Software-oplossing verhoogt beschikbaarheid van database

Database mirroring met SQL Server 2005

Bram Dons

Een van de meest belangrijke eisen voor een business critical applicatie is de beschikbaarheid van de gebruikte database. Het maximaal beschikbaar houden van de database staat dan ook bij menig database-beheerder bovenaan de prioriteitenlijst.

Om de beschikbaarheid te maximaliseren moet worden getracht om de geplande en niet-geplande uitval zo klein mogelijk te houden. Deze eis lijkt op zich nogal vanzelfsprekend maar in de praktijk blijkt het toch behoorlijk lastig om daaraan te voldoen, want er spelen talrijke factoren een rol. Een niet-geplande uitval wordt voornamelijk veroorzaakt door uitval van de hardware (computer en opslagsystemen), corrumpering van disks, stroomuitval, communicatiefouten, natuurlijke rampen, terrorisme, menselijke fouten en andere factoren die kunnen leiden tot de uitval van de productie-database. Geplande uitval heeft voor-namelijk betrekking op veranderingen die van tijd tot tijd nood-zakelijkerwijs aan opslagsystemen en servers moeten worden aangebracht. Daarbij moeten we denken aan hardware en software upgrades.

> Een database mirror bevindt zich constant in een staat van herstel

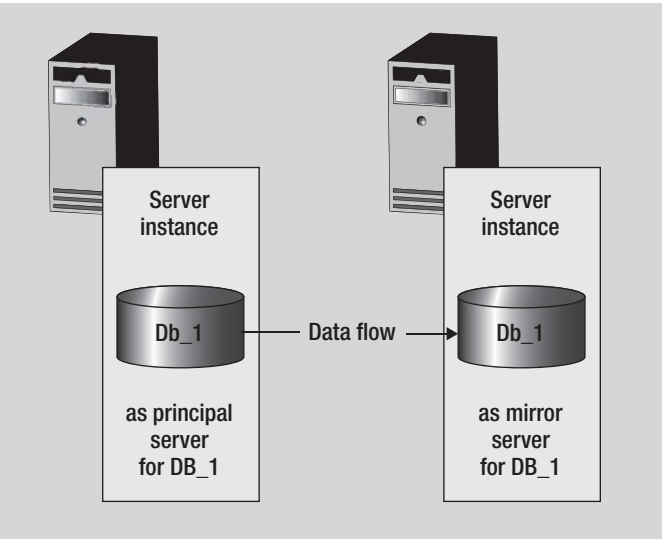
Er bestaat een aantal technieken om de beschikbaarheid van een database bij zowel geplande als niet-geplande uitval te optimaliseren. Zo bestaat er binnen Microsoft SQL Server omgevingen al langere tijd database mirroring-, replicatie-, failover- en clustering-technologie om de beschikbaarheid van databases te verbeteren. Microsoft biedt vanaf de SQL Server 2000 de mogelijkheid om een stand-by server te creëren voor replicatie, log shipping en backup/restore en met de introductie van SQL Server 2005 is daar nu ook *database mirroring* bijgekomen. De vraag rijst dan ook waarom database mirroring nu pas is toegevoegd, voldeden de andere voorzieningen dan niet of boden ze niet het juiste niveau bescherming tegen uitval? In dit artikel trachten we zo goed mogelijk een antwoord te geven op de vraag wat het verschil is tussen alle door SQL Server 2005 ondersteunde database-protectiesystemen en welke voor- en nadelen aan elk verbonden zijn.

SQL Server 2005 Database mirroring

Zoals gesteld, database mirroring is een nieuwe feature in SQL Server 2005 waarmee de inhoud van een database naar een andere database is te spiegelen. Database mirroring biedt een aantal voordelen, waaronder een hogere mate van databeveiliging, beschikbaarheid van een database wanneer deze in productie is en bij upgrades. In het geval van een database crash is een failover mogelijk naar de gespiegelde database. Het betekent echter niet dat mirroring de andere recovery- en failover-oplossingen overbodig maakt, want mirroring is te combineren met de bestaande clustering-, replicatie- en log shipping-voorzieningen. Afhankelijk van het gewenste niveau van hoge beschikbaarheid, veilige transacties, prestaties en failover, kan de gebruiker kiezen uit de diverse combinaties.

Database mirroring concept

Database mirroring is hoofdzakelijk een software-oplossing om de beschikbaarheid van de database te verhogen. Mirroring wordt per database geïmplementeerd en werkt alleen met databases die



Afbeelding 1: Principe Database Mirroring (bron Microsoft).

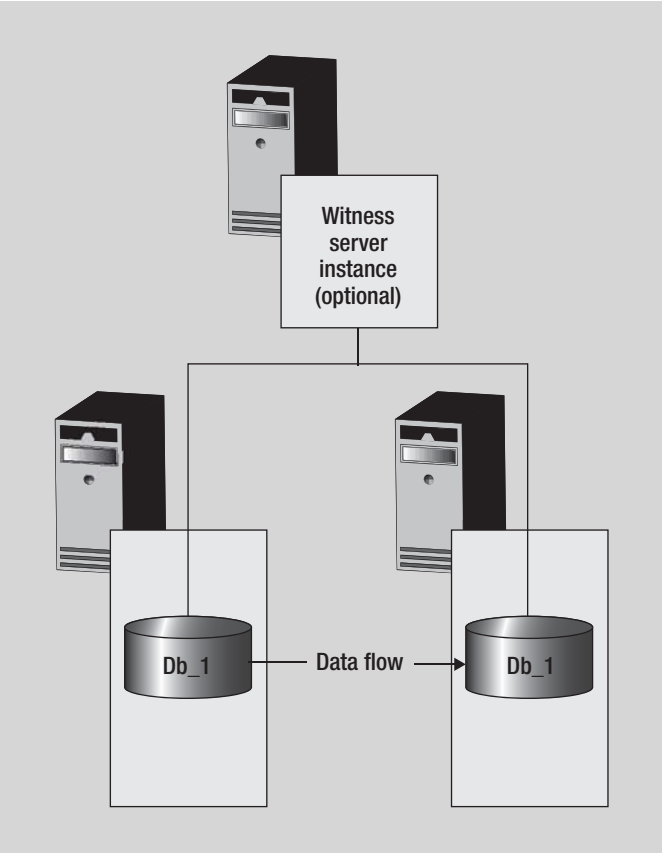
van het 'full recovery' model gebruik maken; mirroring van de *master, msdb, tempdb* of *model databases* is niet mogelijk.
Database mirroring onderhoudt twee kopieën van een enkele database die op verschillende 'instances' van een SQL Server Database Engine (server instances) draaien; de server instances staan op servers die om veiligheidsredenen op verschillende fysieke locaties kunnen zijn geïnstalleerd. Eén server instance, de *principal server*, fungeert als de productie-database en is dus permanent beschikbaar voor de SQL clients terwijl de andere server, de *mirror server*, fungeert als een *hot* of *warm standby server*. Na beëindiging van het initiële database mirroring synchronisatieproces staat op beide servers een identieke database instance en biedt de hot standby mirror server de principal server een snelle failover-faciliteit waarbij geen verlies van data is bij al gecommitteerde database-transacties. In het geval dat beide instances nog niet zijn gesynchroniseerd, fungeert de mirror server als een warm standby server (waarbij uiteraard wel de kans bestaat op verlies van data). Database mirroring brengt continu elke datawijziging op de principal database aan op de mirror database. Dit omvat de wijzigingen aan de fysieke en logische database-structuren, waaronder tabellen, bestanden en indexen.

> De rol van de witness server is om een automatische failover mogelijk te maken

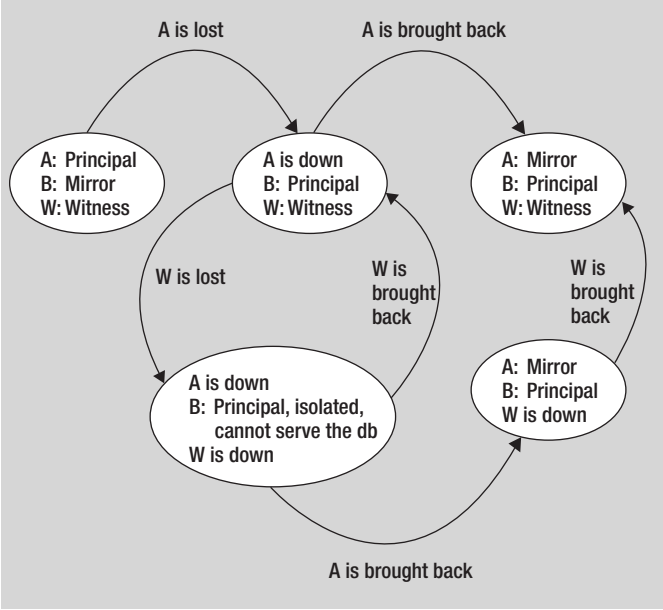
In tegenstelling tot log shipping, dat een volledige transactie-log backup naar een warm standby database verstuurt, werkt database mirroring door een continue stroom van database log records van de principal naar de mirror database te sturen. In alle SQL Server databases worden gewijzigde data eerst in een transaction log opgeslagen, voordat een wijziging aan de eigenlijke datapagina's wordt aangebracht. De transaction log records worden eerst in de database *log buffer* (in het lokale computergeheugen) opgeslagen en daarna zo snel mogelijk naar disk 'geflushed' (ze worden dan zogenaamd 'hardened'). Gelijktijdig worden dezelfde blocken naar de mirror server verzonden. Na ontvangst daarvan plaatst de mirror server de blocken ook eerst in een log buffer, waarna ze zo snel mogelijk naar de disk worden geschreven. Pas daarna worden de op disk opgeslagen transaction log records op de mirror server opnieuw 'afgespeeld' en pas dan bestaat er ook een duplicaat van de principal database-wijzigingen op de mirror server.

Witness server

Naast de twee partner servers (principal en mirror) kan een database mirroring session nog van een derde optionele server worden voorzien, de zogenaamde *witness server*. De rol van de witness server is om een automatische failover mogelijk te maken.
Wanneer database mirroring wordt toegepast in een High Availability (HA) setting dan biedt het bescherming tegen uitval van de principal server. De witness server zelf bevat geen database maar verifieert continu of de principal server nog 'up en running' is. De mirror server initieert alleen een automatische failover als de mirror en de witness server verbonden blijven, nadat beide servers de verbinding met de principal server hebben verbroken. Op het moment dat de mirror server door de witness server wordt geconformeerd over de uitgevallen principal server, dan neemt het automatisch de rol over van de principal server en is de database binnen enkele seconden weer beschikbaar. Een database mirroring session vereist een *quorum* om de database in dienst te houden, waarvan er tenminste twee nodig zijn om een quorum te vormen. De witness server assisteert de principal of mirror server bij het formeren van een quorum. Als een witness server aanwezig is, dan blijven er bij uitval van de principal of mirror database tenminste twee servers over om een quorum te kunnen vormen. De witness server wordt niet als een 'single point of failure' in een database mirroring session beschouwd, omdat als de witness server faalt de principal en mirror gewoon doorgaan om een quorum te vormen.
Database mirroring wordt volledig ondersteund in SQL Server 2005 Standard en Enterprise Edition. Failover servers moeten op dezelfde editie gebaseerd zijn. Server instances die op een SQL Server 2005 Workgroup of Express Edition draaien ondersteunen alleen de witness server.



Afbeelding 2: Optionele Witness Server (bron Microsoft).



Afbeelding 3: Uitval Principal gevolgd door uitval Witness Server (bron Microsoft).

Transactie safety levels

Er zijn drie operationele modes om een database mirroring session in te draaien: High Availability, High Protection en High Performance. De exacte mode is gebaseerd op de instelling van de transaction safety ('FULL' of 'OFF') database-parameter en of een witness server deel uitmaakt van de session. Als safety op FULL staat ingesteld en er is een witness server aanwezig, dan is sprake van een synchrone data transfer en is er een quorum nodig voor de database service; ALTER DATABASE [<dbname>] SET SAFETY FULL. Wanneer safety op 'OFF' staat ingesteld, dan verloopt de communicatie tussen de principal en de mirror asynchroon.

Een database mirroring sessie kan dus in synchrone of asynchrone mode werken. Bij een asynchrone operatie wordt de transactie onmiddellijk 'committed', zonder te wachten op de mirror server die nog bezig is om de log records naar disk te schrijven. Bij synchroon wordt een gecommitteerde transactie door beide servers gecommitteerd. Nadeel is wel dat dit een hogere transactie latency ten gevolge heeft. High Availability mode ondersteunt de maximale database beschikbaarheid met automatische failover naar de mirror databases in het geval dat de principal database uitvalt. De safety staat op 'FULL' en er is een witness server gedefinieerd als onderdeel van de database mirroring session. De HA mode is het meest geschikt wanneer er snelle en betrouwbare communicatiepaden beschikbaar zijn tussen de servers en er een automatische failover voor een enkele database vereist is. In HA mode is de database mirroring self-monitoring, waardoor bij uitval van de principal database of server de witness en de mirror server een quorum van twee servers formeren en de mirror SQL Server een automatische failover gaat uitvoeren. De mirror server kan weer snel beschikbaar zijn omdat van de mirror de transaction logs al gesynchroniseerd zijn met die op de principal server.

De High Protection mode verschilt met de HA mode. Zo is er geen quorum en geen witness server beschikbaar, waardoor een automatische failover niet tot de mogelijkheden behoort. In High Performance mode staat de transactionele safety op 'OFF' en verlopen de datatransfers asynchroon. Er is geen automatische failover mogelijk, er kan verlies van data optreden en handmatige failover is niet geactiveerd. Het enige toegestane type failover is forced service failover, dat een door de database-beheerder handmatig uitgevoerde operatie is. De forced service failover heeft een directe recovery operatie van de mirror database ten gevolg. Zoals gesteld, er kan verlies van data optreden tijdens de recovery fase van de mirror, want meestal zijn er enkele transaction log blocken van de principal niet ontvangen op de mirror (bij een disaster recovery situatie vanaf een remote site).

Database snapshots en mirrors

Wanneer de mirror database zich in een herstelstaat bevindt, dan is deze niet voor clients toegankelijk en leesbaar. Met de SQL Server 2005 Enterprise- en Developer Edition is het echter wel mogelijk om een snapshot van de database te maken, waarna een mirror database op elk moment is te lezen. Een database snapshot biedt een read-only view van een database waarbij consistente data beschikbaar zijn (op het moment dat de snapshot werd gemaakt). De database is op dezelfde wijze toegankelijk alsof het een andere database betreft. Een query op de database snapshot leest de originele versies van elke database die is veranderd na de creatie van de database snapshot file en leest onveranderde data van de originele database. Omdat database snapshots een extra belasting veroorzaken op de mirror server, moet men er rekening mee houden dat het van invloed kan zijn op de prestaties van het database mirroring proces. Omdat slechts één mirror van een database mogelijk is, behoort de scale out naar meerdere read-only servers niet tot de mogelijkheden. In dat geval is transactionele replicatie een betere keuze.

> Database mirroring brengt de standby database veel sneller online dan elke andere HA-technologie

Beschikbaarheid scenario's

Hiervoor is een overzicht gegeven van de diverse safety niveaus bij database mirroring. Hierna geven we een overzicht van een aantal denkbare scenario's waarin de diverse servers en netwerkverbindingen uitvallen en welke gevolgen dit heeft voor de beschikbaarheid van de database. We gaan hierbij uit van twee type denkbare scenario's: de uitval van een of meer servers of databases en het verbreken van een of meer communicatielijnen

tussen de servers. In de volgende scenario's gaan we uit van de gelijktijdige uitval van twee componenten. Het voorbeeld gaat uit van een HA scenario waarbij een principal, witness en mirror server uitvalt en gevolgd door uitval van een ander type server. Tenslotte zien we wat de gevolgen kunnen zijn bij uitval van een of meer communicatielijnen tussen servers. Server 'A' fungeert als principal, server 'B' als mirror en server 'W' als witness server (zie afbeelding 3).

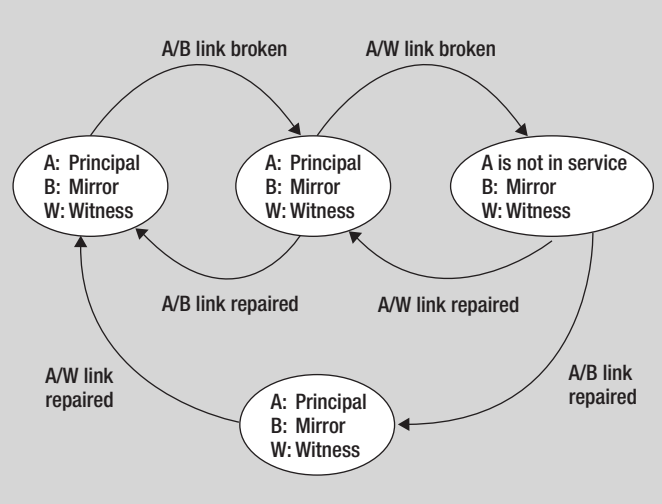
Hiervoor is al de situatie beschreven als de principal server uitvalt en er een failover plaatsvindt. Twee scenario's zijn nog interessanter wanneer de uitval van principal server A gevolgd wordt door de uitval van de nieuwe principal server B of de uitval van witness server W. Wanneer de witness server W uitvalt ná de uitval van de principal server A, dan blijft de nieuwe server B de principal maar werkt verder geïsoleerd, deze kan geen quorum formeren en dus als database server gaan fungeren. Als server A eerst wordt hersteld dan zal server B (omdat er een failover heeft plaatsgevonden) als eerste in aanmerking komen om als principal server te gaan fungeren. Server A ziet dat server B nu als principal fungeert en server A formeert een quorum met server B en wordt dan de mirror. Totdat server W weer beschikbaar is, kan er geen automatische failover plaatsvinden. Als server B uitvalt ná server A en daarna server W, dan zullen de server 'rollen' voor server A en B gehandhaafd blijven, onafhankelijk van de volgorde van opkomst van de servers.

> De HA operating mode vereist drie SQL Server instances

Uit de diverse mogelijke scenario's zijn wel enkele conclusies te trekken. Bij uitval van de principal server hebben uitval van servers en restores geen gevolgen voor de algehele configuratie van de mirror op de nieuwe principal. Als de mirror server eerst uitvalt, dan zal er geen automatische failover plaatsvinden. De daarna volgende uitval van servers en de volgorde van restoratie hebben geen invloed op de functies van de mirroring deelnemers. Dit geldt ook bij een eerste uitval van de witness server.

Scenario bij uitval communicatie

De HA operating mode vereist drie SQL Server instances. Als de server zich op twee of drie onafhankelijke remote sites bevindt dan is de kans aanwezig dat er communicatieproblemen tussen de sites kunnen optreden. Alhoewel de servers op zich stabiel kunnen werken, kan een onderbreking van een of meer communicatielijnen de mirroring session onderbreken. Wanneer we uitgaan van een scenario met drie servers met drie onafhankelijke communicatielijnen dan kunnen er drie verschillende verbindingen uitvallen: A/B, A/W en B/W. Uit afbeelding 4 blijkt dat alleen de verbinding tussen de principal en mirror effect heeft, uitval van de andere



Afbeelding 4: Principal/mirror en witness verbinding verbroken (bron Microsoft).

twee verandert niets aan het gedrag van de database mirroring sessie. Als twee verbindingen gelijktijdig uitvallen dan zal het resultaat hetzelfde zijn als bij uitval van een verbinding, gevolgd door een tweede. De exacte volgorde is onvoorspelbaar. In het geval van een HA-configuratie met drie onafhankelijke servers, volgt er bij uitval van de principal/mirror- of mirror/witness-verbinding geen automatische failover. Een uitval van een principal/witness-verbinding, gevolgd door een uitval van de principal/mirror start wel een automatische failover.

Vier High Availability technologieën

SQL Server 2005 kent nu tenminste vier High Availability technologieën: database mirroring, failover clustering, log shipping en transactionele replicatie. Alhoewel er sprake is van enige overlapping, heeft elke technologie zijn eigen relatieve voor- en nadelen. Bij het vergelijken van de vier technologieën gaan we uit van de volgende basisconfiguraties. Voor database mirroring beschouwen we de HA operating mode met safety FULL en een witness server. Bij failover clustering gaan we uit van een doorsnee twee-node Windows failover cluster met één SQL Server instance. Voor de SQL Server wordt van de ingebouwde log shipping met een aparte monitor gebruik gemaakt. Transactionele replicatie maakt van een aparte distribution server en een subscriber gebruik die dient als een standby server als de publisher server uitvalt.

Mirroring en clustering

Het meest belangrijke verschil tussen database mirroring en failover clustering is het geboden redundantieniveau. Database mirroring biedt bescherming op database-niveau, clustering op server instance niveau. Een ander belangrijk verschil is dat bij database mirroring de principal en mirror server aparte SQL Server instances zijn met verschillende namen. Een SQL Server instance op een cluster heeft slechts één virtuele server-naam en een IP-adres, beide blijven hetzelfde en zijn onafhankelijk van op

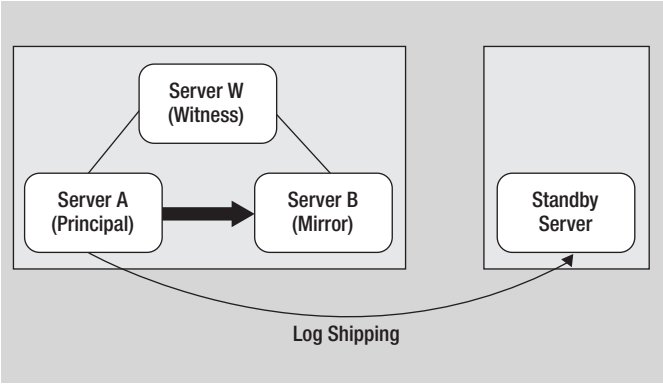
welke node van de cluster de SQL Server instance draait. Als men bescherming van de database op server-niveau wenst dan kan failover clustering een betere keus zijn (bijvoorbeeld, wanneer de applicatie toegang vereist tot meerdere databases). Wanneer men slechts voor één database een hoge beschikbaarheid wenst, dan biedt database mirroring een aantal voordelen. In tegenstelling tot een clustersysteem is er voor database mirroring geen proprietary hardware nodig en kent het geen *potential single point of failure* die de toepassing van *shared storage* nu eenmaal impliceert.

> De witness server assisteert de principal server bij het formeren van een quorum

Database mirroring brengt de standby database veel sneller online dan elke andere HA-technologie. Omdat database mirroring niet te combineren is met clustering, kan men overwegen om een *hot standby* te creëren van een cluster instance database. Let wel, omdat een cluster failover langer duurt dan de time-out waarde van database mirroring, zal een HA mode mirroring sessie op een cluster failover reageren met de uitval van de principal server en de cluster-node in een mirroring state plaatsen.

Mirroring en transactional replication

De enige gelijkenis tussen database mirroring en transactionele replicatie is dat beide zijn gebaseerd op het lezen van de transaction log. Transactionele replicatie wordt vaak gebruikt voor een HA-oplossing, omdat het gebruikerstransacties in enkele seconden vanuit een publisher database naar een subscriber kan afleveren. Database mirroring heeft als voordeel dat het minstens zo snel, zo niet sneller, is dan replicatie en levert bovendien *alle* databasetransacties af en niet alleen de aan gebruikerstabellen gerelateerde. Transactionele replicatie is een geschikte technologie voor uitschalen van data naar meerdere subscribers. Subscriber databases zijn meestal read-only en dus de juiste kandidaat in situaties waar behoefte is aan 'near real-time' data. Database mirroring is compatibel met transactionele replicatie en is meest geschikt voor het bijhouden van een standby van een publisher database. Andere methodes voor de bescherming van een replicatie-publisher (zoals log shipping) kunnen geen standby server onderhouden. Database mirroring is veel sneller dan transactionele replicatie en meer geschikt als hot standby van een publisher database. Als de publisher echter uitvalt, dan moet handmatig de herstelde standby database als publisher weer worden ingesteld en weer verbonden worden met de distributie-server.



Afbeelding 5: Log shipping aanvulling op database mirroring.

Mirroring en log shipping

Zowel database mirroring als log shipping zijn afhankelijk van de restore- en recovery-voorzieningen van de SQL Server database. Een database mirror bevindt zich constant in een staat van herstel, waarbij bijna continu transacties van de principal database worden herhaald, bij log shipping gebeurt dit op periodieke basis. In veel gevallen biedt database mirroring dezelfde soort data-redundantie als log shipping met HA en automatische failover. Er zijn echter scenario's denkbaar waarbij log shipping de beschikbaarheid kan verhogen. Bijvoorbeeld, men heeft een *in-house* HA mirroring-configuratie waarbij log shipping voor disaster recovery-doeleinden wordt toegepast bij een principal server naar een remote site. Als de lokale site uitvalt, dan blijven de data beschikbaar op de secondaire site. Bij een database mirroring failover zal bij log shipping van server B naar de remote standby server normaliter een complete herinitialisatie plaatsvinden.

Conclusie

Het primaire doel van de database mirroring voorziening op de nieuwe SQL Server 2005 is het bieden van een hoge beschikbaarheid en failover-faciliteit als de database server uitvalt. Een andere, al langer bestaande, voorziening binnen de SQL Server omgeving is replicatie. Het nadeel van replicatie is dat bij een vaak veranderend database-schema de database lastig in bedrijf is te houden. Als een primair software-gebaseerde oplossing is database mirroring in principe te vergelijken met log shipping. Echter voor log shipping moeten veel processen worden geïmplementeerd, zoals de creatie van een backup, een backup kopiëren en data restore om deze gesynchroniseerd te houden.
Als echter een van de processen faalt, dan faalt het gehele log shipping proces. Het grote voordeel van database mirroring is dat het twee gesynchroniseerde kopieën van een database op twee aparte servers onderhoud. Dit maakt het mogelijk om automatisch van database server te switchen als de primaire server uitvalt. Zolang als de synchronisatie tussen beide servers bestaat, kan de secundaire server automatisch de verwerking binnen enkele seconden van alle client requests overnemen en gaan er geen data verloren.

Bram Dons is onafhankelijk IT-consultant.