



Sleuteltechnologie verandert IT-architectuur op allerlei niveau's

# Database-virtualisatie

Bram Dons

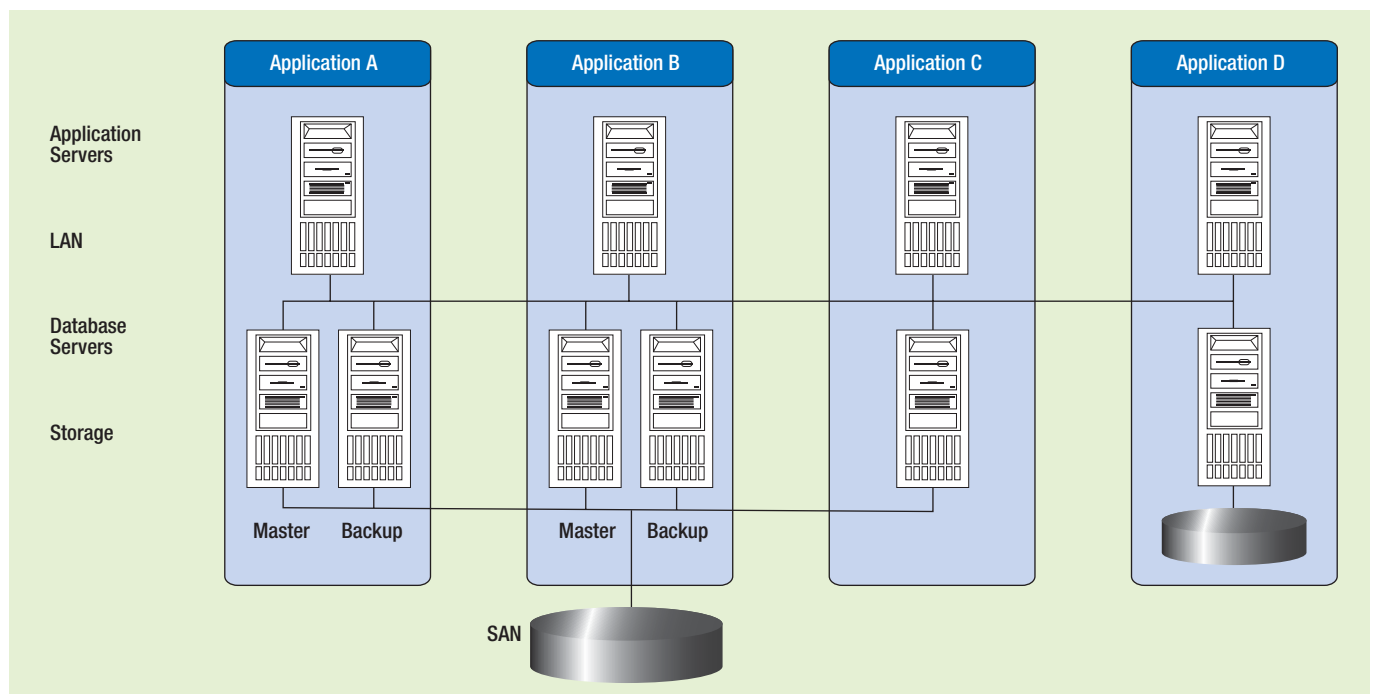
**IT-managers zijn de afgelopen jaren geconfronteerd met toenemende kosten voor aanschaf, beheer en onderhoud van de IT-omgeving. Een voor de hand liggende methode om de kosten terug te dringen is om systemen meer efficiënt te gaan gebruiken, zodat men met minder systemen kan volstaan.**

Bij het streven naar het bereiken van een hogere efficiency-graad analyseert de IT-manager elke laag binnen het OSI-model, waaronder het netwerk, het opslagsysteem, de servers en databases. Alle bronnen binnen de IT-infrastructuur worden daarbij onder de loep genomen, waarbij applicaties zonnodig worden verhuisd naar andere bronnen zodat ze beter kunnen gaan presteren. Teneinde de kosten terug te dringen zijn veel ondernemingen overgegaan tot consolidatie van opslagsystemen en servers. Echter, van oudsher is het probleem bij deze aanpak dat de meeste consolidatie-activiteiten arbeidsintensief zijn. Vanwege

deze tekortkoming van consolidatie hebben veel leveranciers aangezet tot de ontwikkeling van de volgende generatievorm van consolidatie: *virtualisatie*. Virtualisatietechnologie is de afgelopen jaren in de open omgeving al toegepast bij netwerk, opslag en processor. Nieuw is de toepassing van virtualisatie bij databases, en dat is het onderwerp van dit artikel.

## Historie

Het mag niet zo voor de hand liggen, maar virtualisatietechnologie bestaat al lange tijd en werd toegepast voor de komst van de



**Afbeelding 1:** Traditionele dedicated database-architectuur.

computer. Een voorbeeld is de overgang van de handmatig point-to-point bediende telefooncentrales naar een geschakeld telefoonnet, waarbij de gebruiker geen zicht meer heeft via welke knooppunten de verbinding uiteindelijk tot stand komt. IBM valt de eer toe om enkele tientallen jaren geleden als eerste virtualisatietechnologie binnen de mainframe-omgeving te hebben geïmplementeerd.

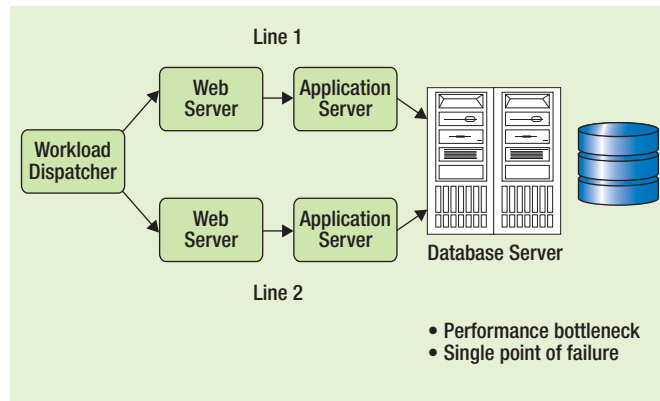
Binnen de IT-omgeving wordt virtualisatie de laatste jaren steeds vaker toegepast. De eerste te virtualiseren component binnen de IT-architectuur was het datanetwerk waar de komst van de netwerk-switch en -router een sleutelrol vervulden (analoog aan het telefonienetwerk). De volgende laag die werd gevirtualiseerd was het opslagsysteem. Met de komst van algemene storage-protocollen en hoge-snelheid datanetwerken is het nu mogelijk om met behulp van een storage switch een opslagsysteem aan meerdere servers te koppelen. Opslag, voorheen direct gekoppeld aan de server (zogenaamde DAS-architectuur) wordt daarmee 'gevirtualiseerd' en virtueel gemaakt in de vorm van een Storage Area Network (SAN). SAN's maken het mogelijk om meerdere applicaties en databases van hetzelfde opslagsysteem gebruik te laten maken. SAN's zijn weliswaar een prima oplossing voor de opslag van applicaties en database servers, maar een belangrijk virtualisatieprobleem blijft op een hoger niveau nog bestaan: database-virtualisatie. Echte database-virtualisatie was tot voor kort nog niet mogelijk en applicatieservers zijn nu voor het grootste deel nog gekoppeld met een bepaalde database server. Failover-functionaliteit wordt nog op het niveau van de database uitgevoerd; dit gebeurt doorgaans met behulp van clustering of andere third-party technologie die er voor moet zorgen dat bij een fout naar een alternatieve database wordt overgeschakeld.

## Van meer recente datum is het gebruik van replicatie op software-niveau

De problemen bij deze aanpak zijn dat elke failover database geconfigureerd moet worden met een equivalente database engine, het beheer van een grote server farm lastig is en het herstel van data bij 'in flight' database-actie zonder hulp van de traditionele commit/rollback schema's niet mogelijk is. Bij alle optredende foutsituaties heeft dit tot gevolg dat de applicatieserver alle database-verbindingen moet herstellen en de uitstaande database requests opnieuw in behandeling moet nemen. Het is aannemelijk dat de toekomstige databases de volgende te virtualiseren component zijn binnen de IT-architectuur.

### Veranderend database-landschap

De laatste tien jaar is het beheer van database-operaties een toenemende uitdaging voor ondernemingen, van welke grootte dan ook, geworden. Niet alleen het aantal databases binnen de ondernemingen is vervoevoudigd, maar ook de hoeveelheid



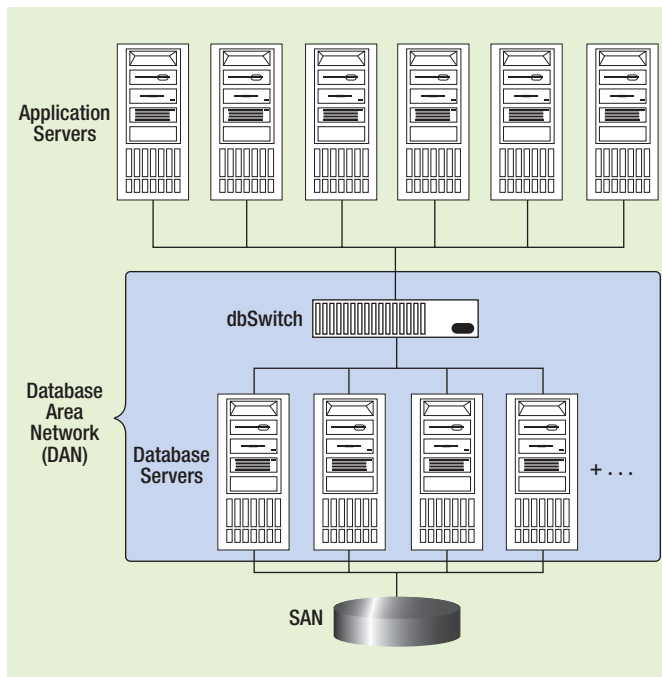
Afbeelding 2: Database vormt single point of failure (bron Avokia).

informatie groeit jaarlijks met een percentage van vaak meer dan 50 procent. De ontwikkeling van nieuwe applicaties en de trend om steeds meer zakelijke 'intelligentie' uit data te distilleren, hebben geresulteerd in complexe infrastructures. Deze trends, in combinatie met steeds hoger gestelde eisen aan beschikbaarheid, regelgeving en veiligheid, hebben belangrijk bijgedragen aan de complexiteit van database-operaties. Het resultaat daarvan is dat het al in hoge mate handmatig en foutgevoelige proces van het beheren van een database steeds meer onbeheersbaar en foutgevoelig wordt.

De belangrijkste reden is dat de meeste databases nog gebaseerd zijn op het traditionele 'dedicated' database server-model, waarbij elke applicatie aan een specifieke database server is gekoppeld, zie afbeelding 1. Deze architectuur kent een aantal tekortkomingen. Ten eerste, de aanschaf en het onderhoud zijn duur omdat de meeste servers 'overbemeten' zijn, teneinde piekbelastingen te kunnen opvangen. Dit resulteert niet alleen in een slechte Return On Investment (ROI), maar bovendien zijn voor iedere server software-licenties nodig voor zowel de DBMS engine als de daarbij behorende secundaire applicaties als backup, file system, volume manager en database monitoring tools. Ten tweede, deze dedicated architectuur is statisch van aard, waardoor wijzigingen aan de database, bijvoorbeeld de migratie van een database instance naar een grotere server, een aanmerkelijke arbeidsintensieve inspanning vragen en een langdurige downtime met zich meebrengen. Een hoge beschikbaarheid voor databases in de huidige IT-organisatie is van groot belang. Echter, de daarvoor benodigde traditionele methoden als replicatie en clustering zijn duur, complex en moeilijk schaalbaar. Tenslotte, database servers zijn 'database-eilanden' geworden, eenzelfde situatie die zich tot voor kort heeft voorgedaan met de storage-eilanden. De afzonderlijke databases moeten individueel beheerd worden, wat leidt tot hogere beheerkosten.

### Clusters

Grote ondernemingen met hoge transactionele volume-applicaties hebben de laatste jaren veel geïnvesteerd in de infrastructuur om de beschikbaarheid van applicaties te kunnen waarborgen. De



**Afbeelding 3:** Database Area Network (DAN).

technologische architectuur is in de loop van de tijd geëvalueerd om het risico van systeemfouten te verminderen. Een doorsnee enterprise applicatie-omgeving bestaat nu uit een aantal lagen van geclusterde front-end web- en applicatie-servers (zie afbeelding 2). Deze configuratie met front-end servers biedt talrijke voordelen. Zo kan de werkbelasting tussen de servers binnen de cluster worden verdeeld waardoor de downtime kan worden gereduceerd of nagenoeg uitgesloten. Clusters van gelijkvormige of identieke servers bieden zekerheid dat bij uitval van een of meerdere nodes andere nodes on-line de taken kunnen overnemen, waarbij de onverwachte extra belasting van de falende node kan worden opgevangen. Wanneer er meer capaciteit nodig is dan kan de front-end capaciteit *horizontaal* worden geschaald door de toevoeging van een of meerdere nodes. De ondersteuning van de applicatie-omgeving aan de back-end kant bestaat nog steeds uit een enkelvoudige database server. Dergelijke systemen zijn ontworpen voor enterprise applicaties en hebben in de afgelopen jaren een totaal ander ontwikkelproces doorgemaakt. Tot voor kort waren technische beperkingen er de oorzaak van dat de horizontale schaalbaarheid van database servers niet mogelijk was en de vele inherente voordelen van clustering niet konden worden toegepast in de database-omgeving. De afgelopen jaren zijn de in kracht toegenomen processoren en de betrouwbaarheid in hoofdzaak aangewend om stand-alone database servers meer robuust en sneller te maken. Wanneer er nu behoefte is aan meer capaciteit, dan is de enige oplossing om de server op te waarderen door een of meerdere CPU's bij te plaatsen. De methode van verticaal schalen van de database is in het algemeen inefficiënt, kostbaar en handhaaft de database als kritische *single point of failure* binnen de data center. Bij een optredende fout op de database server heeft dit als gevolg

dat de applicatie uitvalt. Alhoewel de front-end van deze configuratie schaalbaar is, blijft de flessenhals op de database server-laag bestaan. Een overmatig aantal verbindingen met de applicatie-servers kan bovendien de database destabiliseren. Dat vormt op zich weer een beperking voor het aantal te ondersteunen front-end servers, wat de database server als kritische single point of failure component binnen de architectuur nog eens benadrukt.

## Compensatie beperkingen

Teneinde de beperkingen bij toepassing van een enkele database server-configuratie te compenseren, hebben leveranciers talrijke oplossingen bedacht. Zo is replicatie thans een veel toegepaste database-technologie die de kern vormt van veel hot stand-by en disaster recovery plannen, die beide de zakelijke continuïteit moeten waarborgen. De basis van deze oplossingen is de garantie dat een back-up beschikbaar is wanneer er zich een kritieke fout voordoet op de database-laag. Replicatietechnologie is in verschillende stappen geëvalueerd. De eerste pogingen omhelsden de traditionele back-up, database snapshot en log shipping. De gestelde eisen van ondernemingen aan de beschikbaarheid van data zijn in verschillende stadia geëvalueerd. Bij de eerste oplossingen werd op geregelde tijdstippen een volledige back-up van de database gemaakt. Nadeel van deze methode is data niet 100 procent up-to-date zijn en het restore proces lastig en onbetrouwbaar is. De toepassing van snapshots biedt een meer courante duplicatie van data en in vergelijking met de traditionele back-up is minder bandbreedte nodig. Wanneer er hoge eisen gesteld worden aan up-to-date data, dan kan ook een snapshot niet aan de eisen voldoen.

Log shipping kan worden toegepast voor het synchroniseren van een 'warme' site voor disaster recovery-doeleinden. In plaats van een back-up te maken van de complete database content wordt alleen het database log-bestand naar de target database server verzonden. Het log-bestand wordt gebruikt voor een 'restore' om de target database consistent te maken met de source database. Disk controller replicatie is een complementaire technologie die vaak wordt toegepast in combinatie met andere data-replicatie-technologieën. Het is in feite data-replicatie op hardware-niveau, dat een exacte kopie creëert van een disk. Van meer recente datum is het gebruik van replicatie op software-niveau. Nadeel is dat software-replicatie op file-niveau beperkingen kent, onder andere bidirectionele replicatie. Replicatie met disk controllers of disk mirroring biedt een bruikbare, maar beperkte, oplossing. Een meer geavanceerde benadering werd gevonden in applicatie- en database-replicatie, die een status quo bepalen met de huidige database-redundantietechnologie.

## Virtualisatie

Teneinde aan de hiervoor genoemde problemen met beheer en schaalbaarheid het hoofd te kunnen bieden, overwegen IT-managers thans om door middel van virtualisatie het database-beheerproces te automatiseren. Deze opkomende database-

virtualisatietechnologie biedt de belofte van een hogere efficiency, kostenreductie, en de toevoeging van compleet nieuwe functionaliteiten. Database-virtualisatie biedt gebruikers toegang tot verschillende bronnen op verspreide locaties zonder te hoeven weten waar de data zich daadwerkelijk bevinden. Database-virtualisatie maakt het gebruik van meerdere instances van een DBM mogelijk op verschillende DBMS platforms, gelijktijdig en op een transparante wijze, ongeacht de fysieke locatie. Om een echte database server-virtualisatie en -consolidatie te kunnen bereiken, moet aan een aantal criteria worden voldaan.

## De methode van verticaal schalen van de database is in het algemeen inefficiënt

Ten eerste moet de veiligheid van de data worden gegarandeerd. De oplossing moet voorzien in een automatische failover-functie, transparant voor zowel de gebruiker als de applicatie.

Er moeten geen wijzigingen aan de code van de applicatieserver nodig zijn. De optimalisatie van bronnen moet automatisch over de server farm geschieden en belastingen moeten eenvoudig en dynamisch kunnen worden verschoven om aan een wisselende vraag tegemoet te kunnen komen. De oplossing moet in hoge mate schaalbaar zijn en servers moeten op ieder moment dynamisch toegevoegd kunnen worden. Tenslotte, de installatie en het onderhoud moeten minimaal zijn om de TCO te minimaliseren en de ROI te maximaliseren.

Voor de komst van geavanceerde replicatietechnieken konden applicaties van het two phase commit protocol gebruik maken om meerdere databases te updaten. Bij uitval van een van de databases moet daarbij de update-transactie wachten. Deze methode ondersteunt geen failover of warm standby voor high availability of disaster recovery.

Met asynchrone datareplicatie op de database server zijn minder complexe applicaties te ontwikkelen die toegang bieden tot meerdere databases (in vergelijking met applicaties op basis van het two phase commit protocol). Echter, het definiëren van triggers die voor de talrijke data sets nodig zijn is gecompliceerd en onbetrouwbaar, reden waarom datareplicatie nauwelijks als een bruikbare techniek wordt beschouwd. De thans meest algemene techniek voor toepassing van datareplicatieproducten is gebaseerd op de analyse van het database log-bestand, die door een replicatie-agent wordt uitgevoerd. Ook deze techniek kent een aantal beperkingen. Zo vraagt de replicatie-agent een hoge systeem-overhead en verschilt het log-format van de database server per database-leverancier.

### Savantis dbSwitch

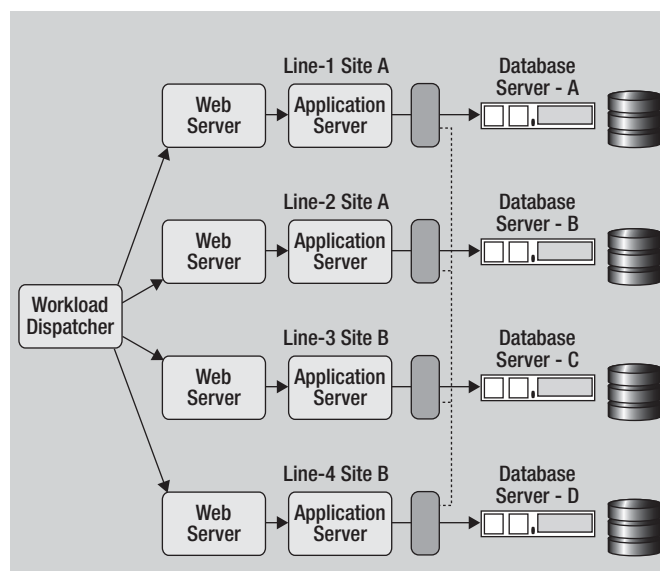
Database-virtualisatie is een nieuwe technologie die nog volop in ontwikkeling is. Sommige leveranciers denken de oplossing te hebben gevonden door middel van op software gebaseerde

applicaties, en andere op hardware gebaseerd. Zo heeft destijds de firma Savantis een switch-gebaseerde database-virtualisatie-oplossing ontwikkeld, 'dbSwitch' genaamd. Het door deze firma ontwikkelde 'Database Area Network' (DAN) is een appliance die zich tussen de applicaties en database servers bevindt en maakt real-time schakeling op database-niveau mogelijk tussen applicatie- en database server (zie afbeelding 3).

De dbSwitch biedt innovatieve resource sharing-algoritmen voor het delen van server-bronnen (CPU en geheugen) tussen diverse heterogene database instances. In tegenstelling tot een shared disk parallele database (bijvoorbeeld Oracle's RAC) biedt dbSwitch server sharing, consolidatie en optimaal gebruik van een gemengde database-omgeving met 'gewone' (niet-parallele) databases.

De DAN-switch vervult een gelijke rol als een Fibre Channel switch in een Storage Area Network (SAN) of de Server Load Balancer (SLB) voor web- en applicatie-servers. In werkelijkheid is de dbSwitch een combinatie van twee hardware- en drie software-componenten. Het hardware-element bestaat uit een standaard router die verbonden is met de database clients en database servers, de ander is een standaard Linux-box waarop de dbSwitch software draait en verbonden is met de router-kant (buiten het 'kritische' data pad). De drie software-componenten bestaan uit de dbSwitch software, DAN Management System (DMS) en Savantis agents. De dbSwitch router is verantwoordelijk voor het doorsturen van packets van een database netwerkprotocol (bijvoorbeeld Oracle's SQL\*NET) van de clients naar de server. De dbSwitch software bestuurt de router, communiceert met de agents en biedt back-end services als persistency en het DMS. Op elke database server draait een agent die luistert naar de TCP/IP-communicatie en fungeert als 'afgevaardigde' van de dbSwitch op de server.

Het voert diverse taken uit, waaronder: herkenning server en database, configuratieveranderingen, monitoring server en



Afbeelding 4: Avokia three-tiered databases-architectuur (bron Avokia).

database instances, instance control en starten, stoppen en herlocatie van databases en bijbehorende opslag. Bij de introductie van de dbSwitch destijds was het bedoeld voor de Oracle-markt, naderhand zijn met succes ook combinaties met DB2 in de praktijk geïmplementeerd. Helaas heeft deze startup-firma het niet gered en is gestopt met de productie van de dbSwitch. Te vroeg in de markt gezet?

## Avokia apLive

Een andere startup-firma, Avokia genaamd, biedt een software-gebaseerde database-virtualisatieoplossing, apLive genaamd. Avokia's apLive heft de beperkingen op van het enkelvoudig database server model door de data laag te virtualiseren met een proprietary transactiereplicatie methodologie. apLive transactiereplicatie maakt niet van database triggers of log-bestand analyse gebruik voor real-time synchronisatie tussen meerdere databases en is 'onafhankelijk' van database platforms en database managers. De volgende databases en OS's staan op de compatibiliteitslijst van Avokia: IBM DB2 op UNIX, Oracle op Linux SuSe en RedHat, SQL Server op Windows NT, 2000 en XP en Sybase ondersteund met JDBC (vanaf) 2.0 en ODBC als database connectivity interface.

apLive vangt de op dat moment uitgevoerde database-transacties af en stuurt deze parallel naar de andere databases in de cluster, waarbij automatisch rekening wordt gehouden met de doorvoer-

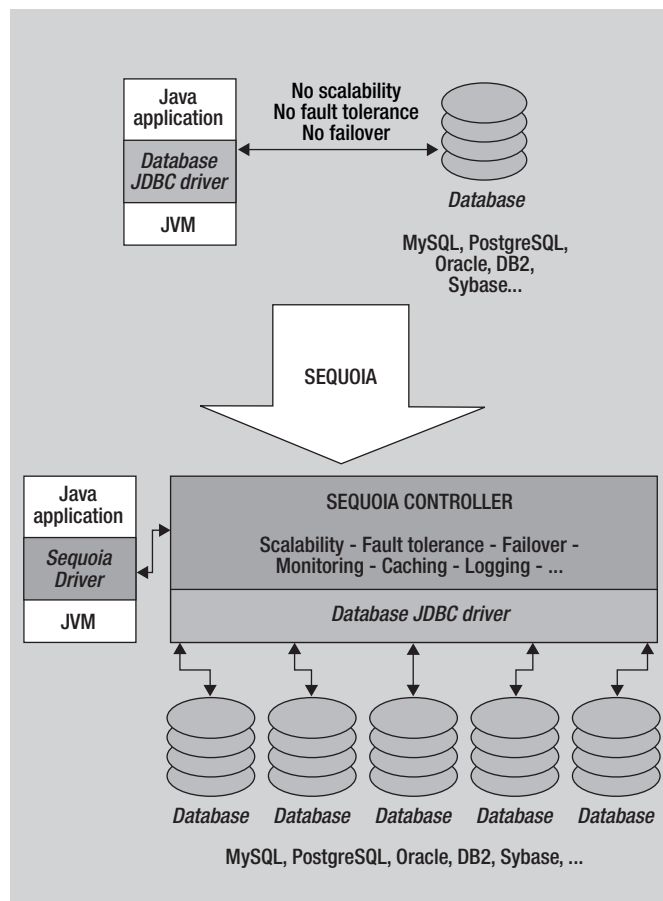
capaciteit. apLive maakt van een Transaction Director gebruik; surrogaat-drivers die in plaats komen voor de door de database-leverancier ondersteunde drivers (het zijn in feite 'wrappers' die elke transactie onderzoeken). De apSuite Director analyseert of het een read-only of een update transactie betreft en omdat 80 procent van alle database-transacties uit leesopdrachten bestaat hoeft slechts 20 procent van de update transacties naar de gerepliceerde databases te worden geschreven.

## Sequoia is via een JDBC-driver met elke RDMS-database te verbinden

apLive is ontworpen op basis van de volgende sleuteleigenschappen: transactiereplicatie op basis van onderschepping van transacties, real-time transactiereplicatie tussen meerdere databases die actief lees en update transacties uitvoeren; automatische transaction routing in geval dat een component binnen de cluster uitvalt; on demand workload balancing voor maximaal gebruik van de beschikbare computerbronnen; eenvoudig en effectief beheer van database-bronnen met on-line onderhoud en dynamische herconfiguratie.

## GridApp Clarity Virtualization

GridApp Clarity van de firma GridApp Systems biedt via haar 'Resource Virtualization & Change management' module een geautomatiseerde virtualisatie-oplossing voor de enterprise database-architectuur. Clarity Virtualization biedt on-line provisioning van database resources; dynamische allocatie van resources voor automatische compensatie bij optredende belastingspieken; centraal beheer van updates; database cloning en 'on demand' toewijzing van capaciteit. Volgens GridApp is op het gebied van geautomatiseerde virtualisatieproducten het kenmerkende verschil tussen GridApp's software-oplossing en de concurrentie dat het in de kern specifiek een beheerlaag is. Bij andere oplossingen zou het naar GridApp's zeggen om abstractie van software-lagen gaan die van (negatieve) invloed zijn op de prestaties of proprietary hardware. Een GridApp oplossing bestaat uit een aantal servers waarop een GridApp management agent draait (dat is een lichtgewicht proces dat wacht op herconfiguratieopdrachten en belangrijke data in de gaten houdt). Een server fungeert daarbij als management node die de andere nodes bestuurt. Servers die geen deel uitmaken van de databases, worden als 'hot spares' beschouwd. GridApp's PowerShare technologie allocereet automatisch bronnen tussen de database instances op basis van prioriteit en prestatie. GridApp's software herconfigureert commodity servers van belangrijke leveranciers, door bronnen on-line te verhuizen. De sleutel van GridApp's oplossing bestaat uit de PowerShare algoritmen die herkennen wanneer procescapaciteit moet worden toegevoegd en weer vrijgegeven. De beheer-software bundelt alle database log-bestanden en biedt de



Afbeelding 5: Architectuur Sequoia (bron Sequoia).

database-beheerder een uniforme blik op de database-infrastructuur, met als resultaat dat de DBA meer controle heeft over de gehele database-omgeving, in plaats van de afzonderlijke databases.

## Continuent Sequoia

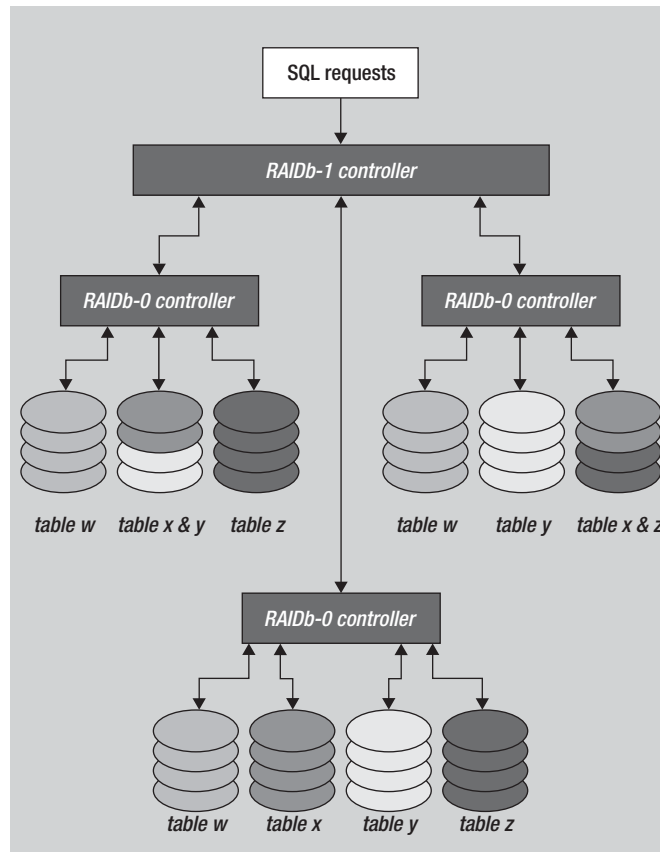
Sequoia is open source database clustering middleware waarmee elke Java-applicatie (stand alone, servlet of EJB container) via JDBC transparant toegang heeft tot een cluster van databases. Sequoia is gebaseerd op het concept Redundant Array of Inexpensive Databases (RAIDb) met een gedistribueerde en gerepliceerde database (zie afbeelding 6).

De database-laag is hoog beschikbaar waarbij Sequoia bestand is tegen database crashes en een transparante failover biedt door middel van data-replicatietechnieken. Daarnaast biedt Sequoia SQL traffic monitoring voor prestatiemetingen en analyse en de ondersteuning van een cluster met heterogene databases. Sequoia is een schaalbaar database cluster doordat bij toevoeging van database nodes de belasting over de nodes wordt verdeeld. Sequoia is via een JDBC-driver met elke RDMS-database te verbinden (in de praktijk betekent dat bijna alle bestaande open source en commerciële databases). Sequoia is een vrij, open source, project en maakt deel uit van de Continuent.org open source projecten. De nieuwste versie, Sequoia 3.0, ondersteunt views, triggers en custom semantics voor specifieke data requests.

De firma Continuent, Inc. is de toonaangevende leverancier van commerciële open source middleware-oplossingen voor database high availability. De nieuwe release van Continuent uni/cluster 2007 is gebaseerd op Sequoia's versie 3.0 en biedt een 'out of the box' geoptimaliseerde oplossing voor MySQL en PostgreSQL databases. De HA cluster-oplossingen bieden geavanceerde beheer-tools, database-specifieke back-up tools en een volledige ondersteuning van het extended SQL dialect voor laatst genoemde databases. Continuent's software werkt met alle belangrijke databases, waaronder MySQL, PostgreSQL, Sybase, Microsoft SQL Server, Oracle en IBM DB2.

## Conclusie

Virtualisatie vormt de sleuteltechnologie die de toekomstige IT-architectuur op allerlei niveau's zal veranderen. Daarmee kan tegemoet worden gekomen aan de steeds hogere eisen die ondernemingen stellen aan prestaties, schaalbaarheid, beheerbaarheid en beschikbaarheid van de IT-omgeving. Virtualisatie is bij verschillende IT-componenten al grotendeels doorgevoerd. Zo is de virtualisatie van opslagsystemen al enkele jaren in volle gang en die van servers vindt thans plaats. Virtualisatie van de proprietary database-omgevingen is het laatste te slechten bastion binnen de IT-architectuur en zal de komende jaren geleidelijk worden ingevoerd. De techniek is echter nog volop in ontwikkeling, er is geen standaard voor beschikbaar en zoals we zagen ontwikkelen leveranciers voor de virtualisatie van databases verschillende typen virtualisatietechnologie. Sommige zijn uitsluitend software-gebaseerd, proprietary of 'open', weer andere



Afbeelding 6: Redundant Array of Inexpensive Databases (bron Sequoia).

maken van Grid-computing gebruik (Oracle). Naast de gangbare technieken is het ook niet uitgesloten dat er toch weer nieuwe ontwikkelingen plaats vinden met switch-based database-virtualisatie.

Veel vragen blijven echter nog onbeantwoord, zoals:

- welke doorbraken in opslag en netwerk zijn nodig om een WAN database-grid economisch haalbaar te maken;
- hoe bouwen we een schaalbaar inter-operabel database-grid;
- hoe moet het gebruik van een database worden gemeten en doorbelast?

Belangrijk voor een heterogene database-omgeving is dat er ook standaarden op dit gebied komen. Maar net zoals er nog steeds geen overeenstemming binnen de storage-wereld is op welke plaats en wijze virtualisatie moet worden geïmplementeerd (op een intelligente switch, opslagnetwerk, server of opslagsysteem), zo zal hetzelfde gebeuren voor database-virtualisatie. Waar de industrie het wel over eens is, is dat database-virtualisatie en SOA een gemeenschappelijk doel hebben: de abstractie van bronnen binnen een netwerk. Afgezien van de plaats en wijze waarop, virtualisatie van storage, processing power en databases zijn allemaal onmisbare componenten voor de toekomstige te implementeren Service Oriented Architecture (SOA) binnen een heterogene IT-architectuur.

**Bram Dons** is onafhankelijk IT-analist.