed omgaan met .NET-objecten. Denk hierbij bijvoorbeeld aan

het type object en de scope waarin een object leeft. Hoe langer je een object in leven houdt, hoe lastiger het wordt voor de garbage collector om deze op te ruimen. We zijn applicaties tegengekomen waar de processor zelfs tot 70% van zijn tijd bezig was met garbage collection. Toch is dit eenvoudig te voorkomen. Door als ontwikkelaar naar specifieke .NET performance counters (onder andere .NET CLR Memory) te kijken kun je al even snel een goed beeld krijgen van de life-cycle van de objecten in de applicatie. Een tool zoals CLR Profiler kan je hierbij helpen.

In weer andere situaties kan het zijn dat het niet lukt om de applicatie de processoren maximaal te belasten. Dit kan verschillende oorzaken hebben; bijvoorbeeld locking of wachten op langlopende webservice calls. Hierdoor kan er queuing in IIS optreden. De requests worden door IIS wel geaccepteerd, maar worden niet afgehandeld door de worker-processen. Om dit soort problemen vroegtijdig te ontdekken heb je echt geen spannende

Hulpmiddelen die Unisys gebruikt bij de analyse van performance:

Visual Studio Performance Analysis - Deze tool gebruik je voor onder andere de volgende doeleinden: identificeren en analyseren van objectallocatie / van langlopende routines en het analyseren van geheugengebruik. Voor meer informatie: msdn.microsoft.com/en-us/library/z9z62c29.aspx

PerfMon - Hiermee kun je diverse counters analyseren en monitoren van processen en de systeemresources.

CLR Profiler - The CLR Profiler bevat een aantal handige grafieken met betrekking tot geheugen allocatie, waaronder een histogram van de gealloceerde types, allocatie- and callgraphs, en een time line die aangeeft welke Garbage Collections er zijn van de verschillende generaties. Download: <http://www.microsoft.com/downloads/>

Sysinternals Process Monitor - Monitoren van processen en de daarbij behorende handles. Voor meer informatie en download: www.sysinternals.com

Sysinternals Process Explorer - Monitoren wat een proces doet op het systeem met betrekking tot Disk I/O, Memory, Threads et cetera. Voor meer informatie en download: www.sysinternals.com

Windows Performance Toolkit (xperf) - Voor het meten en grafisch weergeven van Event Tracing for Windows (ETW) events. Voor meer informatie en download: www.microsoft.com/whdc/system/sysperf/perftools.mspx

SQL Server 2008 - Xevents - Extended Events (Xevents) is een nieuwe algemene event-handler. Toont hoe Microsoft in de toekomst event handling/monitoring gaat aanbieden. Voor een demo van Extended Events zie: www.microsoft.com/sqlserver/2008/en/us/demos.aspx

Debug Diagnostic Tool v1.1 - Een hulpmiddel bij performanceproblemen en memory-pressure.

Dit is een aanrader voor .NET-programmeurs die zich willen verdiepen in de performance van het .NET Framework in zijn algemeen. Voor meer informatie zie: support.microsoft.com/?id=921464 en voor de download: www.microsoft.com/downloads/details.aspx?FamilyID=28bd5941-c458-46f1-b24d-f60151d875a3

Kernrate + Viewer (KrView) - Geeft een visuele representatie van de kernel-/user-mode van CPU-gebruik. Kernrate is een onderdeel van de vWindows2003 resource kit. Voor meer informatie en download: www.microsoft.com/downloads/details.aspx?FamilyID=d6e95259-8d9d-4c22-89c4-fad382eddc1

hardware nodig, maar je moet wel zorgen voor een representatieve belasting door meerdere concurrent gebruikers te simuleren. Kortom, er zijn vele situaties waar het fout kan gaan. Alleen door een applicatie serieus te belasten kom je er echt achter.”

Verschillende tests

Henk van der Valk: “We werken ook nauw samen met verschillende Microsoft-teams uit Redmond om producten flink aan de tand te voelen voordat ze gereleased worden. We voeren gezamenlijk allerlei performancetests uit, met verrassende resultaten. Een mooi recent voorbeeld is dat we weer een mythe hebben ontzenuwd; over het gebruik van meerdere filegroups in SQL Server. Daar zijn lange discussies over gevoerd in de afgelopen jaren. We kwamen er achter dat bij het meerdere malen draaien van dezelfde test de resultaten steeds verschillend waren. We zijn gaan zoeken en het bleek dat de I/O blocksize van de writes naar de disks variabel is wanneer je slechts één filegroup gebruikt. Als je er meer gebruikt is de blocksize fixed. Bij één filegroup varieert de I/O-size tussen de 64 KB en 265 KB. Bij een configuratie van één file per filegroup is die altijd 265 KB. Dat kan op een run van dertig minuten al snel tien minuten schelen. Dat wilden ze in eerste instantie bij het SQL Server-team niet in geloven totdat we de harde bewijzen gaven. De meeste bedrijven plannen Performance testen aan het eind van een developmenttraject en worden ten tijde van de acceptatietest vaak pas de eerste performancetests gedraaid, dit is eigenlijk niet het juiste moment. Gelukkig worden we steeds vaker al aan het begin van een developmenttraject betrokken zodat er nog volop mogelijkheden zijn om vroegtijdig performance aanpassingen door te voeren. De eerste bèta's van SQL Server 2005 hebben we al in september 2004 getest. Zo hebben we onder andere problemen ontdekt in SQL Server Integration Services.

Die liep gewoon vast als je een bestand inlas dat groter was dan 950 MB. Een half jaar later kreeg ik een e-mail met de vraag om opnieuw dezelfde tests te draaien, want het hele onderdeel was opnieuw geschreven. Nu kun je zonder problemen bestanden van 20Gigabyte inlezen, en ook 10-20 maal sneller!”

Dedicated machines

“We testen gemiddeld met een outlook van vijf jaar groei”, vertelt Van der Valk. “Voorheen was er altijd wel constante innovatie van de hardware die meehelpt dat applicaties langer mee kunnen. Als je bijvoorbeeld de hardware van nu vergelijkt met die van drie jaar geleden dan is er winst door de hogere kloksnelheden, van processoren, memorybus en de frequenties van alles. Als je momenteel alleen een upgrade van de hardware doet, heb je al een winst van 35% bij SQL Server Integration Services met dezelfde toepassing. De trend is echter dat de kloksnelheid van processoren niet nog verder omhoog gaat, maar dat we meerdere cores gaan gebruiken. Er kan meer op een socket verwerkt worden. De vraag is echter hoe je dat zo efficiënt mogelijk kunt doen. Daarbij kunnen wij helpen. Ook virtualisatie wordt steeds meer toegepast. Daar kun je zeker voordeel mee behalen, maar het is wel een extra variabele bij performance tuning. Wanneer je een serverpark hebt waarbij veel servers weinig staan te doen, is het een prima middel, maar wanneer de host zwaarder belast wordt, gaat virtualisatie ten koste van de performance. Dus als de host meerdere resources moet delen, merk je dat duidelijk. Ik heb mooie demonstraties gezien waarbij virtualisatie geen impact heeft op de performance, maar we hebben zelf tests gedraaid waarbij bleek dat de applicatie tot 60% trager werd. Bij gevirtualiseerde bedrijfskritische applicaties waarbij eisen gesteld worden aan responsetijden en waar een gegarandeerde werkload belangrijk is, is goed testen dus extra belangrijk .