Sequentiële lees/schrijf-acties aanzienlijk sneller dan random

Schijfperikelen en andere onderschatte 'I/O-zaken'

Henk Boot en Marja van der Loo

Dba's hebben de handen vol aan het in de pas houden van de aan hen toegewezen databases. vooral de prestaties, of liever gezegd de wisselende en tegenvallende prestaties, veroorzaken veel hoofdbrekens. Een niet te onderschatten factor bij deze problemen is de onderliggende hardware en met name de schijven die gebruikt worden voor de gegevensopslag. In dit artikel wordt dieper ingegaan op factoren die van belang zijn voor zowel de planning en het onderhoud als de te meten waarden.

Als voorbeeld wordt uitgegaan van SQL Server 2000 in een Windows 2000-omgeving. Op een aantal specifieke voorbeelden na is de gedachtengang van toepassing op vrijwel alle platforms.

HOGERE DICHTHEID

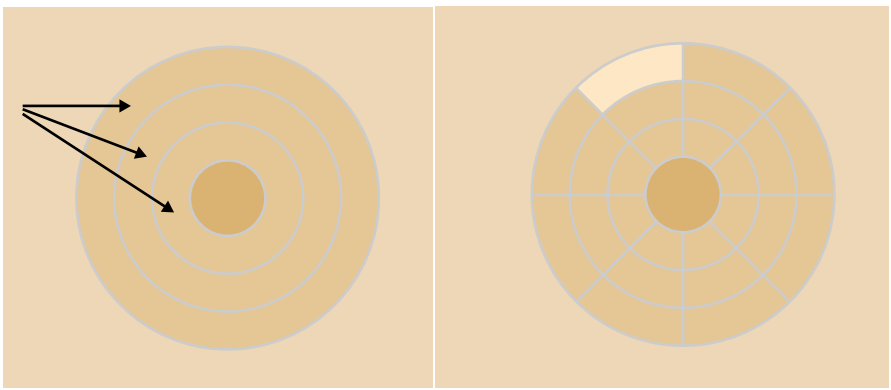
Vaak loopt er een stevige scheidslijn tussen de afdelingen dba en systeembeheer. Daarom geven we eerst een beknopt overzicht van de zaken die met de schijven te maken hebben. In tegenstelling tot veel componenten binnen een computersysteem, zoals cpu, geheugen en de verbindende bussen, zijn op het gebied van harde schijven -in het vervolg disks te noemen- weinig spectaculaire veranderingen aan te geven ten opzichte van de situatie van enkele jaren geleden. De disks draaien harder en zijn voorzien van magnetische lagen die een hogere dichtheid van nullen en enen toelaten, zodat op een beperkte ruimte veel gegevens kunnen worden opgeslagen; dat is eigenlijk al het verschil. Nog steeds vormen ze de meest trage component binnen een systeem. Een disk bestaat uit een of meer platen die aan beide zijden voorzien zijn van een magnetische laag. De moderne disks met een capaciteit van 72 GB bestaan uit twaalf platen. Daartussen bewegen zich de koppen, die zowel kunnen lezen als schrijven. De koppen zelf zijn verbonden met een as die kan draaien; zodanig dat de koppen naar binnen, naar de as van de platen of naar de buitenrand kunnen bewegen. Het is dus mogelijk alle ruimte te bereiken door de koppen naar binnen of buiten te bewegen. Maar alle koppen gaan gezamenlijk doordat ze alle aan dezelfde as zitten.

Iedere database behoort op een eigen RAID 5- of -10-set te staan; evenzo moet ieder transactielogbestand een eigen RAID 10-set hebben

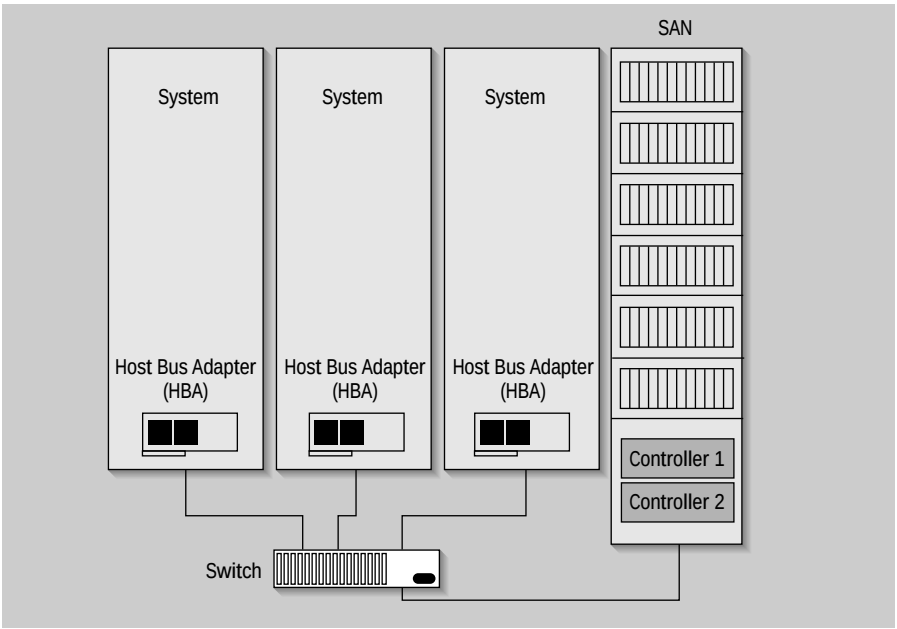
De platen zelf zijn opgedeeld in concentrische tracks, die -in tegenstelling tot bijvoorbeeld de groef van een lp- geen spiraal is. Elke track wordt korter van buiten naar binnen. De tracks zijn zelf weer opgedeeld in sectoren. Afhankelijk van het type disk kent elke track -de binnenste of de buitenste- een gelijk aantal sectoren; ze verschillen enkel in grootte van elkaar. Andere typen disks hebben een indeling die naarmate de tracklengte toeneemt, per track meer sectoren bevat. Aan de buitenrand meer sectoren dan dichter bij de middenas.

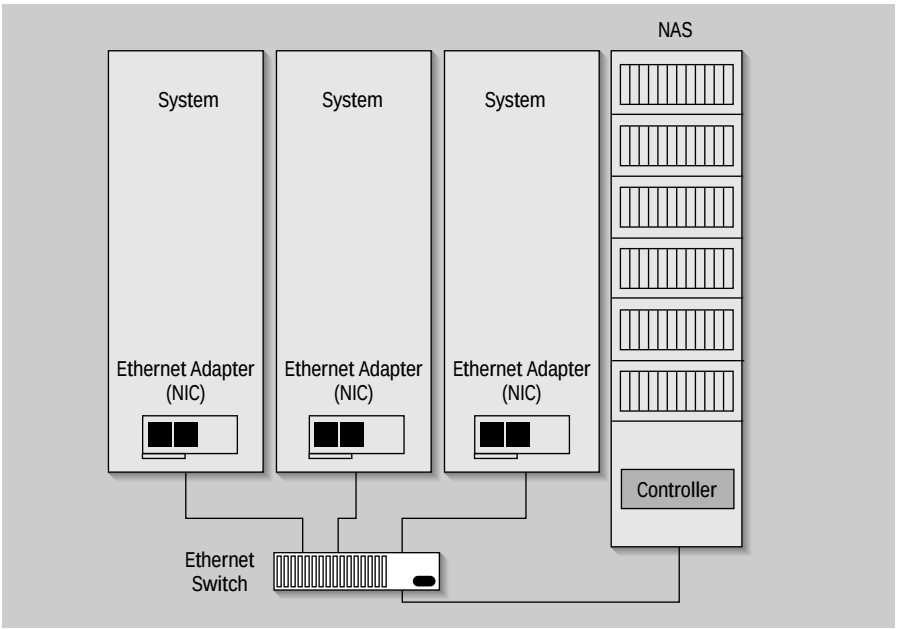
REGEL 1

Om een I/O-opdracht uit te voeren, moet de disk weten in welke richting de kop zich moet bewegen. Via diverse tabellen binnen het besturingssysteem van de computer wordt bijgehouden op welk fysiek



Figuur 1: Tracks en sectoren.





FIGUUR 2 EN 3: GEGEVENSOPSLAG DOOR MIDDEL VAN EEN STORAGE AREA NETWORK (SAN) RESPECTIEVELIJK VIA NETWORK ATTACHED STORAGE (NAS).

adres de gewenste gegevens staan. Dit bepalen is een voor een computer eenvoudig en snel uit te voeren proces. Voor de fysieke schijf is dit echter heel anders. De disk heeft te maken met twee factoren: rotatietijd en zoektijd (*rotational latency* en *seek time*). De disk moet op precies het juiste moment de kop op de juiste plek hebben. Bij disks met een toerental van 15.000 wentelingen per minuut en de te lezen of schrijven plek staat precies iets minder dan 360 graden van de huidige positie van de kop -moet naar positie "-1 graad"- dan kost het altijd nog 15.000/60=1/250 seconde, dat is 4 milliseconde.

Niet altijd zal een volledige wenteling moeten worden afgewacht, het gewenste plekje kan ook precies naast de kop liggen. Gemiddeld kan dan ook 2 milliseconde worden aangenomen. Maar de kop staat natuurlijk nooit boven de gewenste track. Gegevens van diskleveranciers geven aan dat de gemiddelde 18 GB disk met een toerental van 15.000 opm er 0,6 milliseconde over doet om van de ene track naar de naastliggende te komen. Fraaie waarden voor leesacties die een bestand sequentieel doorlezen.

Helaas voor de performance zijn leesopdrachten naar een database *random*. De koppen zullen enorm heen en weer bewegen tussen buitenste en binnenste tracks. Dezelfde disk die zo fraai 0,6 ms neerzette, komt bij *random access* uit op een gemiddelde waarde -voor lees/schrijf-acties- zo hoog als 4,2 ms. Een maximale stap tussen de uiterste tracks kost zelfs 9 milliseconde.

Random lees- of schrijfbewerkingen kosten zelfs 6,2 milliseconde, met pieken van 11 ms bij een volle disk

Een waarde die vaak voorkomt als de schijf voor honderd procent bezet is. Gecombineerd zal op een moderne disk de gemiddelde lees- of schrijfactie 2,6 ms in beslag nemen voor sequentiële opdrachten, namelijk 2 ms rotatievertraging plus 0,6 ms voor de kopbeweging. Random lees- of schrijfbewerkingen kosten zelfs 2 plus 4,2 is 6,2 ms. Met pieken van 11 milliseconde bij een volle disk.

Zo luidt de eerste regel: houd schijven altijd voor 20 a 30% leeg. Dan is er ruimte voor een eventuele groei, maar ook is de prestatie het hoogst (zie ook figuur 4).

REGEL 2

Databases vertrouwen voor hun data-integriteit volkomen op locks. Een lock wordt opgezet zodra het I/O-verzoek wordt gedaan. Is het een exclusieve lock en verloopt de I/O-afhandeling niet vlot genoeg, dan lijken volgende I/O-verzoeken daaronder voor hetzelfde te locken object, een tabel bijvoorbeeld. Nog erger wordt het als de disk het aantal I/O-verzoeken niet kan afwerken voordat het volgende verzoek voor deze disk wordt ontvangen. Queueing (wachtrijvorming) begint te ontstaan.

Intelligente diskcontrollers helpen hier wel een handje om de pijn te verzachten.

Met name RAID-controllers kennen een vorm van sorteren van random I/O-verzoeken. Ze verzamelen een aantal random I/O's en sorteren deze zodanig, dat de kopbewegingen die nodig zijn om deze I/O's af te handelen het meest efficiënt plaatsvinden. Dit geldt echter niet voor de sequentiële I/O-verzoeken.

De tweede regel, die hieruit volgt, is dat random en sequentiële I/O's zoveel mogelijk op verschillende disks moeten worden uitgevoerd.

REGEL 3

Door aparte disks aan te wijzen voor een sequentieel bestand, zoals een transactielog, kan de I/O-controller zijn werk beter doen. Een verdere verbetering treedt op als op dezelfde disk meerdere actieve sequentiële bestanden worden geplaatst, zodat automatisch een random gedrag optreedt, met alle voordelen van dien. Verderop in dit artikel wordt hierbij een kanttekening geplaatst als het om meerdere transactielogs op een disk gaat.

Bij volledige random benadering van gegevens is het raadzaam de gegevens over meerdere spindels (disks) te verspreiden. Het al genoemde RAID (*redundant array of inexpensive* -ook wel: *independent*- disks) is hiervoor uitermate geschikt.

Meerdere disks vormen samen een logische disk. Door het vergrote aantal koppen is -naast een veel grotere capaciteit als zodanig- die capaciteit sneller te exploiteren. Tel daarbij op de intelligentie van de controllers plus de bijbehorende grote tot zeer grote mogelijkheid gegevens te cachen in deze controllers, en het voordeel mag duidelijk zijn.

Windows 2000 komt standaard met een softwareversie van RAID. Deze is weliswaar bruikbaar, maar zeker niet aan te raden bij het gebruik van databases waarbij het om prestaties gaat. Onder meer omdat een van de cpu's, of de enige cpu van het systeem, naast de gewone taken nog eens voor diskcontroller moet spelen, en omdat er geen echte cache-mogelijkheid bestaat, zoals met hardware-RAID.

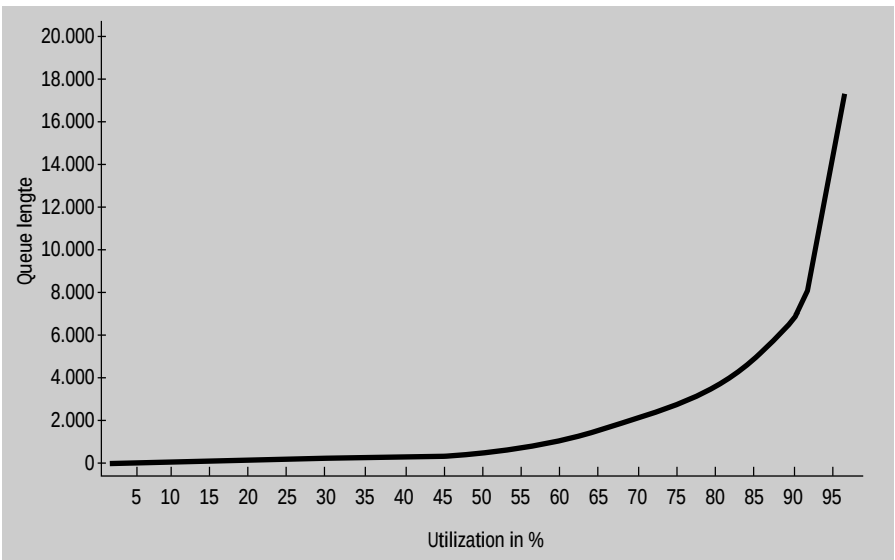
De RAID-caches in de controller dienen een tweeledig doel. Enerzijds bufferen ze gegevens waarvan een algoritme het vermoeden berekent dat deze binnenkort ook opgevraagd zullen worden. Dit is gebaseerd op het principe van *read-ahead*. Anderzijds buffert de cache de te schrijven gegevens. Voor de applicatie lijkt het of de gegevens al fysiek op disk staan (acknowledged), terwijl ze juist iets later echt op de plaat worden geschreven. Zijn de gegevens direct na het schrijven weer nodig, dan kan de cache ze razendsnel weer leveren.

Voor databasegebruik komt een aantal vormen van RAID in aanmerking. RAID 0, waarbij disks in een *stripe set* zijn gevat, valt af. Bij RAID 0 is de inhoud van de logische disk in werkelijkheid verdeeld over twee of meer fysieke disks. Block 1 staat op disk 1, block 2 op disk 2, block 3 op disk 3, block 4 weer op disk 1 enzovoort. Doordat een veiligheidsmarge ontbreekt is RAID 0 ongeschikt voor databasegebruik. Uitval van een schijf betekent uitval van alle gegevens. RAID 1 is al bruikbaarder. RAID 1 bestaat uit een logische disk die is opgebouwd uit twee of soms drie fysieke disks die elkaars spiegelbeeld zijn. RAID 1 wordt ook wel mirroring of shadowing genoemd. Valt een fysieke disk uit, dan zijn alle gegevens nog steeds benaderbaar op een andere schijf van de set. Als op de controller *write caching* is aangezet, zodat schrijfacties 'sneller dan een disk' kunnen worden afgehandeld, is RAID 1 uitermate geschikt voor sequentiële bestanden, zoals een SQL Server-transcatielog.

RAID 5 is wederom een zeer geschikte variant voor database-omgevingen. De logische disk bestaat nu uit een aantal fysieke disks, die tezamen een stripe set à la RAID 0 vormen plus een additionele pariteitsdisk. Door deze toevoeging van pariteit is het verlies van een disk geen groot probleem en is het systeem in staat de verloren gegevens te reconstrueren op basis van overgebleven gegevens en pariteitsbits. RAID 5 is zeer geschikt voor databases die voornamelijk worden gelezen. De kosten in performance voor schrijfacties zijn te hoog door de extra lees/schrijf-acties die het pariteitsmechanisme toevoegt.

De standaard softwareversie van RAID in Windows 2000 is bruikbaar, maar zeker niet aan te raden als het om prestaties gaat

De meest ideale vorm van RAID in een database-omgeving is RAID 10, een gespiegelde stripe set. In combinatie met een batterij-ondersteunde write cache is RAID 10 snel in het lezen van een van de mirrors, schrijven verloopt redelijk snel en de betrouwbaarheid is zeer hoog.



FIGUUR 4: DE QUEUE-LENGTE LOOPT SNELLER OP DAN DE TOENAME VAN HET SCHIJFGEBRUIK.

Regel drie is dat het beter is een RAID-set van acht 9 GB-schijven te hebben dan één disk van 72 GB.

De database-engine van SQL Server is zeer gevoelig voor I/O-vertragingen. Doordat zeer veel applicaties tegen de database 'aanpraten', die elk onder meer hun interne structuur en vele bij te houden locks hebben, ontstaan al snel problemen als locks niet tijdig kunnen worden losgelaten en gaan blokkeren. SQL Server houdt dan domweg op te performen en staat te wachten op het opheffen van de blokkades. Kenmerkend voorbeeld is een onjuiste indexering, die full table scans doet ontstaan. Veel I/O's van elk 6 tot 10 milliseconden zorgen voor locks en wachtrijen voor de disks.

METEN

Op het scheidsvlak tussen dba en systeembeheer ligt ook het meten van de databaseperformance tegen de achtergrond van het I/O-subsysteem. In het geval van Windows 2000 kan de beheerder met behulp van System Monitor informatie boven water halen over het gedrag van

De database-engine van SQL Server is zeer gevoelig voor I/O-vertragingen

fysieke disks. Alleen, wat te doen als de fysieke schijf meerdere partities ofwel logische disks bevat? Op disk 2 bestaan bijvoorbeeld de partities F en G. Daarvan zijn aparte gegevens gewenst. Wil de beheerder deze te zien te krijgen, dan moet hij een optie activeren van binnen Computer Management Console: 'diskperf'. Deze kent een aantal parameters. Diskperf zonder parameters rapporteert de status van diskperf. Met de toevoeging van parameter "-y" wordt het mogelijk gegevens van logische disks te verkrijgen. Het grote nadeel van diskperf is wel dat het systeem een herstart nodig heeft om de ingestelde diskperf-parameter te activeren. Heel slordig.

Is diskperf eenmaal geactiveerd met de optie voor logische disks, dan zijn de volgende waarden van belang voor zowel PhysicalDisk als LogicalDisk. Het aantal leesopdrachten per seconde voor de geselecteerde disk of diskarray is "Disk Reads/sec". Schrijfopdrachten staan bij "DiskWrites/sec". De mate waarin de disk in staat is de gevraagde I/O's op tijd af te werken wordt weergegeven met "Avg. Disk Queue Length", terwijl "Avg. Disk Sec/Read" en "Avg. Disk Sec/Write" aangeven hoe lang een gemiddelde lees- of schrijfactie duurt.

Aan de hand van deze gegevens en die van de leverancier van de disks kan een probleem worden opgespoord. Stel, we gebruiken een RAID 10-set, die bestaat uit vier gemirrorde disks van elk 9 GB en een theoretische capaciteit van 110 I/O's per seconde. Dit getal komt uit een gemiddelde random I/O zoek- en rotatietijd van totaal 6,2 ms, die overeenkomt met 160

I/O's per seconde. Maar door te hoge vulling en de tijd die de controller nodig heeft is een getal van 110 I/O's reëler te noemen. Nu meten we 100 reads/sec, 50 writes/sec, een queue-lengte van 2 en de gemiddelde leestijd is 0,009 seconde en 0,007 voor een schrijfactie. Om nu te zien of een grenswaarde op komst is, kan de volgende formule worden gebruikt:

*I/O's per disk = (reads/sec + 2*writes/sec)/aantal disks*

Dit leidt tot (100+(2*50))/8=25 I/O's per seconde. Niets aan de hand. Anders zou het zijn als bijvoorbeeld het aantal reads/sec 500 zou zijn en het aantal writes/sec 300. Duidelijk mag zijn dat er dan een wachtrij ontstaat. Maar hoeveel disks zouden dan nodig zijn? Uitgaande van bovenstaande formule geldt dat het totaal aantal I/O's 500+2*300, dus 1100 I/O's is. De theoretische grens is 110; er zijn tenminste 10 fysieke disks nodig om obstructies te voorkomen. Voor RAID 5-omgevingen kan de grenswaarde worden bepaald door de formule aan te passen naar:

*I/O's per disk = (reads/sec+4*writes/sec)/aantal disks*

In een voorbeeld dat 100 reads/sec en 50 writes/sec meet in een 4-disk set is de uitkomst (100+4*50)/4=75, wat onder de grens van 110 ligt. Met deze formules is eenvoudig te bepalen of een grens is benaderd of al wordt overschreden.

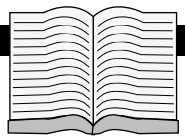
DE LAATSTE REGEL

Mede op basis van deze gegevens kunnen we een aantal regels opstellen die ertoe leiden dat uit de bestaande diskconfiguratie de meeste databaseperformance valt te halen. Databasetransactielogbestanden worden altijd sequentieel geschreven en gegevensbestanden ondervinden random I/O's, ook al is de leeswijze sequentieel. Dit laatste omdat zelden of nooit slechts één enkel (user)proces de database benadert. We moeten dit soort bestanden dus gescheiden opslaan. Hierboven is al gemeld dat als meerdere bestanden van sequentiële aard op dezelfde disk worden geplaatst, het uiteindelijke resultaat toch een random benadering is. Sequentiële lees/schrijf-acties zijn aanzienlijk sneller dan random, door de verminderde zoektijden.

Hieruit volgt de laatste regel: iedere database behoort op een eigen RAID 5- of -10-set te staan, en evenzo moet ieder transactielogbestand een eigen RAID 10-set hebben.

Pas op deze manier ingericht kan een databasesysteem optimaal presteren, gezien vanuit het I/O-subsysteem. ●

Henk Boot en Marja van der Loo (www.bootstrap-it.com) zijn partners in Bootstrap Systems, dat is gespecialiseerd in toepassingen op het gebied van performance en betrouwbaarheid.



Congressen, beurzen e.d.

10-14/6: IBM Software Symposium 2002
Waarin o.a. opgenomen: DB2 Technical. Wenen. Org./inf.: www.ibm.com/events/software/symposium2002

12-13/6: Corporate Portals
Symposium met beurs. Londen, Heathrow Marriott. Org./inf.: Butler Group, www.butlergroup.com/events/portals

1-5/7: Microsoft TechEd Europe 2002
Voor ontwikkelaars, systeemarchitectecten en databasebeheerders. Barcelona, congrescentrum Montjuic. Org./inf.: www.microsoft.com/europe/teched/home.asp

Cursussen, seminars e.d.

13/5: Netwerken en 'communities'
Avondseminar. Utrecht, Jaarbeurs. Org./inf.: Cibit Adviseurs en Opleiders, www.cibit.nl, (030) 2308900.

16/5: Best practices in data warehouses
Seminar met Nigel Pendse, Erik Fransen, Rick van der Lans en Rudi De Backer. Diegem (B), Sofitel, 14.00-21.00 uur. Org./inf.: I.T. Works, www.itworks.be/DW_Best_Practices.html, (00) 32 9 2415613.

16/5: Workshop Brio Reports
Tevens aandacht voor Brio Intelligence 6.6. Org.: Brio Software Nederland, www.brio.nl (0184) 448100.

16/5-20/6: Data Warehousing
Zes avondcolleges. Utrecht, geb. Hogeschool Domstad. Org./inf.: Management Studiecentrum, www.studiecentrum.com, (010) 4603041.

22/5: Business intelligence in een realtime wereld
Middagseminar. Kosten: geen. Org.: The Vision Web. Info: www.topmanagement.giarte.com/showevent.html?event_id=22

22-23/5: Next-generation intranets
Seminar met Peter Hinssen. Diegem (B), Sofitel, 14.00-21.00 uur. Org./inf.: I.T. Works, www.itworks.be/next_generation_intranets.html, (00) 32 9 2415613.

27-28/5 en 3/6: Dimensionaal modelleren
Cursus door Harm van der Lek. Amsterdam, Planetarium Gaasperplas. Org./inf.: VanderLek Advies, www.vdlek.nl, (035) 6216928.

30-31/5: Benchmarking your Knowledge Management Intranet
Amsterdam, Apollo-hotel. Org./inf.: www.kmmagazine.com/events/

13/6: Demomiddag
Door leverancier Brio Software Nederland. Sliedrecht, 14.00-16.30 uur. Org./inf.: www.brio.nl, (0184) 448100.