Een boekje open over High Availability

Droom en realiteit van 100% beschikbaarheid

Jos Visser
<josv@osp.nl>

Open Solution Providers
Dalsteindreef 16
1112 XC Diemen
tel: 020-4950 222
fax: 020-4950 223
e-mail: <info@osp.nl>
web: http://www.osp.nl

Neem voor meer informatie over de HA (of andere) dienstverlening van Open Solution Providers contact op met Erik Meinders of Lucien Versteeg:

Open Solution Providers
Dalsteindreef 16
1112 XC Diemen
tel: 020-4950 222
fax: 020-4950 223
e-mail: <info@osp.nl>
web: http://www.osp.nl

Inhoudsopgave

1 Introductie

Organisaties worden in toenemende mate afhankelijk van informatietechnologie. De beschikbaarheid van centrale computersystemen is dan ook van cruciaal belang voor het functioneren van de organisatie. Het uitvallen van die centrale computersystemen, of, nog belangrijker, van de diensten die door die computers worden geboden, heeft meestal grote gevolgen voor de organisatie: omzetverlies, imagoschade en verloren arbeidsproductiviteit.

Om de beschikbaarheid van de centraal geleverde informatiediensten te waarborgen kan worden overgaan op **clustering**: een technologie waarbij door het inzetten van extra hardware wordt gepoogd een omgeving te bouwen waarin het uitvallen van een component niet leidt tot het permanent uitvallen van de applicaties. Naast deze hogere beschikbaarheid kan een ander voordeel zijn dat in de normale situatie applicaties over meerdere systemen kunnen worden verdeeld, om zodoende de prestaties van die applicaties te verbeteren.

Nu worden er reeds sinds het begin van de informatierevolutie systemen gebouwd met een hoge beschikbaarheid. Denk hierbij bijvoorbeeld aan ticketreserveringssystemen van luchtvaartmaatschappijen en aan de computersystemen die geld- en betaalautomaten bedienen. Het gaat hier echter om specifiek voor die doeleinden ontwikkelde ("proprietary") fout-tolerante computers en programmatuur.

Voor de meeste toepassingen waar verhoogde beschikbaarheid gewenst is zijn dergelijke dure en gesloten systemen niet nodig. Het is namelijk met standaard (open) computersystemen en programmatuur ook mogelijk om een redelijke mate van hogere beschikbaarheid te realiseren. Vrijwel alle Unix leveranciers hebben een oplossing voor "High Availability" en ook Microsoft tracht zich met Windows NT in deze markt te positioneren.

Opmerking

Microsoft heeft onlangs aangekondigd dat hun HA product "Microsoft Cluster Server" vooralsnog geen onderdeel uit gaat maken van de Windows 2000 "Advanced Server" en "Data Center" edities.

Helaas is clustering van die meer generieke systemen geen sinecure: het vereist diepgaande kennis van alle onderdelen van de configuratie: hardware, besturingssysteem, netwerken en applicaties. Daarnaast brengt ook het beheren van een HA-cluster aanzienlijke uitdagingen met zich mee. De oorzaak van deze complexiteit zit hem in het feit dat wij trachten met specifieke software en configuraties generieke hardware en generieke applicaties, die geen van allen waren ontworpen voor een HA-omgeving, verhoogd beschikbaar te maken.

Daartegenover staat echter dat als de cluster correct wordt geïmplementeerd er vele voordelen aan kleven: verhoogde beschikbaarheid, betere prestaties (door applicatieverdeling) en het wordt makkelijker systeemonderhoud te plegen.

HA-clustering wordt door de meesten als zeer ingewikkeld afgespiegeld. Gedeeltelijk is dit terecht, maar er doen helaas ook veel misverstanden de ronde. De Open Solution Providers (OSP) is een onderneming die zich specialiseert in het implementeren, controleren en onderhouden van HA_clusters.

Neem voor meer informatie over de dienstverlening van Open Solution Providers contact

op met Erik Meinders, tel: 020-4950 222, e-mail: info@osp.nl.

In deze publicatie doen wij een boekje open over High Availability.

2 Beschikbaarheid

Ondanks de opkomst van de personal computer heeft de centrale server met daarop centrale applicaties zijn waarde voor de organisatie behouden. De moderne centrale applicatie is echter anders van opbouw dan vroeger. Werd in het verleden met domme terminals aangelogd op een centraal mainframe, tegenwoordig zijn client-server applicaties de gewoonste zaak van de wereld.

In die client-server wereld spant een centrale database samen met centrale applicatielogica teneinde een client-applicatie welke op de computer van de gebruiker draait te bedienen. Deze client-applicatie kent uiteenlopende verschijningsvormen, en varieert van een eenvoudige "telnet" terminal emulator of een web browser tot en met een volledige "fat client". De nieuwste verschijningsvorm van client-server wordt gevormd door de Internet applicaties waarbij medewerkers, klanten of andere relaties over het Internet toegang zoeken tot een of andere centraal aangeboden informatiedienst.

Het verhogen van de beschikbaarheid komt in weze neer op het verlagen van de ongeplande niet-beschikbaarheid (engels: "unplanned downtime"). Geplande niet-beschikbaarheid komt echter ook voor. Denkt u maar aan applicatiemigraties, upgrades naar nieuwe versies en andere applicatiegebonden fenomenen. Met een HA-cluster heeft u ook de gereedschappen in handen om de geplande niet-beschikbare periode van de centrale applicaties zoveel mogelijk te verkleinen. Hierop komen we echter verderop in dit boekje terug.

Ongeplande niet-beschikbaarheid van de centrale applicaties brengt zoals reeds eerder is gesteld grote gevolgen met zich mee, en dient zoveel mogelijk te worden voorkomen. Wat zijn nu echter de redenen van die ongeplande niet-beschikbaarheid?

2.1 Oorzaken van "downtime"

In de ervaring van OSP zijn de redenen van ongeplande "downtime" de volgende:

1. Hardware storingen.
2. Vergissingen en fouten van de systeembeheerder.
3. Software storingen (bugs)

De eerst genoemde oorzaak, hardware storingen, is meestal de reden voor organisaties om over te gaan tot het implementeren van een HA-cluster. Alhoewel de hardware steeds betrouwbaarder wordt blijft de kans op uitvallen levensgroot aanwezig. Daarnaast hebben we ook steeds meer hardware, waardoor de cumulatieve kans op uitvallen van één component toeneemt. Door kritieke hardware componenten meervoudig uit te voeren kan het uitvallen van zo'n component worden opgevangen door hetzij een reservecomponent in te schakelen of door de applicatie te migreren naar een andere node in de cluster.

De tweede genoemde oorzaak is echter veel lastiger op te vangen. De moderne systeembeheerder is een veelgeplaagd persoon, die steeds meer en steeds complexere systemen moet beheren. Het aantal vrijheidsgraden in de hedendaagse infrastructuren neemt immer toe, en daarmee ook de kans op fouten. Een theoretisch goed onderlegd, goed getrainde en ervaren systeembeheerder is dan ook een "must".

Een ietwat paradoxale bijwerking van HA-clusters is dat deze lastig zijn te beheren, en daardoor de kans op fouten van de systeembeheerder wordt vergroot. Oftewel, de cluster die

de beschikbaarheid van de centrale applicaties dient te vergroten bergt in zich gevaren waarmee de beschikbaarheid van de applicaties juist kan worden verkleind!

Over de derde categorie oorzaken van ongeplande niet-beschikbaarheid heeft de gemiddelde klant slechts weinig invloed. Software is helaas nooit foutloos, en er zijn voorbeelden te over van applicaties die door bugs in de software kortere of langere tijd niet beschikbaar waren. Een gedegen test- en acceptatietraject kan dit probleem enigszins bedwingen.

2.2 Mate van beschikbaarheid

Om het effect van de implementatie van een HA-cluster te kunnen meten is het van belang om een gevoel en een maat te krijgen voor de beschikbaarheid van een applicatie. In Service Level Agreements wordt die mate van beschikbaarheid meestal uitgedrukt in een percentage:

$$beschikbaarheid = \frac{Tijd_{up}}{Tijd_{totaal}} * 100$$

Voor de rekenvoorbeelden gaan we uit van de beschikbaarheid over een jaar van een applicatie die 7x24 uur beschikbaar moet zijn.

Als vuistregel hanteren wij dat een (foutloze) applicatie op een losstaande computer ("standalone") een beschikbaarheid van ongeveer **95%** heeft. Dit klinkt al tamelijk hoog, maar bedenk dat dit iets meer dan **18 dagen** ongeplande down tijd per jaar is!

In onderstaande tabel ziet u een weergave van ongeplande down tijd bij een aantal beschikbaarheidspercentages.

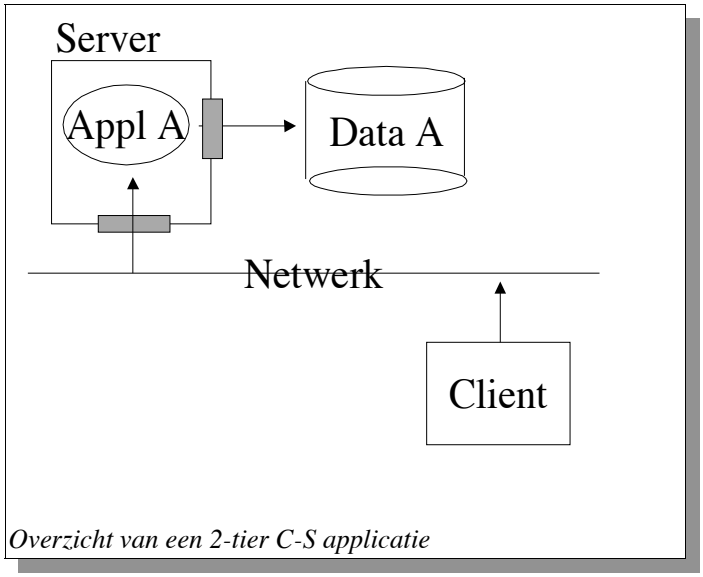
Beschikbaarheid%	*Tijd*
95.000%	18 dagen, 6 uur
98.000%	7 dagen, 8 uur
99.000%	3 dagen, 15 uur
99.900%	9 uur
99.990%	53 minuten
99.999%	5 minuten

Zoals u in bovenstaande tabel kunt zien zakt de ongeplande niet-beschikbaarheid pas onder een dag bij een beschikbaarheidspercentage van 99.9%. OSP gaat ervan uit dat een juist beheerde en correct functionerende HA-cluster een beschikbaarheid van 99.999% kan halen.

Merk op dat met een HA-cluster het verhogen van de beschikbaarheid met minder dan 5% een kostentoeslag van meer dan 200% met zich meebrengt!

3 Anatomie van een applicatie

Om de toepassing van HA-clusters goed te kunnen beoordelen is het van belang om eens te kijken hoe een moderne centrale applicatie eigenlijk functioneert.



Overzicht van een 2-tier C-S applicatie

Een client applicatie (doorgaans op de PC van de gebruiker) richt via het netwerk verzoeken aan de applicatie. Deze verwerkt de verzoeken en maakt daarbij gebruik van applicatiedata die op één of meer disks is opgeslagen.

Heel veel populaire applicaties voldoen aan dit schema:

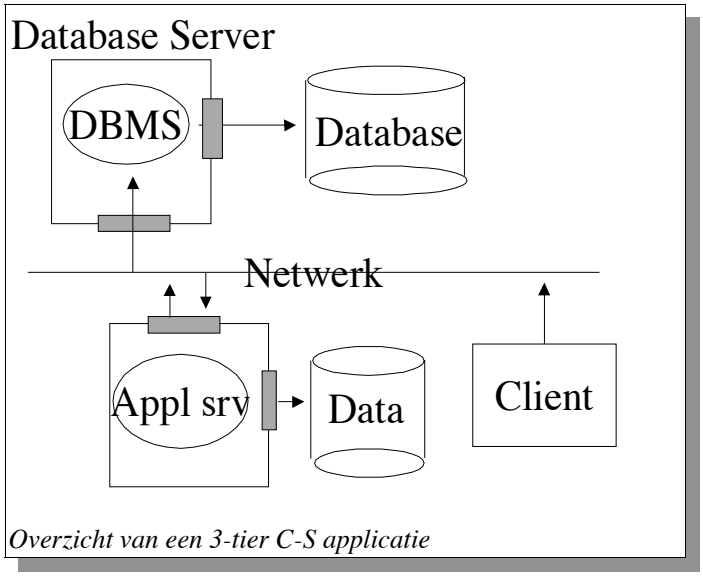
Client applicatie	*Server applicatie*	*Netwerkprotocol*
Web browser	Web server	HTTP
Redirector	File server	SMB, NFS
Algemeen	Oracle RDBMS	Sql*Net

Complexere Client-Server applicaties zijn opgebouwd volgens de "3-tier" architectuur. Hierbij worden drie lagen onderkend:

1. De client
2. De Applicatie Server
3. De Database Server

De clients richten zich via het netwerk tot de applicatie server. Deze is verantwoordelijk voor het afhandelen van alle verzoeken (van alle clients). Als de applicatie server gegevens nodig heeft dan neemt hij over het netwerk contact op met de database server voor het uitvoeren van opvragingen (queries) of mutaties.

Voorbeelden van 3-tier Client-Server applicaties zijn de ERP-oplossingen van leveranciers SAP en BAAN. Ook veel Internet E-Commerce oplossingen zijn opgebouwd als 3-tier applicaties.



Overzicht van een 3-tier C-S applicatie

4 Single Point of Failure

Een losstaande computer configuratie bevat heel veel hardware componenten, ieder van welke cruciaal zijn voor de beschikbaarheid van de configuratie, en daarmee voor de beschikbaarheid van de applicaties die er op die configuratie draaien. Als het uitvallen van één component de uitval van de gehele configuratie bewerkstelligt dan noemen we zo'n component een **Single Point of Failure** (SPOF).

Bij gebrek aan een correcte Nederlandse term voor Single Point of Failure hebben wij ervoor gekozen om hier de Engelse terminologie te handhaven.

In een enkele server omgeving (2-tier Client-Server) zitten een aanzienlijk aantal van die SPOF's, waaronder:

1. Moederbord van de server
2. Disk controller
3. Ieder van de disks
4. Voeding
5. Netwerk interface kaart

In een complexere server configuratie (bijvoorbeeld in een 3-tier Client-Server omgeving) zitten vanzelfsprekend twee keer zoveel van die uitvalpunten, wat de omgeving gevoeliger maakt voor (met name) hardware storingen.

Bij het ontwerpen en bouwen van een HA-cluster trachten we alle "Single Points of Failure" in de configuratie weg te werken. Het uitvallen van één component leidt dan niet langer tot de uitval van de gehele configuratie. Een goed in elkaar gestoken cluster kan zelfs een aantal combinaties van storingen ("double failure") verhelpen.

Het is echter onmogelijk om te garanderen dat alle combinaties van dubbele storingen kunnen worden opgelost.

4.1 HA componenten

Leveranciers van hardware hebben de afgelopen jaren hun producten aanzienlijk weten te verbeteren. Naast enkelvoudige kwaliteitsverbeteringen (producten gaan langer mee, hogere "Mean Time Between Failure") hebben veel producten ook ingebouwde "High Availability" eigenschappen gekregen. Met name de hardware componenten die in de veeleisende kritieke ("high end") server markt worden toegepast hebben vaak een ingebouwde weerstand tegen storingen.

Zo kunnen de "high-end" servers van de leidende merken zelfstandig processors of geheugenbanken uitschakelen als de hardware merkt dat die subcomponenten in ongerede zijn geraakt. Verder hebben veel systemen een ingebouwde dubbele voeding, en kunnen moderne opslagsubsystemen zelfstandig het verlies van een disk of een controller compenseren. Al die verbeteringen dragen bij an de beschikbaarheid van de servers, en daarmee van de applicaties die op die servers draaien.

Al die verbeteringen zijn echter niet gratis. Genoemde "high-end" servers en opslagsubsystemen zijn behoorlijk prijzig. Één van de voordelen van moderne

clustertechnologie is dat het ook met goedkopere componenten mogelijk wordt een hoge mate van beschikbaarheid te bewerkstelligen.

5 Data opslag

Één van de meest relevante aspecten van een computerconfiguratie is de opslagstructuur. De data die met een applicatie wordt bewerkt is in veel gevallen veel kostbaarder dan de gehele configuratie, en terecht wordt daar dan ook veel aandacht aan besteed. Uitval van schijven heeft niet alleen "downtime" tot gevolg, maar kan ook tot verlies van data leiden, waardoor de gevolgen van de uitval nog groter worden.

In dit hoofdstuk gaan we in op de theoretische mogelijkheden om de kostbare data te beschermen tegen schijfuitval.

Onder "data" verstaan we hier ook programmatuur en andere soorten bestanden.

Naast het regelmatig veiligstellen van de data (backup) wordt in vrijwel alle gevallen ook overgegaan tot het toepassen van zogenaamde RAID[1] technologie om data redundant op te slaan. De basis van RAID is dat meer schijven dan strict noodzakelijk worden ingezet om de data meervoudig op te slaan. Bij uitval van een disk drive of controller kan het systeem meestal de data van een andere schijf afhalen of anderszins herstellen; meestal geheel transparant voor de applicatie, de gebruiker en zelfs voor de systeembeheerder.

RAID-opslag van data is een essentieel onderdeel van iedere HA-cluster.

De meest toegepaste RAID-technologieen zijn "Mirroring" (RAID-1) en "Striping met parity" (RAID-5). Deze technologie kan worden gebruikt met "losse schrijven" (JBOD, "Just a Bunch Of Disks), of door het opstellen van een geavanceerd opslagsubsysteem.

Bij de JBOD-methode dient de mirroring en/of striping-met-parity te worden verzorgd door het besturingssysteem of door aparte "volume management" producten die in het besturingssysteem zijn geïntegreerd[2]. Het voordeel van de JBOD methode is de lagere kostprijs van de oplossing, omdat er met gewone schijven en gewone disk controllers kan worden gewerkt. De prestaties van JBOD RAID-oplossingen zijn echter doorgaans iets lager, en het beheer ervan is complexer. Een RAID opslagsubsysteem verdient in principe de voorkeur (want presteert beter en is eenvoudiger te beheren), maar is factoren duurder.

5.1 JBOD Mirroring / RAID-1

Bij mirroring wordt een logisch datavolume twee of meer keer fysiek opgeslagen op verschillende schijven. Iedere kopie van het volume noemen we een "plex".

De termen "volume" en "plex" komt van de Veritas Volume Manager, een populair disk en volume management pakket, wat onder andere onder Digital Unix en Solaris wordt gebruikt. Andere volume management pakketten zoals HP's en IBM's "Logical Volume Manager" gebruiken overeenkomstige terminologie.

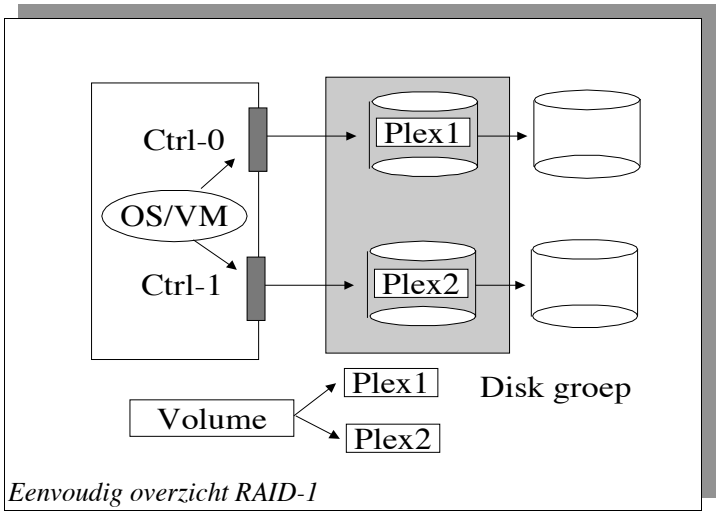
In de normale situatie wordt iedere schrijfactie *n* keer uitgevoerd (met *n*>1), en kan een leesactie worden uitgevoerd van een willekeurige plex. Één van de bijwerkingen van mirroring is dan ook dat de prestatie van schrijfoperaties daalt, terwijl die van leesoperaties stijgt (omdat het besturingssysteem bij lezen natuurlijk de schijf selecteert waar hij de data het snelst vanaf kan halen, i.c. de schijf met de kortste wachtrij).

1 RAID - Redundant Array of Inexpensive Disks

2 Verderop in dit hoofdstuk gaan we in op deze "Volume Managers"

Bij het opzetten van de mirroring moeten de plexen natuurlijk zodanig over de schijven worden verdeeld dat bij uitval van een schijf er altijd minstens één complete plex overblijft. Het besturingssysteem (of de volume manager) zal automatisch alle lees- en schrijfacties naar de overgebleven plex(en) dirigeren.

In een dergelijke configuratie zijn de schijven ook bij voorkeur achter verschillende disk controllers geplaatst en staan de plexen op schijven achter verschillende controllers. Valt dan namelijk een disk controller uit dan kunnen de overgebleven plexen via de overgebleven controller(s) worden benaderd.



Eenvoudig overzicht RAID-1

De inhoud van een volume doet voor de volume manager niet ter zake: het kan een file systeem zijn, een swap gebied of een database. Bij het opzetten van de volumes moet goed rekening worden gehouden met de plaatsing ervan over de fysieke schijven. Alle volume managers ondersteunen het concept "disk groep" wat assisteert bij automatische plaatsing van plexen over schijven. Veel volume managers ondersteunen 3-weg of meer mirroring waarbij van een volume meer dan twee plexen worden opgeslagen. Naast leesperformance voordelen kan dit ook voordelen met zich meebrengen ten aanzien van de back-up. Hierop gaan we verderop in dit boekje in.

Vanzelfsprekend is het ook mogelijk om plexen van verschillende volumes te mengen op de schijven in een disk groep. Hieraan zijn echter eventueel performance consequenties verbonden. Deze vallen echter buiten het bestek van deze discussie.

Bij de plaatsing van plexen over schijven worden veel vergissingen gemaakt. Een verkeerd geplaatste plex kan tot gevolg hebben dat uitval van een enkele disk of controller het gehele volume niet-beschikbaar maakt en kan zelfs tot dataverlies leiden!

Een niet te onderschatten aspect van mirroring is de synchronisatie van de plexen na een systeem- of disk crash. Als zo'n crash gebeurt terwijl de volume manager een schrijfactie uitvoert naar de plexen dan kan het gebeuren dat de ene plex al is bijgewerkt, terwijl de andere schijf nog niet aan het schrijven was toegekomen. De plexen zijn dan ongelijk, en dit is natuurlijk een zeer ongewenste situatie. De volume managers adresseren dit probleem door een resynchronisatie uit te voeren. Afhankelijk van de situatie kan het echter zijn dat enige assistentie van de systeembeheerder noodzakelijk is.

5.2 JBOD Striping met parity / RAID-5

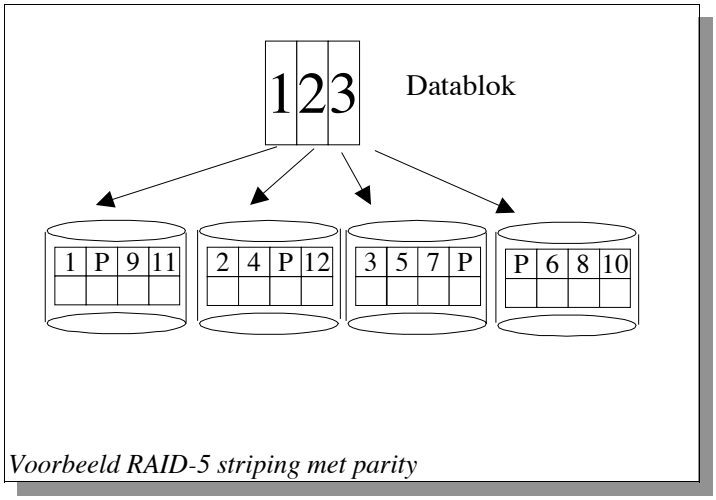
Één van de consequenties van RAID-1 is dat er minstens een dubbele hoeveelheid schijfruimte nodig is. Met een iets andere technologie kan een soortgelijke bescherming tegen uitval van schijven en controllers worden bewerkstelligd met een lagere overhead.

Bij RAID-5 (striping met parity) wordt een volume verdeeld over een *n* aantal schijven. Ieder blok wat naar het volume wordt geschreven wordt over *n-1* schijven verdeeld, en op de n^{de} schijf wordt een blok weggeschreven met daarin herstelinformatie. Dit laatste blok wordt (overigens technisch gezien onterecht) het "parity" blok genoemd. Zo wordt bijvoorbeeld een 64 Kb blok verdeeld in 4 keer 16 Kb en een 16 Kb "parity" blok. Bij RAID-5 worden de "parity" blokken met de datablokken om-en-om over de schijven verdeeld.

Bij leesacties vanaf een RAID-5 volume worden normaal gesproken alleen de datablokken gelezen, en wordt het "parity" blok met rust gelaten.

Bij uitval van een schijf zijn een *x* aantal data- en parity blokken verloren gegaan. Als nu informatie van de uitgevallen schijf moet worden gelezen wordt in plaats daarvan de overige datablokken en het parity blok opgehaald. Via een wiskundige methode kan de data in het ontbrekende blok worden bepaald uit de nog beschikbare datablokken en de parity informatie[3].

Niet alle volume managers ondersteunen RAID-5 volumes.



Voorbeeld RAID-5 striping met parity

Aan RAID-5 zitten diverse performance consequenties. Aan iedere schrijfactie zit enige vertraging omdat het parity blok moet worden uitgerekend. Daarnaast vereist het optimaliseren van de performance een goed inzicht in de wijze waarop het besturingssysteem en/of de applicaties data naar het volume wegschrijft. Bij een onhandige keuze voor de blokgrootte in het RAID-5 volume kunnen de prestaties van het volume aanzienlijk kelderen!

Microsoft stelt in de documentatie van Windows NT dat een schrijfactie naar een RAID-5 partitie drie keer zoveel geheugen vereist als een gewone schrijfactie.

3 Dit lijkt stug maar kan echt!

Bij uitval van een schijf blijft het volume beschikbaar. Wordt een uitgevallen schijf vervangen dan moet er een tijdsintensieve "herbouw" (Engels: "rebuild") operatie worden uitgevoerd die de nieuwe schijf voorziet van de juiste inhoud. Assistentie van de systeembeheerder is ook hier noodzakelijk.

Daar staat echter tegenover dat de hoeveelheid extra benodigde disks geen factor maar een percentage is van het originele aantal.

5.3 Opslagsubsystemen

U kunt uw veilige dataopslag ook voor u laten regelen door een geavanceerd opslagsubsysteem (RAID-array) op te stellen. Deze subsystemen bieden faciliteiten voor mirroring en striping-met-parity in de hardware, zonder dat het besturingssysteem hier iets vanaf weet Voorbeelden van dergelijke geavanceerde opslagsubsystemen zijn die van leveranciers EMC, Artecon, Hewlett-Packard en Data General / Clariion.

Let op! Een aantal leveranciers verkopen disk opslagsystemen onder de naam "array" die in tegenstelling tot wat u zou verwachten geen ingebouwde RAID faciliteiten bieden.

Eigenschappen van dit soort opslagsubsystemen zijn:

- Ingebouwde HA-faciliteiten zoals dubbele controllers en gescheiden voedingen.
- Ingebouwde faciliteiten voor RAID-1 en RAID-5.
- "Hot-pluggable" schijven (kunnen online worden vervangen zonder dat het systeem of de array uitgeschakeld behoeft te worden).

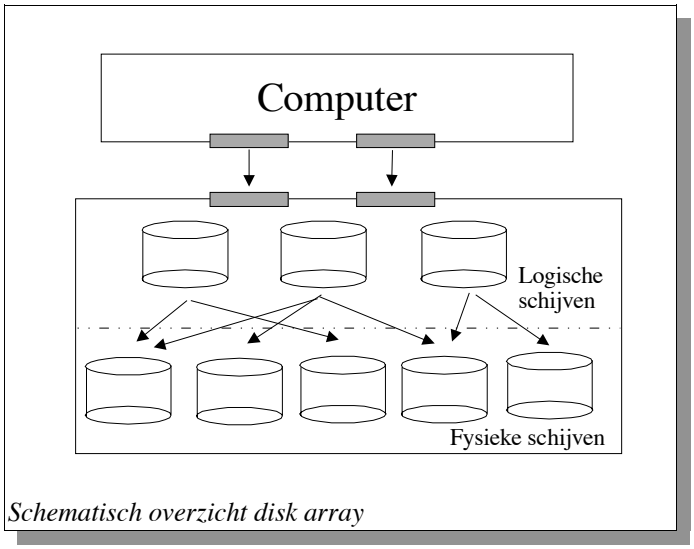
Aangezien bij dit soort opslagsubsystemen de mirroring en/of striping-met-parity geheel intern wordt geregeld is dit volledig transparant voor het systeem (en in enige mate ook voor de systeembeheerder). De array dient uitgebreid te worden geconfigureerd en voorziet in "logische schijven" die intern worden gemirrored of in een RAID-5 constructie zijn opgenomen. Het besturingssysteem ziet de logische schijf in de array als een fysieke schijf, en gebruikt die alsof die schijf rechtstreeks op het systeem zou zijn aangesloten. Indien een fysieke schijf in de array uitvalt wordt dit door de array opgelost zonder dat het systeem daar iets van merkt.

Één van de fraaiste opslagsubsystemen die op dit moment wordt verkocht is de Hewlett-Packard "AutoRaid" array (gebaseerd op technologie van Data General). Deze array is volledig de baas over de interne schijven, en beslist onder andere zelfstandig op basis van disk I/O patronen of een logisch volume via mirroring of via RAID-5 wordt beschermd. Als een extra schijf aan de array wordt toegevoegd balanceert hij zelf de reeds bestaande logische schijven over de fysieke schijven.

Met een dergelijke disk array hoeft u zich geen zorgen meer te maken over de uitval van een schijf. Het kan echter altijd nog gebeuren dat de disk controller waarmee de array aan het systeem is gekoppeld uitvalt. Om dit op te kunnen vangen bevatten de arrays meerdere aansluitpunten, en kan de array via twee paden aan het systeem worden gekoppeld. We noemen dit "Dynamic Multipathing" of "Multiple Physical Volume Links". Het besturingssysteem (of de volume manager) moet dit concept natuurlijk wel in zich herbergen omdat het moet weten dat via de andere controller dezelfde (logische) schijven kunnen worden benaderd.

Bij gebruik van een dergelijke complex opslagsubsysteem worden de logische schijven trouwens "gewoon" met behulp van de volume manager ingedeeld in disk groepen en worden in die disk groepen logische volumes gemaakt. De volume manager hoeft zich echter niet langer druk te maken over mirroring of striping-met-parity, dat wordt immers geheel door de array verzorgd.

Andere diensten die door het opslagsubsysteem kunnen worden verzorgd zijn snapshot kopieën van een logische schijf (kan handig zijn bij het maken van een back-up) en het online synchroniseren van een tweede array met de eerste (kan handig zijn voor uitwijkdoeleinden).



Schematisch overzicht disk array

5.4 Volume Managers

Naarmate schijven steeds groter worden, en het gebruik ervan complexer wordt, bleek dat besturingssystemen vaak tekort schoten in mogelijkheden om de ruimte op die schijven efficient in te delen en te administreren. Traditioneel worden schijven in zijn geheel aan het besturingssysteem ter beschikking gesteld om bestanden op te plaatsen of om te gebruiken als overloopgebied voor het interne geheugen ("swap space" of "page files").

Om administratieve redenen bleek het echter steeds vaker gewenst om een schijf op te delen in meerdere logische schijven, of om juist twee fysieke schijven als één logische schijf te zien. Het eerste werd gefaciliteerd door schijven in partities (logische deelschijven) te kunnen opdelen. Vrijwel alle besturingssystemen ondersteunen dan ook wel een of ander model voor partitionering. Deze partitionering is echter meestal statisch: eenmaal gemaakt is het niet eenvoudig om de partitie te vergroten, te verkleinen, of (deels) op een andere schijf te leggen. Daarnaast moeten deze statische partities meestal aaneengesloten op een schijf liggen, wat versplintering van schijfruimte kan veroorzaken. Zo kan het bijvoorbeeld gebeuren dat er in principe voldoende vrije ruimte op een schijf ter beschikking is, maar dat die ruimte niet aaneengesloten is en daardoor niet kan worden gebruikt om één nieuwe partitie te maken.

De wens om dynamischer met partities om te gaan heeft ertoe geleidt dat steeds meer besturingssystemen zijn uitgerust met een apart disk management subsysteem: de zogenaamde "Volume Managers".

Leveranciers als Hewlett-Packard, IBM, SGI en CompaQ leveren met hun variant van Unix een ingebouwde volume manager mee (de "Logical Volume Manager" of de "XVM Extended Volume Manager" (SGI)). Onafhankelijke software leverancier Veritas heeft een uitgebreid Volume Manager pakket ontwikkeld wat op meerdere systemen kan draaien (de "VXVM Veritas Extended Volume Manager"). De functionaliteit van deze Volume Managers is min of meer overeenkomstig.

5.4.1 Volume Manager functionaliteit

De taak van Volume Managers is om disk ruimte te ordenen en die als dynamische partities, logische volumes genaamd, ter beschikking te stellen. De rest van het besturingssysteem ziet een logisch volume als een schijf, en kan die dus gebruiken om bestanden, databases of swap ruimte op te plaatsen. Voordelen van het gebruik van Volume Managers zijn:

- Een logisch volume hoeft niet aaneengesloten op dezelfde schijf te liggen. De Volume Manager kan de segmenten ("extents" of "subdisks") van het logische volume over meerdere schijven verdelen.

- Logische volumes kunnen worden vergroot en verkleind. Indien de specifieke bestandssysteemtechnologie het toestaat kan dit zelfs on-line, zonder dat het logische volume tijdelijk hoeft te worden afgekoppeld!

- Logische volumes kunnen on-line van schijf naar schijf worden verhuisd (om bijvoorbeeld een te vervangen schijf vrij te spelen).

- Volume Managers bieden softwarematige ondersteuning voor mirroring (RAID-1) en soms voor RAID-5 (zonder dat daar bijzondere hardware voor nodig is). Om deze reden is het gebruik van Volume Managers in High Availability omgevingen dus meestal min of meer verplicht (zelfs als er gebruik wordt gemaakt van een geavanceerde disk array verdient het gebruik van een Volume Manager veruit de voorkeur gezien de andere voordelen die het met zich meebrengt).

5.4.2 Disk groepen

Één van de grondslagen van Volume Managers is dat fysieke schijven worden gegroepeerd in zogenaamde disk groepen (ook wel "Volume Groups"). Disk groepen worden gebruikt om de beheerder nog enige controle te geven over waar delen en plexen van logische volumes terecht komen. Een logisch volume wordt gemaakt in een disk groep, en de Volume Manager zorgt ervoor dat de blokken van het logische volume worden verspreid over de schijven in die disk groep.

Bij gebruikmaking van een Volume Manager is de disk groep de eenheid waarmee schijven aan andere systemen kunnen worden verbonden. Het heeft immers geen zin om slechts één schijf uit een disk groep te verplaatsen, want op die ene schijf staat slechts een gedeelte van een gedeelte van de logische volumes die er in de disk groep zitten.

5.4.3 Volume Managers en disk arrays

Als er gebruik wordt gemaakt van een geavanceerde disk array dan zitten er tussen het bestandssysteem en de fysieke schijf twee abstractieniveau's:

1. De disk array combineert fysieke schijven in logische schijven, en biedt onder mirroring en

eventueel ook RAID-5 mogelijkheden.

2. De logische schijven van de disk array zijn de "fysieke" schijven van de Volume Manager die weer in disk groepen worden verdeeld. Eventueel kan ook de Volume Manager mirroring en/of RAID-5 bieden.

In een dergelijke omgeving wordt er vrijwel altijd voor gekozen om de mirroring/RAID-5 door de disk array te laten verzorgen. De Volume Manager blijft dan verantwoordelijk om logische schijven in groepen te verdelen en beheert de ruimte in die disk groepen.

Alvorens wordt overgegaan tot de inrichting van een dergelijke omgeving is natuurlijk een goed plan van aanpak noodzakelijk.

---

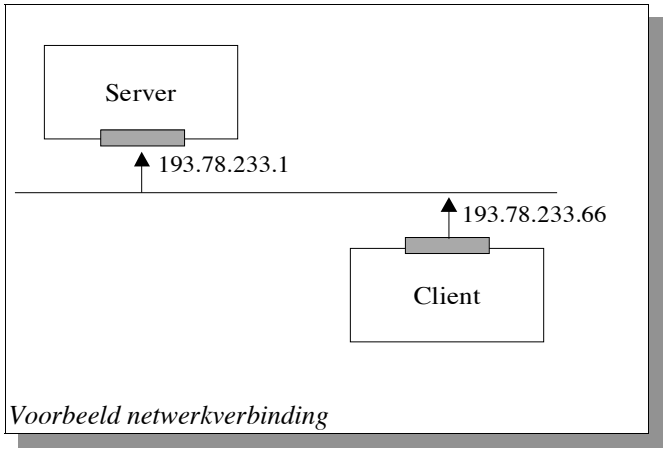
6 Netwerk failover

Een andere component die essentieel is in een client-server omgeving is de verbinding tussen de applicatie en het netwerk. Het moge duidelijk zijn dat als de server niet meer met het netwerk verbonden is, de applicatie (alhoewel deze misschien nog wel "draait") niet meer beschikbaar is. Het is dus zaaks ook dit gedeelte van de server verhoogd beschikbaar te maken.

Vrijwel alle moderne client-server applicaties maken gebruik van een op TCP/IP gebaseerd applicatieprotocol. Voor zover er nog andere protocollen worden toegepast zijn die hard op hun retour ten faveure van het op open standaarden gebaseerde TCP/IP. De unieke identificatie van de server in het netwerk wordt gevormd door zijn IP-adressen: 4-bytes logisch netwerkadressen. De meest gebruikelijke notatie van een IP-adres is vier getallen in de reeks 1 tot en met 255, gescheiden door punten, bijvoorbeeld: "193.78.233.1".

Naast IP-adressen worden in een TCP/IP netwerk ook host namen gebruikt. Deze worden echter door de TCP/IP software onmiddellijk naar een IP-adres vertaald met behulp van een zogenaamde "name server" of via een lokale configuratie file. Voor de netwerkverbinding is het van geen enkel belang of de server initieel via een IP-adres of via een host naam is geïdentificeerd.

De verbinding tussen de server en het netwerk wordt verzorgd door een interface kaart, die bemiddelt tussen het interne geheugen en de netwerkbekabeling. Een IP-adres vormt een unieke identificatie van een interface in het netwerk. Door een IP-adres (of een met dit IP-adres verbonden host naam) te gebruiken selecteert de client met welke server hij wil communiceren.



Voorbeeld netwerkverbinding

Als hetzij de interface kaart van de server, hetzij de kabel tussen de interface kaart en de rest van het netwerk in ongerede raakt kan de client computer niet langer met de server communiceren. Zoals leverancier Sun het stelt: "The network is the computer".

> Configuraties ter waarde van miljoenen guldens zijn verbonden met kabeltjes van nlg. 7,50!

Om dit probleem op te lossen dienen er twee maatregelen te worden genomen:

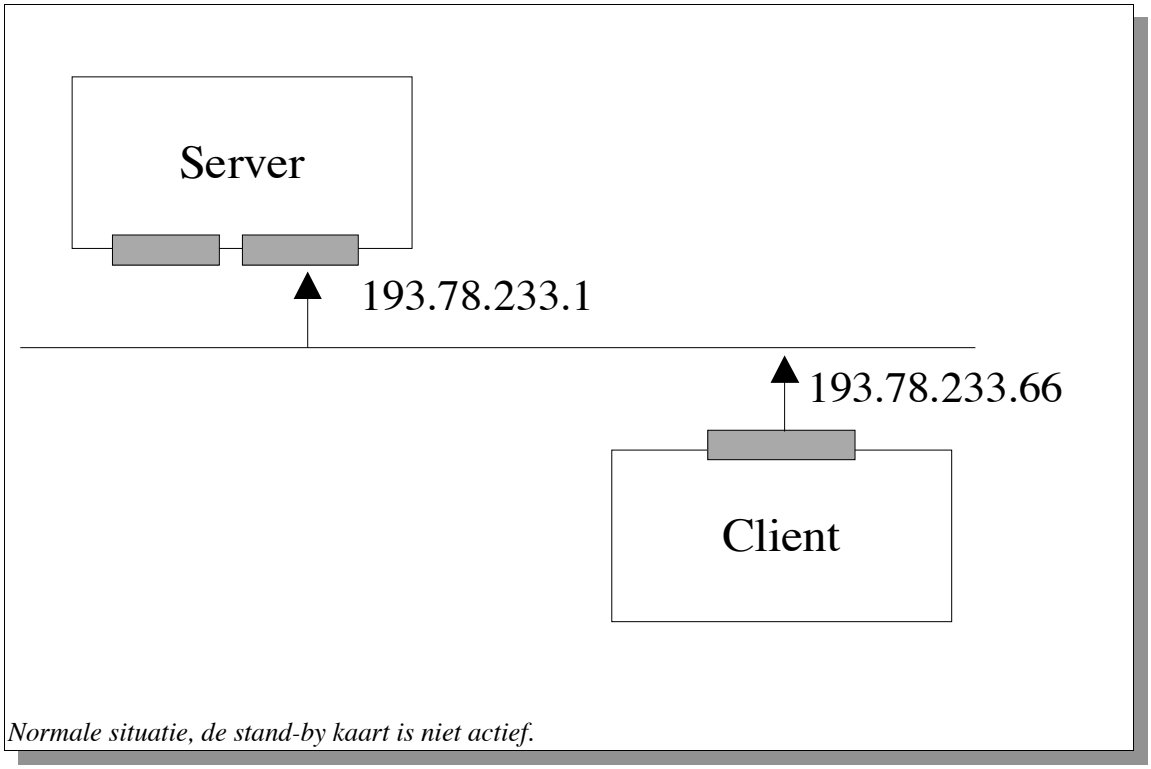
1. In de server dienen extra netwerkkaarten te worden opgenomen die als stand-by

interface kunnen dienen.

2. De verbinding tussen die kaarten en het netwerk dient over gescheiden componenten te lopen.

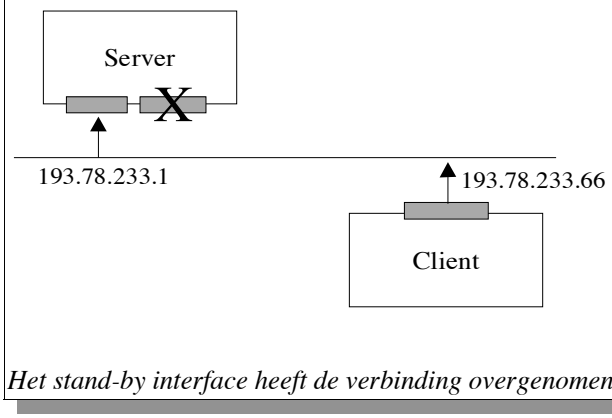
6.1 Stand-by netwerkkaart

Door in de server een tweede netwerkkaart op te nemen kunnen we uitval van het primaire netwerk interface opvangen.



Normale situatie, de stand-by kaart is niet actief.

Software in de server monitort het actieve interface, en zodra hij ziet dat dit interface is uitgevallen configureert deze software onmiddellijk de IP-adressen van de primaire kaart op het stand-by interface. De verbindingen worden dan voortgezet via het nieuwe interface. Door de wijze waarop TCP/IP in elkaar steekt heeft dit geen gevolgen voor de verbindingen. Bezijden een korte vertraging merkt de applicatieprogrammatuur op de client en op de server niets van de overname.



Het stand-by interface heeft de verbinding overgenomen

Technische noot!

De stand-by interface kaart heeft natuurlijk een ander MAC adres dan de primaire kaart. Na de overname zendt de server dan ook een ARP broadcast voor zichzelf uit, die hij eveneens zelf beantwoordt. Alle nodes in het subnet worden dan geacht hun ARP-tabellen bij te werken met het nieuwe MAC-adres. Oudere DOS en Windows clients doen dit echter niet, en herkennen de nieuwe interface kaart dan ook niet. Dit kan worden verholpen door deze oudere clients in een apart subnet achter een router te zetten.

Deze netwerk "failover" faciliteit wordt doorgaans niet standaard door het besturingssysteem geboden, maar wordt toegevoegd door de HA-software van de leverancier. Hewlett-Packard's MC/ServiceGuard biedt deze faciliteit bijvoorbeeld in combinatie met een netwerk "bridge". Sun's Enterprise Cluster levert een faciliteit mee met de naam "Public Network Management" die de failover tussen netwerkkaarten regelt.

6.2 Gescheiden netwerken

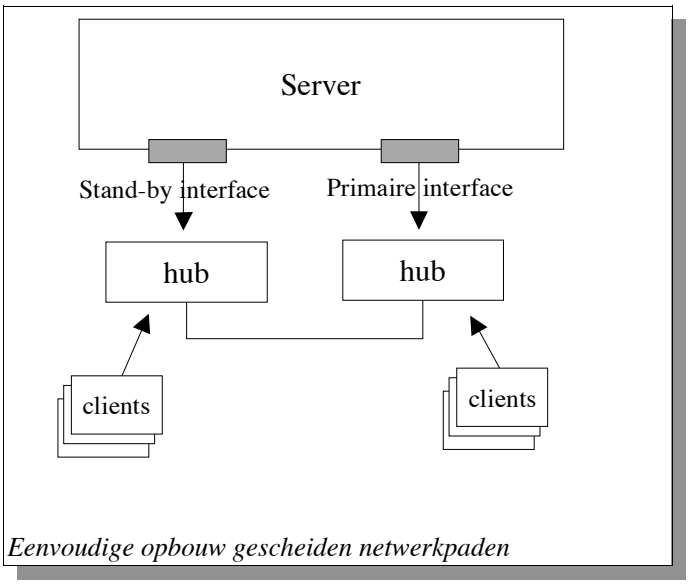
In het geval dat wordt overgegaan op het plaatsen van een stand-by netwerkkaart verdient het ook aanbeveling om de interconnectie tussen de server en het netwerk via twee gescheiden netwerken te laten lopen. Deze filosofie kan heel ver gaan en leidt ultimo tot het aanleggen van een volledig redundant netwerk tot en met de werkplekken! Of dit verstandig is hangt natuurlijk af van de situatie. In deze sectie laten we alleen de eerste stappen van deze opbouw de revue passeren.

Hedendaagse netwerken zijn meestal opgebouwd uit hubs en/of switches. De traditionele coax-kabel heeft in veel gevallen reeds het veld moeten ruimen. De verbinding tussen een interface kaart en de hub/switch wordt gerealiseerd met een point-to-point netwerkkabel, in de meeste gevallen een zogenaamde UTP[4] kabel. Het netwerkapparaat waarmee de interface kaart is verbonden kan uiteenlopen van een "domme" hub (die in feite niets meer is dan een eenvoudige electrische versterker) tot en met een zeer ingewikkelde switch die allerlei redundantie en HA-mogelijkheden in zich herbergt.

In het geval van een redundante netwerkkaart moeten we er echter voor zorgen dat de beide netwerkkaarten met een andere hub/switch verbonden zijn. Deze hubs dan wel switches kunnen dan weer hetzij rechtstreeks, hetzij via weer andere netwerkapparaten met elkaar verbonden zijn.

4 Unshielded Twisted Pair

Bij een simpele opbouw is in het ergste geval de helft van de clients niet meer in staat om met de server te communiceren. Via complexere maatregelen kan een betere redundante interconnectiviteit worden gerealiseerd. Het zou echter te ver voeren daar nu op in te gaan.



Eenvoudige opbouw gescheiden netwerkpaden

7 HA-clusters

In de voorgaande hoofdstukken hebben we gezien hoe we door het dubbel uitvoeren van componenten als schijven, controllers en netwerkkaarten ons konden beschermen tegen het uitvallen van die componenten. Voor één situatie hebben we echter nog geen oplossing: het uitvallen van de gehele server, bijvoorbeeld ten gevolge van een fatale storing op het moederbord.

Het ligt voor de hand ook deze storingen aan te pakken door het meervoudig uitvoeren van deze componenten. De mogelijkheden hiertoe zijn echter beperkt, en we moeten daarom een meer grofstoffelijke aanpak hanteren: het opstellen van meerdere computers en die op zodanige wijze groeperen, verbinden en configureren dat ze elkaars applicaties kunnen draaien. We noemen een dergelijke groep van systemen een "High Availibility cluster" (of kortweg, "cluster").

> Het woord "cluster" wordt ook gebruikt in andere contexten waarin groepen van systemen samenwerken, zoals bijvoorbeeld in performance clusters en rekenclusters.

Een HA-cluster bestaat uit twee of meer nodes. Uit hoeveel nodes de cluster maximaal kan bestaan hangt af van de specifieke cluster oplossing. Zo limiteert de oplossing van leverancier Sun[5] een cluster tot maximaal vier nodes, terwijl HP's MC/ServiceGuard tot en met acht nodes in een cluster kan accommoderen.

> Verreweg de meeste opgestelde clusters bestaan uit twee nodes.

In de cluster draaien één of meer applicaties, ieder op hun "thuisnode". Zodra een systeem uitvalt zorgt de cluster software ervoor dat de applicaties die op de uitgevallen node draaiden verhuizen naar een back-up node en aldaar opnieuw worden opgestart. Als de gebruiker van die applicatie (hetzij een mens, hetzij een programma), merkt dat de verbinding wordt verbroken is het de bedoeling dat deze tracht de verbinding opnieuw op te bouwen. Dat de applicatie intussen op een andere node draait is (als het goed is) transparant. We gaan verderop in dit hoofdstuk dieper in op de mechanismen die hieraan ten grondslag liggen.

> In tegenstelling tot wat vaak wordt gedacht is een applicatie "switch" dus niet transparant voor de gebruiker. De verbinding wordt verbroken en moet opnieuw tot stand worden gebracht. Één van de eisen aan de client applicatie is dan ook dat die bij een verbroken verbinding deze tracht te herstellen.

Iedere applicatie heeft door de systeembeheerder een lijst van nodes geconfigureerd gekregen waarop die applicatie kan en mag draaien. Één van die nodes is de primaire, of "thuis" node van de applicatie. De overige nodes worden "back-up" of "adoptive" nodes genoemd.

7.1 Opslagarchitectuur

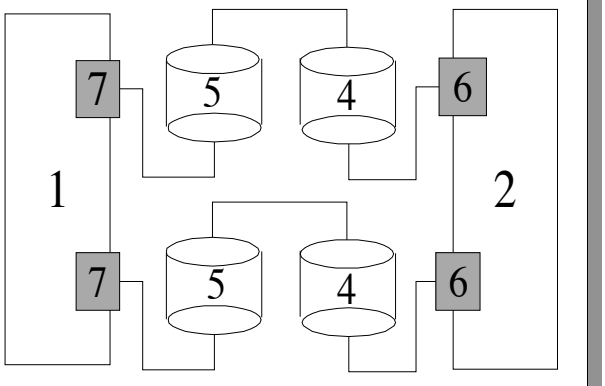
Één van de bijzondere eisen die aan een cluster worden gesteld is dat meerdere nodes in de cluster bij dezelfde schijven kunnen komen. Als namelijk een applicatie van de ene naar de andere node verhuist is het wel zo practisch als de nieuwe node ook bij de data van die applicatie kan komen. Dit is alleen mogelijk indien de schijven waarop die data staat van meerdere kanten kan worden benaderd. In het geval we een applicatie op drie verschillende nodes willen draaien is het ook nodig om die schijven van drie kanten te kunnen benaderen.

5 Sun Enterprise Cluster

7.1.1 SCSI schijven

De meest eenvoudige disk-oplossing voor clusters wordt geboden door standaard SCSI schijven. Iedere SCSI schijf bevat twee connectoren. Via de ene connector wordt de schijf verbonden met zijn voorganger, en via de andere met zijn opvolger. De eerste schijf wordt vanzelfsprekend verbonden met de SCSI controller in het systeem. Normaal gesproken wordt de connector op de laatste schijf met een speciale "terminator" (een afsluitweerstand) afgesloten. In een cluster wordt de laatste schijf echter verbonden met een andere node in de cluster!

Technische noot Dit betekent dat de SCSI controller van die andere node op een ander SCSI-id moet worden gezet, bijvoorbeeld 5 in plaats van 7. Dit heeft dan vervolgens weer consequenties voor de SCSI-id's die aan de schijven in de SCSI-ketting kunnen worden uitgedeeld. Hier worden veel vergissingen mee gemaakt.



SCSI schijven in een HA-cluster

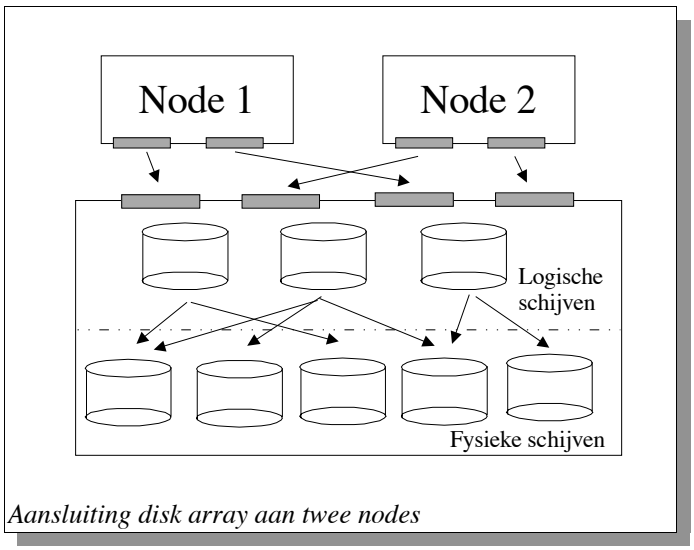
Een belangrijke beperking bij SCSI in een HA-cluster is dus dat de schijven maar tussen twee systemen kunnen worden gedeeld. Door toepassing van een speciale SCSI Y-kabel (ook wel V-kabel genoemd) kan een serie schijven eventueel tussen drie nodes worden gedeeld, maar dit wordt niet door alle leveranciers ondersteund. Verder is het correct configureren (en met name termineren (afsluiten)) van een dergelijke SCSI-configuratie geen sinecure.

Bovenstaande getekende configuratie is tamelijk standaard, en wordt dan ook door alle leveranciers ondersteund. Bij de inrichting van het systeem en de applicatie dient er rekening mee te worden gehouden dat alle programmatuur, data en configuratie die nodig is voor de applicatie hetzij dubbel is aangebracht (op beide nodes), danwel dat die op de gedeelde schijven staat.

7.1.2 RAID disk arrays

Niet zelden worden HA-clusters gebouwd met de geavanceerde HA-opslagsubsystemen die in een eerder hoofdstuk zijn beschreven. De grotere disk arrays hebben meer dan twee connectiemogelijkheden, zodat zonder al te veel problemen meer dan twee systemen kunnen worden aangesloten. De echte "high-end" disk arrays ondersteunen meer dan 8 systemen op één array!

De verbinding tussen de systemen en de array zijn meestal gebaseerd op SCSI, maar kunnen ook op andere wijze zijn geïmplementeerd. Een andere tamelijk populaire technologie wordt gevormd door optische verbindingen. Het betreft hier echter meestal leverancierafhankelijke technologieën.



Aansluiting disk array aan twee nodes

Doorgaans hebben dit soort disk arrays meerdere connectiemogelijkheden die verbonden zijn met interne controllers. Om optimaal gebruik te maken van de HA-voorzieningen van de array worden de verbindingen met de nodes meestal gekruist om ook bij uitval van een array controller nog tenminste een pad naar de logische schijven in de array over te houden.

7.1.3 "Optische" schijven

Veel leveranciers hebben naast SCSI-schijven ook zogenaamde "optische" schijven in hun productaanbod. "Optisch" slaat hier niet op de opslagmethodiek (die is onveranderlijk magnetisch van aard), maar op de verbinding tussen de systemen en de schijven. De voordelen van glasfiber zijn evident: het is sneller en minder gevoelig voor magnetische verstoringen. Om die reden is het in veel omgevingen te prefereren boven SCSI interconnectie.

De "optische" technologie kent zijn eigen specifieke praktische aspecten ten aanzien van het gebruik in een HA-cluster. Aangezien de technologie echter leverancier-specifiek is valt er in zijn algemeen weinig over te zeggen. Om die reden wordt hier niet verder ingegaan op deze technologie.

7.1.4 System disks

Een onderwerp wat tot nu toe buiten beschouwing is gebleven tijdens de behandeling van de opslagstructuur is de plaats van de systeemschijven. Ieder computersysteem heeft één of meer schijven waarop onder andere het besturingssysteem en aanverwante applicaties staan geïnstalleerd. Op deze schijven staan meestal ook de "swap" partities die door het systeem worden gebruikt in geval er (tijdelijk) meer geheugen nodig is dan er fysiek in de computer is geïnstalleerd. Uitval van een systeemschijf heeft vrijwel altijd ook uitval van de computer tot gevolg, en om die reden dienen de systeemschijven ook te worden beschermd via mirroring of iets dergelijks.

Over de fysieke architectuur en plaatsing van de systeemschijven bestaat tussen de diverse geleerden nogal wat verschil van mening. Een en ander is bijvoorbeeld afhankelijk van de opslagarchitectuur die is gekozen voor de applicatieschijven en de specifieke volume manager die wordt gebruikt in het cluster. Mijn voorkeur gaat er meestal naar uit om de systeemschijven niet in de gedeelde diskconfiguratie op te nemen maar deze op aparte SCSI controllers te plaatsen. Om ook voor de systeemschijven de nodige beveiliging te kunnen realiseren adviseer ik om deze dubbel uit te voeren (inclusief de controller) en mirroring toe te passen. Door diverse systeembeperkingen kunnen de systeemschijven meestal niet met software-RAID5 (geïmplementeerd door de Volume Manager) worden beschermd.

7.2 Netwerkstructuur

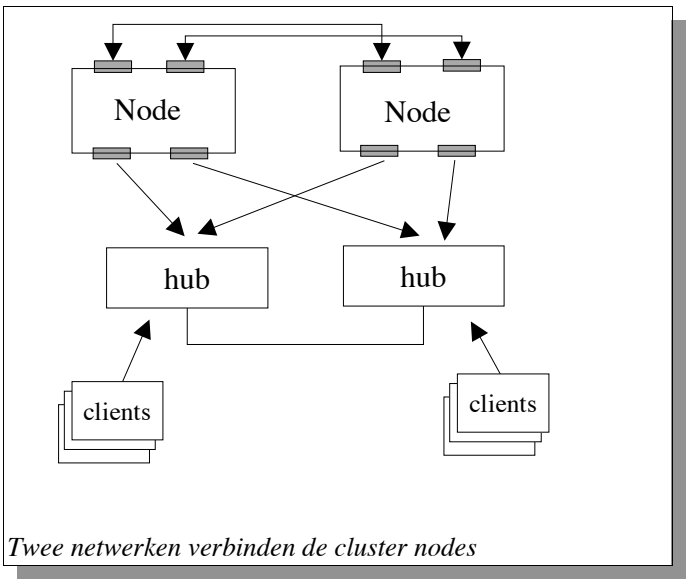
Netwerken worden in clusters voor twee doeleinden gebruikt:

1. Om de clients toegang te bieden tot de applicaties die in de cluster draaien.

2. Om de cluster software op de cluster nodes onderling te laten communiceren.

Voor die twee toepassingen worden meestal verschillende netwerken aangelegd. Via het "publieke netwerk" zijn de cluster nodes verbonden met het intranet van de organisatie, en het is via dit netwerk dat clients en server (applicaties) met elkaar communiceren. In het "privé netwerk" zijn doorgaans alleen de cluster nodes opgenomen.

Om tegen uitval van een enkel netwerkinterface beschermd te zijn worden beide netwerken meestal dubbel uitgevoerd, inclusief twee netwerkadapters per netwerk in de cluster nodes.



Twee netwerken verbinden de cluster nodes

De precieze benodigde infrastructuur verschilt per cluster leverancier. Het principe van dubbele uitvoering van kritieke netwerken wordt echter algemeen ondersteund. Zodra één netwerkadapter het begeeft wordt dit onderkend door de cluster software en wordt de communicatie voortgezet over de overgebleven adapter(s).

In sommige gevallen komen we naast deze twee netwerken nog wel eens een derde en/of vierde netwerk in een cluster tegen. Die extra netwerken worden dan gebruikt voor applicatie specifieke doeleinden zoals snelle intra-applicatie communicatie of het maken van een back-up.

7.3 Cluster software

Naast een bijzonder hardware opbouw is een cluster ook uitgerust met speciale software diens taak het is om de cluster in de gaten te houden en in geval van problemen de benodigde herstelacties te nemen. Deze zogenaamde "Cluster Software" is hoog-gespecialiseerde programmatuur die een essentiele rol vervult in de cluster. Voorbeelden van cluster software zijn:

- Hewlett-Packard's MC/ServiceGuard
- Sun's Enterprise Cluster
- IBM's HACMP
- Microsoft's Cluster Server
- Compaq Tru64 Cluster

Deze hoogwaardige software vervult absoluut niet-triviale taken binnen het cluster, en moet innig samenwerken met het besturingssysteem, de Volume Manager en applicaties teneinde een verhoogd beschikbaar informatiesysteem te kunnen leveren. De problematiek van cluster software is dat zij wordt geacht om componenten die niet zijn bedacht om in een HA-omgeving te functioneren aaneen te smeden tot een HA-cluster.

De web sites van voornoemde leveranciers kunnen u meer vertellen over de specifieke ins en outs van deze software.

7.4 Applicaties

De bedoeling van het bouwen van een cluster is natuurlijk om applicaties verhoogd beschikbaar te krijgen. De dubbele uitvoering van systemen, schijven en netwerken dienen er alleen maar toe om een omgeving te realiseren waarbij ieder denkbare enkelvoudige storing niet leidt tot het uitvallen van de applicatie. In het ernstigste geval valt een complete cluster node uit en moet een applicatie in zijn geheel naar een andere node verhuizen.

7.4.1 Applicatiedefinitie

De terminologie van de verschillende leveranciers loopt nogal uiteen. De een spreekt over "packages", de ander over "logical hosts" en "data services". Wij trachten hier de achterliggende concepten zo neutraal mogelijk te beschrijven, en hanteren daarom de term "applicatie" voor een stelsel van processen en faciliteiten die verhoogd beschikbaar moeten worden gemaakt.

De definitie van een applicatie in een cluster omvat:

- Een startcommando waarmee de cluster software de applicatie kan starten.

- Een stopcommando waarmee de cluster software de applicatie kan stoppen.

- Meestal ook een sonde (probe) via welke de cluster software de status van de applicatie kan monitoren.

- Een lijst van disk groepen die bij de applicatie horen (en waar de data en eventueel ook de programmatuur van de applicatie staat).

- Een lijst van bestandssystemen die op die disk groepen staan en die bij deze applicatie horen.

- Een lijst van netwerkadressen (IP-adressen) die bij de applicatie horen. Het is de bedoeling dat clients deze applicatie uitsluitend via deze netwerkadressen benaderen.

- Een lijst van cluster nodes waarop de applicatie kan/mag draaien. Één van die nodes is de primaire node van die applicatie, en de rest zijn de back-up nodes.

De start- en stopcommando's en sondes van een applicatie zijn meestal scripts die voor dat doel door de cluster implementator zijn ontwikkeld. Die scripts ondernemen dan de stappen die nodig zijn om de applicatie te starten, te stoppen of te controleren. Cluster leveranciers verkopen meestal integratiehulpmiddelen waarmee populaire applicaties zoals NFS, Oracle RDBMS en SAP relatief eenvoudig kunnen worden geclusterd. Deze integratie "toolkits" zijn niets meer (maar ook niets minder) als de volledig uitgewerkte start- en stopscripts en eventueel applicatiespecifieke sondes.

> Het ontwikkelen van dit soort scripts is niet triviaal omdat met allerlei uitzonderingsgevallen rekening moet worden gehouden.

Een HA-applicatie valt volledig onder het beheer van de cluster software. Dit heeft onder andere tot gevolg dat de applicatie alleen nog maar mag worden gestart en gestopt via het cluster framework. Het framework biedt commando's waarmee applicaties handmatig kunnen worden gestopt en gestart, maar zal natuurlijk ook zelfstandig beslissen dat het tijd is om een applicatie te stoppen, naar een andere node te verhuizen, en daar weer te starten.

7.4.2 De applicatie start

Als de cluster software besluit om een applicatie op een bepaalde node op te starten (hetzij omdat het daartoe een commando heeft ontvangen, hetzij om dat het op basis van allerlei gebeurtenissen in de cluster heeft besloten dat dit een goed plan is) onderneemt het de volgende activiteiten:

1. De disk groepen die bij die applicatie horen worden geactiveerd en aan deze node gekoppeld.

2. De bestandssystemen op die disk groepen worden gecontroleerd en aangekoppeld (mount).

3. De netwerkadressen die bij deze applicatie horen worden op de netwerkadapters van deze node geconfigureerd.

4. Het startcommando van de applicatie wordt uitgevoerd.

5. Eventuele sondes voor deze applicatie worden gelanceerd.

7.4.3 De applicatie stopt

Indien het cluster framework besluit de applicatie te stoppen worden de tegenovergestelde acties uitgevoerd in omgekeerde volgorde:

1. Sondes worden gestopt.
2. Het stopcommando wordt uitgevoerd.
3. Netwerkadressen worden van de adapters verwijderd.
4. Bestandssystemen worden gesloten en afgekoppeld (unmount).
5. Disk groepen worden gedeactiveerd en afgekoppeld.

7.4.4 De applicatie verhuist

Een applicatie verhuizing is niets meer en niets minder dan het stoppen van een applicatie op de ene node, en het starten ervan op de andere. Applicatieverhuizingen worden meestal ingegeven door:

- Een opdracht van de beheerder.
- Een sonde die aangeeft dat de applicatie niet meergoed functioneert op deze node.
- Het cluster framework zelf als hij waarneemt dat de node waarop de applicatie draait ingestort is (lijkt).

In het laatste geval heeft er op de voormalige node van de applicatie natuurlijk geen nette "stop" plaatsgevonden. Één van de eisen die we aan applicaties stellen die in een HA-cluster moeten draaien is dan ook dat ze succesvol kunnen starten na een "crash". In een volgend hoofdstuk gaan we dieper in op de eisen die aan applicaties moeten worden gesteld willen ze succesvol in een cluster kunnen draaien.

7.5 Cluster lidmaatschap

Aangezien de verhoogde beschikbaarheid in de hier beschreven clusters wordt geregeld door applicatieverhuizing is het van belang dat de cluster software continue bijhoudt welke nodes nog in de cluster zitten en welke niet. Één van de componenten van de cluster software houdt zich dan ook met niets anders bezig dan het bijhouden van wie er (nog) lid is van de cluster. Meestal is dit een apart proces op iedere cluster node.

Iedere node in de cluster zend regelmatig via het privé netwerk signalen uit met de strekking dat de node nog steeds "leeft" (de zogenaamde "heartbeat"). De ze hartslagsignalen worden door alle andere nodes in de cluster opgepikt en die weten zodoende dat de uitzendende node nog steeds in de cluster zit. Het is natuurlijk van belang dat de hartslag van de clusternodes niet wordt opgehouden door netwerk- of systeemvertraging. Om die reden draaien die hartslagprocessen met hoge prioriteit in het systeem en verdient het aanbeveling om de hartslag over een separaat netwerk te laten verlopen. Sommige clusterleveranciers ondersteunen het uitzenden van hartslag over seriële (RS-232, nul-

modem) verbindingen.

Op het moment dat de hartslag van een node stopt concluderen de overgebleven nodes dat die node wel uitgevallen zal zijn. Vervolgens wordt gekeken welke applicaties op die uitgevallen node draaiden, en voor die applicaties worden een verhuizing geïnitieerd naar de overgebleven nodes (conform de applicatiedefinitie in de cluster).

Als de hartslag van een node dan weer aanvangt wordt de conclusie getrokken dat de node weer leeft. Afhankelijk van de clusteroplossing en -configuratie worden dan al dan niet automatisch sommige applicaties weer terugverhuisd naar die node.

Het proces wat het verantwoordelijk is voor het uitzenden en verwerken van hartslagsignalen speelt dus een cruciale rol in een cluster. Als dit proces abusievelijk stopt (hetzij door een programmeerfout, hetzij door een "kill" door de beheerder) kan de situatie ontstaan dat de andere nodes (in strijd met de realiteit) concluderen dat deze node "down" is. Dit is echter niet het geval, de node is "up" en de applicaties draaien nog gewoon.

De andere nodes in de cluster weten dit echter niet, en starten een applicatieverhuizing. Dit betekent dat de applicatie twee keer wordt opgestart, op twee verschillende nodes! Dit kan leiden tot dubbele netwerkadressen en bestandssystemen die vanaf twee nodes tegelijk worden benaderd (==corruptie!). Deze situatie moet natuurlijk ten alle tijde worden voorkomen.

Om die reden zijn er in cluster nodes speciale voorzieningen getroffen die ervoor zorgen dat als één van de cruciale processen van de cluster software ineens stopt, het gehele systeem zo snel mogelijk wordt afgesloten. In Unix geschiedt dit meestal met een door de software geïnitieerde systeem crash (een zogenaamde "panic").

7.5.1 Gespleten clusters

Een situatie die tot veel problemen kan leiden is als alle hartslagnetwerken van een cluster in ongerede raken (een meervoudige storing). Iedere node in de cluster denkt dan namelijk dat hij de enige nog overgebleven node is, en zou theoretisch kunnen proberen alle applicaties naar zich toe te trekken. Dit is natuurlijk niet zo'n goed idee, en de cluster moet dit beter oplossen. Ook kan het voorkomen dat bijvoorbeeld in een drie-node cluster de communicatie tussen node 1 en de rest verstoord raakt. De cluster is dan gesplitst in twee subclusters, eentje met daarin alleen node 1, en eentje met daarin node 2 en node 3.

> In een twee node cluster is het in ongerede raken van de hartslagnetwerken niet te onderscheiden van de situatie waarin één van beide nodes is uitgevallen. De constatering dat van de andere node geen hartslag meer wordt ontvangen kan door beide oorzaken komen!

Hoe het probleem van gespleten clusters precies wordt opgelost is sterk afhankelijk van de specifieke clusteroplossing. Een aantal standaardmechanismen komen we echter in vrijwel al die oplossingen tegen:

7.5.1.1 Lock disks

In het geval de cluster uiteen valt in twee gelijke delen (met name bij een 2-node cluster) wordt meestal gebruik gemaakt van een lock disk. Dit is een verder alleszins normale disk uit de gedeelde disk configuratie die echter bij de configuratie van de cluster is aangewezen om in geval van een gespleten cluster op te treden als scheidsrechter.

In het geval een gespleten cluster wordt vermoedt (geen hartslag meer van andere node(s)) proberen beide subclusters de lock disk te reserveren (met een eigen protocol of een "SCSI RESERVE" opdracht). De subcluster waarvan de reservering slaagt mag doordraaien, en de ander wordt geacht zichzelf zo snel mogelijk uit te schakelen.

In een 2-node cluster wordt dit reserveringsmechanisme dus ook gebruikt als één van de nodes uitvalt, omdat deze gebeurtenis niet is te onderscheiden van het uitvallen van de hartslagnetwerken.

7.5.1.2 Meerderheid van stemmen

Indien een cluster splitst in twee ongelijke delen is het niet ongebruikelijk dat de grootste subcluster doorgaat. De nodes in de kleinere subcluster schakelen zichzelf dan zo spoedig mogelijk uit. Deze situatie doet zich bijvoorbeeld voor bij een 5-node cluster die splitst in een 3-node en een 2-node subcluster.

7.5.1.3 Node x gaat door

Indien een cluster splitst in twee subclusters van gelijke grootte (bijvoorbeeld een 4-node cluster die splitst in twee 2-node clusters), en waar geen "lock disk" is aangewezen, moet een andere grondslag worden gekozen om te beslissen wie er door mag, en wie niet.

Voor dit soort situaties heeft de cluster software meestal een ingebakken (of configureerbaar) beleid, bijvoorbeeld: "de subcluster met daarin de laagst genummerde node mag door".

8 Applicaties

De doelstelling van de in dit boekje beschreven HA-clusters is om applicaties die daar niet expliciet voor zijn ontwikkeld toch een zeer hoge graad van beschikbaarheid te geven. Zonder enige ondersteuning van de applicatie is dat helaas niet mogelijk, de applicatie moet aan een aantal voorwaarden voldoen wil deze met enige mate van succes in een HA-cluster kunnen draaien. In dit hoofdstuk gaan we onder andere in op de voorwaarden waaraan een applicatie moet voldoen om "geclustered" te kunnen worden.

8.1 Eisen aan de applicatie

8.1.1 Herstartbaar

Één van de basiseisen aan een applicatie is dat deze na een crash automatisch weer op kan komen, dus zonder dat daarvoor handmatige acties nodig zijn. De reden daartoe is voor de hand liggend: Als een cluster node uitvalt verhuist de cluster software de applicatie hulpbronnen (disk groepen, bestandssystemen, netwerkadressen) naar een andere node en start de applicatie aldaar. Voor de applicatie is het dan net alsof het systeem waarop hij draait is gecrasht: zijn databestanden zijn in een corrupte staat en moeten wellicht eerst worden gerepareerd alvorens de applicatie verder kan. Verder was de applicatie ten tijde van de crash wellicht bezig met één of meer transacties, die dan moeten worden herstart of teruggerold.

Als dit allemaal niet automatisch kan komt de applicatie niet spontaan weer op na de verhuizing.

8.1.2 Plaats data en configuratie

Om een applicatie succesvol te kunnen clusteren is het verreweg het handigst als alle data en configuratie van die applicaties op de bestandssystemen in de gedeelde disk groepen kunnen worden geplaatst. Dit betekent dat de locaties van die gegevens hetzij configureerbaar moet zijn, of dat we het besturingssysteem zover moeten kunnen krijgen (bijvoorbeeld via bestands- en directory links of aliassen) dat deze de vaste niet-gedeelde locaties doorverwijst naar plaatsen op de gedeelde schijven.

Deze eis geldt met name voor de variabele data (databases) van de applicatie, die tengevolge van applicatie activiteit wijzigen en die ten alle tijd up-to-date beschikbaar moeten zijn. Meer statische configuratie hoeft niet altijd op de gedeelde schijven te staan, maar er moet dan wel worden voorzien in een procedure waarmee die configuratie op alle nodes in de cluster (waar de applicatie kan draaien) gelijkt wordt gehouden.

8.1.3 Afhankelijkheden fysieke node

Een andere, meer verborgen, eigenschap van veel applicaties is dat ze na installatie op één of andere manier afhankelijk zijn van een eigenschap van de fysieke node waarop ze zijn geïnstalleerd, zoals de nodenaam (hostnaam), het fysieke netwerkadres van de node, het CPU-id (bijvoorbeeld voor licenties) of andere nodespecifieke zaken.

Indien een dergelijke afhankelijkheid bestaat is het niet triviaal om deze om te buigen. Veelal zal daar de assistentie van de applicatieleverancier voor nodig zijn.

8.1.4 Opstarttijd

Wellicht een voor de hand liggende voorwaarde, maar het helpt indien een applicatie binnen afzienbare tijd opstart. Het clusteren van een applicatie die een zeer geruime tijd

nodig heeft om op te starten werkt wel, maar is niet altijd even zinvol. In één geval wat wij aan de hand hebben gehad deed een applicatie er meer dan drie kwartier over om te starten. In geval van een verhuizing (bijvoorbeeld ten gevolge van een crash) had de cluster software de applicatie binnen 2 minuten op de andere node gestart, maar was de applicatie pas 45 minuten later weer ter beschikking voor de organisatie!

8.2 Andere applicatie aandachtspunten

Een ander applicatiegerelateerd item wat om de hoek komt kijken bij HA-clusters is waar de programmatuur van de applicatie dient te worden geïnstalleerd. De keuzes zijn: in of buiten de gedeelde disk groepen.

Deze vraag kan alleen worden beantwoordt met voldoende applicatiekennis, en is onder andere ook afhankelijk van hoeveel "instanties" van die applicatie er in het cluster draaien.

In het geval dat een applicatie slechts één keer in de cluster draait als HA-applicatie verdient het meestal (indien mogelijk) de voorkeur om de programmatuur in de gedeelde disk groepen te installeren. De programmatuur verhuist dan automatisch met de applicatie mee naar de nieuwe node, en kan daar meteen worden gebruikt.

Het komt echter soms voor dat een bepaalde applicatie meerdere keren in de cluster draait, wellicht op verschillende nodes. We noemen dit dan "instanties[6]" van die applicatie. Dit komt bijvoorbeeld voor bij relationele database systemen waar we meerdere databases als aparte HA-applicaties beschouwen en separaat heen en weer verhuizen tussen de cluster nodes. In dat geval verdient het meestal aanbeveling om de programmatuur op iedere node op de systeemschijven te installeren (buiten de gedeelde disk groepen). Op de gedeelde disk groepen staan dan alleen de databestanden van die applicatie (instantie).

Waar in dat laatste geval wel rekening mee moet worden gehouden is dat onderhoud op de applicatieprogrammatuur (patches, nieuwe versies) op alle nodes van de cluster worden uitgevoerd!

6 Een anglicisme? "*Instances*"

9 Het beheer van HA-clusters

Een punt dat in onze ervaring te weinig aandacht krijgt van organisaties is het beheer van de opgestelde clusters. De installatie en configuratie van een cluster mag zich meestal in warme belangstelling koesteren, en niet zelden worden daarbij grote hoeveelheden externe technici en adviseurs ingeschakeld.

Draait de cluster eenmaal dan neemt de aandacht ervoor aanzienlijk af, en dat terwijl een cluster andere beheeractiviteiten, -vaardigheden en procedures vereist dan een groep losstaande systemen. De ervaring die wij daarmee in de praktijk hebben is erg slecht, en heeft ertoe geleidt dat wij veronderstellen dat de overgrote meerderheid van de clusters in "het veld" als het erop aan komt niet doet waarvoor ze zijn opgesteld!

Het beheren van een HA-cluster is tamelijk complex, en wel om de volgende redenen:

- De beheerder heeft veel kennis nodig van de meest uiteenlopende onderwerpen: het besturingssysteem de applicatie, netwerken, disk arrays, scripting, en, niet te vergeten, de HA cluster software.
- De cluster kent veel en complexe configuratiemogelijkheden.
- De cluster heeft veel punten waarop dingen fout kunnen lopen ("failure modes").

Niet zelden zien we dat organisaties overstappen naar een nieuw systeem, en dat meteen als HA-cluster uitrusten. De beheerders van die organisaties hebben doorgaans (te) weinig tijd om zich al bovenstaande onderwerpen in afdoende mate eigen te maken. Het resultaat is dan ook dat die beheerders gedurende de eerste tijd dat het cluster actief is er nog niet goed mee om weten te gaan. De kans op fouten is dan (dus) groot.

Tot overmaat van ramp zijn ook externe technici (bijvoorbeeld van de applicatieleverancier) lang niet altijd op de hoogte van de specifieke aandachtspunten van het cluster.

9.1 Beheeraspecten

9.1.1 Dubbele configuraties

Één van de problematische aspecten van het beheren van een cluster is dat diverse configuratie-elementen op alle nodes van de cluster identiek (of, in ieder geval, in overeenstemming met elkaar) moeten zijn geconfigureerd. Een goed voorbeeld daarvan zijn zaken als gebruikers, groepen, printer definities en netwerkpoorten. Dit zijn zaken die op iedere node separaat kunnen worden geconfigureerd, en waarvan de kans bestaat dat die configuraties dus per cluster node verschillend zijn.

Als een applicatie bijvoorbeeld er vanuit gaat dat bepaalde groepdefinities aanwezig zijn (bijvoorbeeld omdat die door de installatieprocedure van die applicatie zijn gecreëerd), is het zaaks om ervoor te zorgen dat dit dan ook op alle nodes in de cluster het geval is. Dit betekent meestal dat na installatie van de applicatie in ieder geval moet worden nagegaan welke elementen in de systeemconfiguratie door de installatieprocedure zijn aangepast, en die aanpassingen ook op de andere nodes in de cluster uit te voeren.

Bij alle beheeractiviteiten op de cluster dient dus te worden nagegaan of aangebrachte configuratiewijzigingen ook op andere nodes moet worden aangebracht. Dit vereist diepgaand inzicht in het besturingssysteem.

Moderne besturingssystemen bieden mogelijkheden om bepaalde configuratie-elementen over meerdere systemen heen gelijk te houden, bijvoorbeeld door ze regelmatig te synchroniseren of ze op te slaan in een centrale database. Voorbeelden van dit soort technologie zijn NIS en NIS+ onder Unix. Het verdient wellicht aanbeveling om dit soort technologie in de cluster te implementeren (en dit natuurlijk op zodanige wijze dat het weer geen "Single Point of Failure" introduceert).

Naast de besturingssysteem configuraties heeft ook de HA-cluster software configuraties die soms op alle nodes moeten worden aangebracht, en soms slechts op één node (afhankelijk van de clusteroplossing). Ook hier is diepgaande kennis van de cluster vereist om te kunnen bepalen welke wijzigingen waar moeten worden aangebracht.

9.1.2 Beheerhulpmiddelen

De meeste besturingssystemen leveren tegenwoordig een of andere interactieve "tool" mee voor het uitvoeren van dagelijkse beheeractiviteiten. Windows NT zit hier natuurlijk vol mee, terwijl ook de Unix leveranciers zich niet onbetuigd laten met programma's als "SAM" (HP-UX), "SMIT" (AIX) en "AdminTool" (Solaris).

Helaas zijn die tools meestal niet "cluster-aware": de wijzigingen die ze aanbrengen worden hetzij op de verkeerde plaats in het besturingssysteem aangebracht, hetzij slechts op één node in de cluster. Tenzij de beheerder precies weet wat hij/zij doet verdient het wellicht aanbeveling om die hulpmiddelen ter zijde te laten liggen in de cluster.

9.1.3 Applicaties

Zoals reeds vele malen vastgesteld zijn de meeste applicaties die we verhoogd beschikbaar willen maken ook niet "cluster-aware": ze zijn niet geschreven om in een cluster te kunnen draaien. Naast de reeds eerder genoemde voorwaarden die we moeten stellen om de applicatie überhaupt in een cluster te kunnen laten draaien dienen we bij het applicatiebeheer ook rekening te houden met het feit dat de applicatie in een cluster draait. Dit kan bijvoorbeeld betekenen dat we applicatieconfiguratiewijzigingen meervoudig moeten uitvoeren, of dat we speciale configuraties moeten aanbrengen om ervoor te zorgen dat de applicatie zich op juiste wijze gedraagt.

9.2 Controle

Vertrouwen is goed, controle is beter.

In onze optiek dient een cluster regelmatig te worden doorgelopen en te worden gecontroleerd op juiste werking. Naast de controle op de juistheid van configuraties vinden wij ook dat de cluster op regelmatige basis moet worden getest. Dit betekent: handmatig applicaties naar andere nodes verhuizen, maar ook op gecontroleerde wijze een systemen, schijven en netwerken uitschakelen en kijken of de cluster reageert zoals verwacht.

Veel organisaties kijken vreemd tegen dit advies aan, "er is toch zojuist een cluster gekocht wat altijd op zou moeten zijn?" Waarom dan die regelmatige controle. De praktijk wijst helaas uit dat een eenmaal werkend cluster relatief eenvoudig weer in ongerede raakt. De controle is bittere noodzaak om het rendement van de investering in de cluster te kunnen oogsten.

Open Solution Providers heeft rondom die controle een unieke dienst ontwikkeld: de "OSP HA-Audit". Clusterspecialisten van OSP controleren dan regelmatig of uw cluster nog in optima forma conditie is, en begeleiden uw eigen beheerders bij het inrichten en uitvoeren van het beheer.

9.3 Er is ook goed nieuws!

Het goede nieuws is natuurlijk dat een HA-cluster, mits goed opgezet en correct beheerd, de beschikbaarheid van applicaties tot grote hoogtes opvoert en tegelijk het beheer van continu-beschikbare applicaties kan vereenvoudigen.

Met een cluster kunt u systeemonderhoud en dergelijke, wat voorheen altijd 's-nachts en in het weekeinde diende plaats te vinden, gewoon overdag uitvoeren. De kritieke applicaties heeft u dan namelijk "gewoon" naar de andere nodes verhuisd. Ook de invoering van nieuwe versies en het aanbrengen van patches kan op die manier worden getest.

10 Literatuurverwijzingen

10.1 Boeken

1. Clusters for High Availability, Peter Weygant, ISBN 0-13-494758-4.

2. In search of Clusters, Gregory F. Pfister, ISBN 0-13-899709-8.

3. Windows NT Cluster Server, David Libertone, ISBN 0-13-096019-5.

10.2 Web Sites

1. http://www.sun.com/clusters

2. http://www.hp.com/go/ha

3. http://www.ibm.com/servers/aix/products/ibmsw/cluster

4. http://www.linux-ha.org

5. http://www.unix.digital.com/cluster/index.html

6. http://www.sgi.com/software/failsafe