OS/390 ... waarom toch?

Achtergronden van de betrouwbaarheid en beschikbaarheid van OS/390-mainframes

Ing. P. Mienes RE

OS/390 en zijn voorganger MVS/ESA staan bij ingewijden bekend als bijzonder betrouwbare, goed beheerste en beheersbare omgevingen. Hoe komt dat toch? Wat maakt deze mainframeomgeving zo uniek? Waarom zijn de verantwoordelijke techneuten met een gemiddelde leeftijd van ruim veertig jaar zo trots op 'hun' systemen? Welke combinatie van factoren maakt juist de OS/390-omgeving bij uitstek geschikt voor de *werkelijk* bedrijfskritische toepassingen? En waarom is het volgens sommigen uit het stenen tijdperk stammende OS/390 een serieuze kandidaat voor Unix-applicaties en webservers?

Historie

In 1964 introduceerde IBM de System/360 (S/360)-architectuur, bedoeld als 'general purpose' computer. Dit houdt in dat de architectuur, en met name de instructieset in de CPU, niet is toegespitst op een bepaald type gebruik, zoals bijvoorbeeld wetenschappelijke berekeningen. S/360 werd in de praktijk voornamelijk toegepast bij het automatiseren van administratieve processen. De apparatuur vond zijn plek bij vrijwel alle banken, (pensioen)verzekeraars, industrieën, handel, energiebedrijven, universiteiten, enkele grote gemeenten en een enkele provincie.
S/360 werd opgevolgd door S/370 en later in de negentiger jaren door de S/390-architectuur. Hierbij is steeds vastgehouden aan het beginsel van 'upward compatibility', waarbij de meeste (assembler)programma's die waren ontwikkeld op de S/360-architectuur ook nog functioneerden op de S/370- en vandaag nog steeds functioneren op de S/390-architectuur. Voor de gebruikers van deze systemen is dit van groot belang gebleken, omdat de programma's of informatiesystemen veel langer meegingen dan ooit voor mogelijk was gehouden.

Op de S/360-, S/370- en S/390-hardwarelijn zijn drie commercieel verkrijgbare operating systems ontwikkeld:
* VSE (Virtual Storage Extended) – voor de kleinere mainframeomgevingen;
* MVS (Multiple Virtual Storage) – voor de grotere mainframeomgevingen, inmiddels opgevolgd door OS/390;
* VM (Virtual Machine) – dit operating system had als bijzondere eigenschap dat het op één fysieke machine verschillende logische VSE- en MVS-machines (-images) kon 'hosten', bijvoorbeeld een productie- en een acceptatie-MVS.

De functionaliteit van VM is bijna overal vervangen door een nieuwe faciliteit in de hardware (sinds begin negentig) om logische partities (LPAR's) te definiëren. Van de genoemde operating systems is MVS (respectievelijk OS/390) vandaag de dag het meest in gebruik in grote omgevingen waar de betrouwbaarheid en beschikbaarheid van het computerplatform van wezenlijk belang zijn. Dit artikel richt zich op OS/390 in samenhang met de S/390-architectuur.

OS/390 is een product van IBM dat alleen kan worden geïnstalleerd op een systeem met de S/390-architectuur. Dergelijke systemen worden geleverd door IBM, Amdahl, Hitachi Data Systems en Comparex.

Betrouwbaarheid en beschikbaarheid

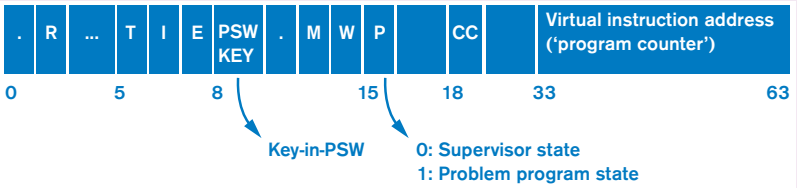
In de introductie is de vraag opgeworpen hoe het toch komt dat OS/390 zo'n betrouwbaar en goed beschikbaar systeem is. Het antwoord op deze vraag kan worden gevonden in de combinatie van:
* architectuur (S/360, S/370, S/390, samen met MVS of OS/390);
* volwassenheid van beheertools;
* volwassenheid van beheerprocedures.

Het is namelijk juist het samenspel tussen deze aspecten dat leidt tot het hoge niveau van betrouwbaarheid en beschikbaarheid. Recente ontwikkelingen in het samenwerken van systemen dragen andermaal bij tot een significante toename van de beschikbaarheid van informatiesystemen tot wel 99,999%.

Een korte anekdote: tien jaar geleden nam ik deel aan een bijeenkomst van MVS-systeemprogrammeurs, waar een grote bank een presentatie verzorgde. Gemeld werd dat de MVS-systemen een beschikbaarheid van 99% bereikten, terwijl de fouttolerante systemen van de bank bij 95% bleven steken. De fouttolerante systemen waarvan nota bene de hoge beschikbaarheid het enige bestaansrecht was, scoorden aanzienlijk slechter! Weliswaar was de bron (de MVS-systeemprogrammeur van de bank) niet geheel onpartijdig, maar deze techneuten in hart en ziel hebben over het algemeen de aardige eigenschap eerlijke mensen te zijn.

Abend	= ABnormal END; wordt op andere systemen ook wel dump of crash genoemd
APF	= Authorized Program Facility
Batch window	= periode dat er in het systeem ruimte wordt gemaakt voor het uitvoeren van de bulk van de batchjobs, in het algemeen gedurende de nacht; veelal zijn de onlinesystemen in deze periode niet actief
Betrouwbaarheid	= vertrouwelijkheid + integriteit + controleerbaarheid
CICS	= transactiemanager (Customer Information Control System)
DASD	= schijf (Direct Access Storage Device)
dataset	= gegevensbestand (file); logische eenheid van ruimteallocatie op DASD of tape
DB2	= databasemanager; relationele database
IMS	= transactie- en databasemanager (Information Management System); hiërarchische database
Loadlibrary	= PDS met loadmodules
Loadmodule	= uitvoerbare code (executable) van een programma
PDS	= partitioned dataset; speciaal type dataset, ook wel library genoemd; een PDS bevat afzonderlijk identificeerbare members ('bestandjes')
Privileged instructies	= instructies die de integriteit van OS/390 kunnen ondermijnen
PSW	= Program Status Word; het register (64 bits) in de CPU dat de voor de CPU relevante gegevens voor een taak bevat
Sourcelibrary	= PDS met sources van programma's
Subsysteem	= (binnen het kader van dit artikel:) programma's die algemene diensten leveren, zoals het Job Entry Subsystem (JES2), transactiemanagers, databasemanagers, netwerkmanagers (VTAM)
Supervisor state	= hoge autorisatie; als een taak in supervisor state wordt uitgevoerd, dan kan die taak alle instructies – ook de privileged instructies – uitvoeren
Taak	= onderdeel van een programma
Tape	= 'ouderwetse' magneetband (reel), maar ook cartridge
User exit routine	= door leverancier geboden mogelijkheid om kleine wijzigingen of uitbreidingen te maken op bijvoorbeeld het operating system; de routine krijgt doorgaans de besturing in supervisor state
User SVC routine	= ongeveer hetzelfde als een user exit routine, echter een User SVC routine krijgt per definitie de besturing in supervisor state
Volume	= één tape of één DASD, geïdentificeerd door een volumelabel van zes tekens
VSAM	= veelgebruikte opslagmethode (Virtual Storage Access Method)

Kader 1. Begrippen OS/390-omgeving



Figuur 1. Het PSW bepaalt welke instructies mogen worden uitgevoerd en welke delen van het geheugen mogen worden geadresseerd.

Dit artikel beschrijft op hoofdlijnen de controls die in de S/390- en OS/390-architectuur zijn aangebracht om de betrouwbaarheid en beschikbaarheid van informatiesystemen en gegevens te waarborgen. Daarna wordt aangegeven hoe beheertools en -procedures worden toegepast om eveneens een bijdrage te leveren aan die betrouwbaarheid en beschikbaarheid. Afsluitend zal worden ingegaan op een ontwikkeling sinds medio jaren negentig: Parallel Sysplex. Bij de bespreking van de beheertools zullen ook enkele van de meest in het oog springende functies worden besproken, zodat een meer compleet beeld wordt geboden van de beheeraspecten in een OS/390-omgeving. Hierbij komen beheertools van de volgende leveranciers aan de orde: IBM, Computer Associates, BMC, Candle, Landmark, Beta Systems Software en Consul Risk Management.
Om het artikel leesbaar te houden voor degenen die al (enigszins) bekend zijn met het platform OS/390 zullen de in deze omgevingen gangbare termen en afkortingen worden gebruikt. Deze worden in kader 1 eerst toegelicht voor de overige lezers.

Architectuur

Instructieset

In de S/390-architectuur zijn de instructies verdeeld in twee categorieën: privileged en unprivileged instructies. Glashelder is bepaald wie de privileged instructies mogen uitvoeren: alleen de taken die in Supervisor State draaien, ofwel de taken die in het PSW het Problem State bit op 0 hebben staan. Taken met het Problem State bit op 1 mogen alleen de unprivileged instructies uitvoeren (zie figuur 1).
Unprivileged instructies zijn ongevaarlijk, dat wil zeggen dat er geen mogelijkheid bestaat dat daarmee de integriteit van OS/390 aangetast kan worden. De kracht van de privileged instructies laat zich wel raden: hiermee kan de integriteit van OS/390 volledig worden tenietgedaan, waarmee ook de in beveiligingssoftware en informatiesystemen opgenomen controls hun waarde geheel verliezen.
Degene die hier voordeel mee wil behalen, stelt zich vervolgens de vraag: 'Hoe krijg ik dat Problem State bit op 0, dan ligt immers het systeem voor me open'. Het meest eenvoudige antwoord luidt: met de MODESET macroinstructie. Het ligt voor de hand dat deze instructie niet zomaar door elke taak kan worden uitgevoerd: hiervoor moet de taak geautoriseerd zijn, bijvoorbeeld via de Authorized Program Facility (APF). Na de behandeling van de geheugenbescherming wordt verder ingegaan op het begrip APF. Voor de volledigheid wordt hier vermeld dat er voor het OS/390-systeem nog twee andere redenen zijn om de taak als geautoriseerd te beschouwen: de taak draait in Supervisor State of met een key in het PSW kleiner dan 8. Ook in die situaties mag dus de MODESET macro-instructie worden gegeven.

Veel systeemtaken (onderdelen van OS/390 of subsystemen) draaien standaard in Supervisor State. Hetzelfde geldt voor veel user exits en voor alle user SVC's. User exits en user SVC's bieden de mogelijkheid om kleine wijzigingen of uitbreidingen te maken op het operating system, subsystemen of beheertools. Deze stukken pro-

grammatuur zijn alle potentieel in staat de integriteit van OS/390 te ondermijnen. Het volgende kan daarom worden gesteld:

De adequate bescherming van source- en loadlibraries van systeemsoftware, subsystemen, user exits en user SVC's tegen ongeautoriseerde wijzigingen is een essentiële control ter waarborging van de integriteit van OS/390.

Als dit goed is geregeld, dan resteren er blijkens het bovenstaande nog slechts twee andere belangrijke aandachtspunten: de key in het PSW en APF-autorisatie.

Geheugen

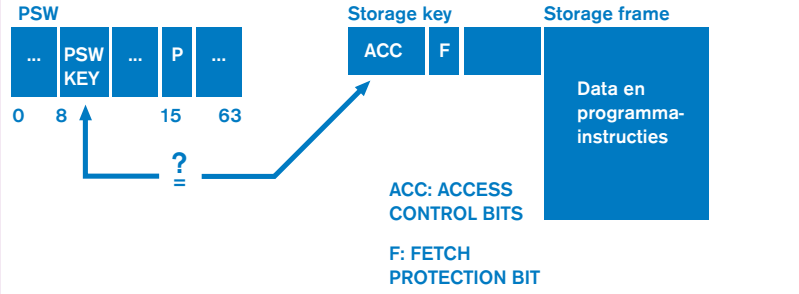
OS/390 is een multi-user- en multi-taskingsysteem: het wordt altijd gebruikt door meerdere (functioneel gescheiden) programma's tegelijkertijd. De programma's en de gegevens staan allemaal in hetzelfde geheugen van de computer, en het is dus zaak dat er een absolute scheiding wordt gerealiseerd tussen de verschillende geheugengebieden.
De geheugenbescherming is al aan de basis goed aangepakt: elk 4Kb fysiek geheugen (central storage frame) is in de hardwarearchitectuur afzonderlijk beveiligd met een storage key, ter grootte van acht bits (zie figuur 2). Hiervan gebruikt het systeem er vier voor de zogenaamde access control bits om de waarde 0 tot 15 te kunnen bevatten; een ander bit, het fetch-protection bit, wordt gebruikt als een vlaggetje. OS/390 gebruikt standaard de waarde 8 in de access control bits voor alle 'normale' gebruikers, beheertools en informatiesystemen. Waarden kleiner dan 8 worden gebruikt voor systeemtaken en subsystemen.
Het fetch-protection bit speelt alleen een rol bij het lezen van storage frames: als het fetch-protection bit 0 is, dan mag iedere taak de gegevens in het frame lezen. Staat het bitje op 1, dan hangt het af van de waarde van de access control bits van het frame en van de key die in het PSW van de taak staat. Alleen als de waarden gelijk zijn mag de taak het frame lezen.
De regels voor het schrijven naar het storage frame zijn eenvoudig: dit mag alleen als de waarden overeenkomen. Er is één uitzondering op de lees- en schrijfregels: een taak die in het PSW een key heeft gelijk aan 0 (ook wel aangeduid als 'key zero'), mag in elk storage frame lezen en schrijven.
Nu zijn er twee relevante vragen: 1. 'Hoe kan ik de storage key van een storage frame aanpassen aan mijn wensen?' en 2. 'Hoe kan ik de key in het PSW aanpassen, het liefst naar 0?'. Het antwoord op de eerste vraag is: met een privileged instructie (bijvoorbeeld met SET STORAGE KEY EXTENDED). Het antwoord op de tweede vraag luidt: bijvoorbeeld met de (privileged) MODESET macro-instructie.
De toegang tot het geheugen wordt dus indirect bepaald door de controls rond de instructieset.

Als een taak in Problem State draait, een normale PSW-key van 8 heeft en niet APF-geautoriseerd is, dan zijn er voor de taak geen mogelijkheden om buiten het toegewezen stuk geheugen in het systeem te lezen of te schrijven.

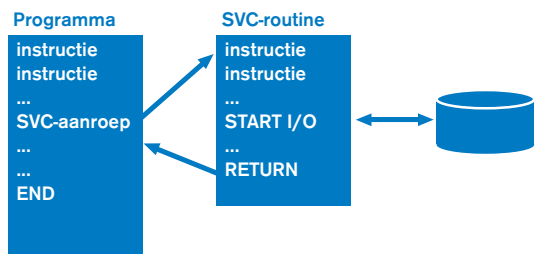
OS/390 gebruikt de solide bescherming van de fysieke storage frames bij het beheer en de beveiliging van de geheugenruimten (address spaces) voor de verschillende gelijktijdige gebruikers van het systeem. Elke gebruiker (persoon, subsysteem, beheertool of informatiesysteem) heeft een virtuele geheugenruimte ter grootte van maximaal 2Gb. Een deel van deze virtuele geheugenruimte wordt door alle gebruikers gezamenlijk gebruikt: de common area. Juist bij het beheer van deze common area is het voor het systeem van groot belang om onderscheid te kunnen maken tussen 'gewone' taken (met key 8 in het PSW) en (sub)systeemtaken (key kleiner dan 8). De S/390-hardware bevat ook voorzieningen voor het beheer van de gescheiden adresruimten. Gebruikmakend van deze en de eerder beschreven voorzieningen in de hardware zorgt OS/390 ervoor dat 'gewone' taken niet in elkanders geheugenruimte kunnen lezen of schrijven. De gescheiden adresruimten zijn overigens niet alleen van het allergrootste belang voor de betrouwbaarheid van het OS/390-systeem. Het is ook een heel belangrijke eigenschap voor de beschikbaarheid van OS/390: als een taak abend, dan heeft dat slechts gevolgen voor de address space waarin de taak draait. De abend heeft nooit kwalijke gevolgen voor andere taken of gebruikers: zij zijn zowel uit oogpunt van beveiliging als van beschikbaarheid volledig van elkaar geïsoleerd.

APF

De Authorized Program Facility is een mechanisme in OS/390 (dus in de software) dat het mogelijk maakt om programma's in staat te stellen zichzelf in Supervisor State te brengen. Een programma is APF-geautoriseerd als het afkomstig is uit een APF-geautoriseerde loadlibrary én bij het aanmaken van de loadmodule de indicatie Authorization Code (AC) = 1 heeft meegekregen. Het is een APF-geautoriseerd programma toegestaan om de MODESET macro-instructie uit te voeren, en daarmee zichzelf in Supervisor State te brengen. Met MODESET kan het ook de key in het PSW aanpassen, bijvoorbeeld naar 0, om daarmee overal in het geheugen te kunnen schrijven en lezen.
SYS1.PARMLIB is van oudsher de library waarin de parameters voor MVS en OS/390 zijn vastgelegd. Aan de hand van deze parameters wordt het systeem opgebouwd tijdens het initialisatieproces, de IPL (Initial Program Load). In de SYS1.PARMLIB members IEAAPFnn – en sinds enkele jaren in PROGnn – is een lijst opgenomen van de loadlibraries die APF-geautoriseerd moeten zijn. Een library is alleen APF-geautoriseerd als deze in één van de geldende members IEAAPFnn of PROGnn is opgenomen, maar kan in principe ook met een opera-



Figuur 2. De key in het PSW wordt vergeleken met de access control bits van het storage frame.



Figuur 3. Een programma dat een I/O wil doen, roept hiervoor een SVC aan; START I/O is namelijk een privileged instructie.

torcommando dynamisch worden toegevoegd aan de lijst van loadlibraries (indien toegestaan door het securitypakket).

Alleen expliciet benoemde loadlibraries kunnen APF-geautoriseerde loadmodules bevatten.

SVC's

Voor een veelheid van acties zou het voor een taak noodzakelijk zijn om een hoge autorisatie te hebben. Bijvoorbeeld om een I/O te doen, een bericht naar de operatorconsole te schrijven, het alleenrecht op het schrijven naar een dataset aan te vragen (integriteitsmaatregel), een timer te zetten of om een subprogramma aan te roepen.
Om gewone, niet-geautoriseerde taken in staat te stellen zulke systeemtaken uit te voeren, is in MVS en OS/390 de mogelijkheid gecreëerd om deze taken op een beheerste manier te delegeren aan het operating system. Hiertoe doet de (niet-geautoriseerde) taak een Supervisor Call (SVC), waarna de geautoriseerde SVC-routine de taak uitvoert (zie figuur 3).

Samenvatting

De S/390-architectuur zorgt er in samenwerking met OS/390 voor dat:
* programma's (met uitzondering van systeemsoftware en APF-geautoriseerde programma's) slechts instructies mogen uitvoeren die de integriteit van OS/390 niet kunnen ondermijnen en slechts in het eigen geheugengebied kunnen lezen en schrijven;
* lezen en schrijven in common storage alleen wordt toegestaan voorzover het de integriteit van OS/390 niet bedreigt;
* noodzakelijke taken die een hogere autorisatie behoeven kunnen worden aangeroepen via een goed gedefinieerde en gecontroleerde interface: SVC's.

S/390 en OS/390 zorgen samen voor de solide basis voor een adequaat beheer van informatiesystemen. Maar, aan een basis alleen heb je niet veel. Het is juist de aanvulling, of soms ook samensmelting, met volwassen beheertools en beheerprocedures die het geheel tot een betrouwbaar platform maken met een grote beschikbaarheid.

Beheertools en beheerprocedures

In het begin van de S/360-jaartelling waren er in het geheel geen beheertools, en maar nauwelijks beheerprocedures. De configuratie was simpel: een centraal mainframe met wat direct gekoppelde apparatuur. Het onderscheid tussen test, acceptatie en productie bestond niet, omdat er in veel gevallen maar één systeem was. Er draaide nog maar een beperkt aantal applicaties, voornamelijk in de batch.
In de zeventiger jaren begon de behoefte te ontstaan aan beheertools. Eén van de eerste gebieden waar men de behoefte voelde was bij het beheren van de (toen al) enorme hoeveelheden tapes. Vrij snel daarna kwamen beveiliging en scheduling binnen het gezichtsveld, gevolgd door tools om het problem- en change-managementproces te ondersteunen (medio tachtiger jaren).
Tegenwoordig kan een grote omgeving onmogelijk nog zonder deze beheertools. Een gemiddeld rekencentrum heeft in de productieomgeving:
* vele tienduizenden programma's voor duizenden applicaties;
* duizenden loadlibraries, waarvan circa een honderdtal APF-geautoriseerd;
* vele tienduizenden datasets;
* vele duizenden of enkele tienduizenden gebruikers (en terminalaansluitingen);
* vele honderden DASD-volumes;
* vele tienduizenden tapes;
* tientallen berichten per seconde die op de operatorconsole verschijnen;
* vele duizenden jobs (batchprogramma's) per dag;
* honderdduizenden pagina's printoutput per dag.

De grote aantallen waren destijds een belangrijke reden om beheertools te ontwikkelen en te implementeren. Grote aantallen zijn tegenwoordig echter geenszins nog de enige aanleiding voor het gebruiken van beheertools. Ook in kleinere omgevingen en bij kleinere aantallen is gebleken dat het zinvol is in een zo vroeg mogelijk stadium te beginnen met de implementatie van beheertools, zeker bij een voorziene groei ... en welk bedrijf plant tegenwoordig geen groei?

Aanvankelijk ontwikkelden systeemprogrammeurs vaak zelf de (eenvoudige) beheertools, hetzij omdat er nog geen commercieel beschikbare tools bestonden, hetzij omdat het goedkoper leek. In de afgelopen twintig jaar is men in de mainframewereld overtuigd geraakt van de wenselijkheid om zelfgeschreven tools uit te bannen, ten gunste van de onderhoudbaarheid (onderhoudskosten), de betrouwbaarheid en de continuïteit. Het zo veel mogelijk werken met standaardsystemen en tools bevordert in hoge mate de uitwisselbaarheid en overdraagbaarheid van (informatie)systemen, hetgeen bij overnames of outsourcing een belangrijk punt kan zijn. Paans en Gianotten ([Paan93]) gaan in *Management van computercentra* in op de problematiek van dit soort lokale modificaties die ook in de vorm van user exits en user SVC's kunnen bestaan.

In Unix-omgevingen was in de tachtiger, begin negentiger jaren een herhaling van de geschiedenis waarneembaar: terwijl voor die omgevingen inmiddels reeds redelijke beheertools beschikbaar waren gekomen, programmeerden de beheerders er nog steeds op los zodat door een samenspel van scripts en programma's beheertaken werden geautomatiseerd. De commerciële tools kwamen er bij hen niet in, omdat er altijd wel een functie te benoemen was waar de tools niet aan volde-

den. Het bevredigen van alle technische behoeften leek in die gevallen te prevaleren boven het komen tot een in alle opzichten beheersbare omgeving.
Vrijwel overal hebben we in de afgelopen tien jaar een ongebreidelde groei kunnen zien in het aantal Unix- en NT-systemen bij bedrijven, wat bijna zonder uitzondering leidde tot enorme beheerproblemen. Voor een deel kan deze beheerproblematiek succesvol worden opgelost met beheertools en -procedures. Een serieus alternatief is evenwel het overbrengen van de Unix-applicaties en webservers naar het OS/390-platform, om daar direct te profiteren van de bewezen betrouwbaarheid, beschikbaarheid en schaalbaarheid.

Zoals gezegd zijn de beheertools en -procedures in de loop der jaren ontstaan uit de groeiende behoefte. Onderstaand zijn de beheerprocessen vermeld onder hun 'historische' aanduiding (die je nog vaak tegenkomt), met tussen haakjes de beheerdiscipline waar het beheerproces onder valt.

Beheerprocessen die primair een bijdrage leveren aan de betrouwbaarheid:
* tapemanagement (Operations Management);
* scheduling (Workload Management);
* security management (Security Management);
* change management (Change Management);
* output balancing (Operations Management).

Beheertoolprocessen die primair een bijdrage leveren aan de beschikbaarheid:
* performance en capacity management (Performance en Capacity Management);
* DASD-management (Performance en Capacity Management);
* consolemanagement (Operations Management).

Parallel Sysplex sluit het rijtje af, en is feitelijk noch een beheerproces, noch een beheertool, maar een concept, ... een werkend concept!

Natuurlijk is het zo dat volgens de huidige inzichten een beheerorganisatie niet zonder problem management, configuration management en enkele andere managementdisciplines kan. In mijn optiek zijn het echter de bovenstaand opgesomde beheerprocessen die vanuit het verleden het meest hebben bijgedragen aan de betrouwbaarheid en beschikbaarheid van OS/390-omgevingen. Onderstaand worden ze stuk voor stuk behandeld.

Tapemanagement

Tapes werden en worden om verschillende redenen gebruikt:
* back-up van DASD-volumes om de gebruikelijke redenen, zoals full-restore bij een head-crash van een DASD-volume, of selectieve restore als een gebruiker een dataset per ongeluk heeft verwijderd of bij een abend van een productiejob;
* back-up van DASD-volumes om op een externe locatie veilig te stellen in verband met een eventuele uitwijk;
* uitwisseling van gegevens met andere partijen, vooral in de tijd dat datacommunicatie nog geen gemeengoed was.

Op een tape kunnen duizenden datasets worden opgeslagen, gewone datasets, sourcelibraries, loadlibraries, etc. Aan elk van de datasets kan een expiratiedatum worden meegegeven.
Dertig jaar geleden, toen er nog geen tapemanagementpakketten bestonden, was het grootste probleem het volgende: als het systeem (op verzoek van een gebruiker) om een vrije tape vroeg (een 'scratch tape'), dan was aan de buitenkant van de tape niet te zien of de tape nog geldige gegevens bevatte, gegevens waarvan de expiratiedatum nog niet was bereikt. Er waren meer problemen en vragen, zoals: welke tapes zijn in gebruik?; welke tapes kunnen worden hergebruikt?; welke tapes moeten worden verplaatst naar een externe kluis?; waar bevindt zich welke tape?; wanneer moet een tape worden afgedankt? Tapemanagementpakketten zoals CA-1, DFSMSrmm en TLMS geven de beheerders antwoorden op de vragen, en voorzien in de volgende functies:
* Volledige administratie van alle datasets op tapes. Per dataset of per groep van datasets worden regels vastgelegd hoe lang de dataset op tape moet blijven bestaan. Ook kan een aantal generaties worden opgegeven of een combinatie van expiratiedatum en aantal generaties. Op deze manier kan het vrijgeven van tapedatasets volledig worden geautomatiseerd.
* Volledige administratie van alle tapevolumes. Het pakket houdt per tape bij of zich nog geldige datasets op het tapevolume bevinden. Pas wanneer dat niet meer het geval is wordt de tape vrijgegeven en belandt in de scratch-pool. Tot dat moment is een tape zeer goed beschermd tegen abusievelijk overschrijven.
* Zonder tapemanagementpakket was er sprake van een lacune in de beveiliging tegen ongeautoriseerd lezen, omdat op tape slechts de laatste 17 tekens van de datasetnaam werden vastgelegd; tussenkomst van het tapemanagementpakket zorgde ervoor dat alle maximaal 44 tekens van de datasetnaam werden betrokken in de toegangscontrole.
* Administratieve ondersteuning van het proces dat 'vaulting' wordt genoemd. Dit houdt in het rouleren van tapes tussen verschillende (externe) kluizen op basis van vaultingrules die door de tapebeheerder in het pakket zijn vastgelegd. Correcte vaulting waarborgt dat de juiste tapes zich bevinden op de juiste locaties ten behoeve van (disaster) recovery.
* Administratieve ondersteuning bij het bewaken van de kwaliteit en de ouderdom van de tapes. Het pakket legt per tape vast wanneer de tape is aangeschaft, wanneer de tape voor het eerst is gebruikt, hoe vaak de tape is gebruikt, hoeveel leesfouten er zijn geweest en wanneer de tape voor het laatst is gebruikt.
* Uitvoerige rapportagemogelijkheden over tapedatasets, tapevolumes, scratch-pool en inhoud van kluizen (vaults).

Uit het bovenstaande zal duidelijk zijn geworden dat een tapemanagementpakket de procedures rond tapes op alle fronten ondersteunt, en antwoord geeft op de vele vragen rond het beheer van tapes.
In een groot rekencentrum is het niet ongebruikelijk dat met het tapebeheer 1 FTE gemoeid is; dit is nog exclusief het door de operators 'mounten' (ophangen of in de cartridge-unit stoppen) van tapes. Taperobots brengen hier voor beiden een besparing van werk: voor de tapebeheerder bij de fysieke handelingen rond de scratch-pool,

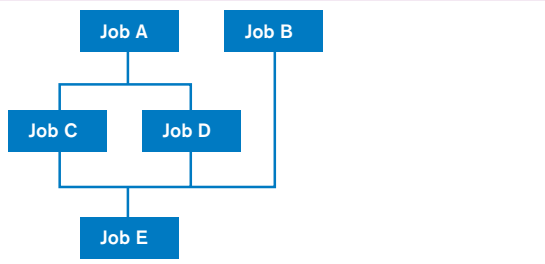
en voor de operator bij de honderden tot duizenden mounts per dag.

Scheduling

De taak van een schedulingpakket is vrij simpel te omschrijven: ervoor zorgen dat de juiste jobs met de juiste parameters en de juiste invoer-/uitvoerbestanden in de juiste volgorde op het juiste tijdstip op het juiste systeem worden uitgevoerd. Elk van deze voorwaarden is onmiskenbaar van belang voor de betrouwbaarheid van de gegevens in de informatiesystemen, met name voor de integriteit: als jobs bijvoorbeeld in de verkeerde volgorde worden uitgevoerd, baseren batchprogramma's zich op onjuiste gegevens en is ook het resultaat niet integer. Hetzelfde geldt als met verkeerde parameters (datum/weeknummer) wordt gedraaid of als de job op een verkeerd systeem wordt uitgevoerd (bijvoorbeeld op een acceptatiesysteem, waarna de output automatisch naar het productiesysteem wordt gestuurd om daar te worden geprint).

Schedulingpakketten als OPC/ESA, CA-7/11, Control-M en Scheduler dragen met de volgende functies bij aan de betrouwbaarheid maar ook aan de beschikbaarheid van het systeem:

* Per job of per groep van jobs kunnen de predecessors (voorgangers) en de successors (opvolgers) worden vastgelegd in het pakket. Op deze manier zijn eenduidig de volgordes en afhankelijkheden bepaald (zie figuur 4). Ook kan per job worden bepaald welke actie moet worden uitgevoerd indien de job abend.
* Runschema's kunnen door het systeem automatisch worden aangepast afhankelijk van feestdagen in de week. Bij het definiëren van een job in het schedulingpakket wordt aangegeven wat er met de job moet gebeuren als de rundag op een feestdag valt, of bijvoorbeeld wanneer de dag ervoor of erna een feestdag is. De gehele verwerking wordt aldus van tevoren bepaald en eenduidig vastgelegd zodat de runschema's volledig reproduceerbaar zijn en niet afhankelijk van de ene of juist die andere operator, systeembeheerder, werkvoorbereider of applicatiebeheerder.
* Rekening houdend met de historische looptijden van de job en het historische resourcegebruik (hoeveelheid tijdelijke schijfruimte, aantal benodigde tape-units) kan een schedulingpakket zinvolle workload balancing verzorgen: als de resources er (tijdelijk even) niet zijn, dan heeft het geen zin een job te starten die van deze resources afhankelijk is.
* Jobs die dreigen te laat te eindigen kunnen automatisch tot een van tevoren gedefinieerde actie leiden, bijvoorbeeld een fel gekleurd bericht op de operatorconsole, het tijdelijk tegenhouden van additioneel werk of het oppiepen van de shiftleader.
* Het schedulingpakket verzamelt in de loop der tijd veel gegevens over de resources die door de verschillende jobs worden gebruikt. Hiervan maakt een andere functie van het pakket dankbaar gebruik: simulatie en modelling. Hiermee kan bijvoorbeeld een antwoord worden verkregen op de vraag: 'Wat gebeurt er met mijn batch window als ik er twee tape-units bijplaats?', of 'Welk beeld van de batchverwerking kan ik verwachten als job Q 25 procent meer CPU gaat verbruiken?'. De simulatie kan hierbij bijvoorbeeld een complete nachtverwerking naspelen en weergeven.
* Deze pakketten kennen uitgebreide rapportagemogelijkheden tot en met flowcharts van groepen afhankelijke jobs.
* Het vastleggen van de definities vindt over het algemeen plaats via een on-line-interface (normaliter via TSO/ISPF). In het pakket (of in het aanwezige securitypakket) kan worden bepaald welke medewerker onderhoud mag plegen aan de definities van welke (groepen) jobs. Op soortgelijke wijze kan worden bepaald wie welke scheduling-gerelateerde commando's mag geven voor welke jobs. Op deze manier kan een organisatie het centrale beheer van het schedulingpakket in bepaalde mate decentraliseren.



Figuur 4. Afhankelijkheden tussen jobs: dit schedule start met jobs A en B; job E mag pas starten na succesvolle beëindiging van jobs B, C en D.

Schedulingpakketten zijn niet meer weg te denken uit de huidige grote verwerkingsomgevingen: de duizenden jobs per etmaal, met hun afhankelijkheden van elkaar en van on-linesystemen, en het steeds beperkter wordende batch window maken een automatische uitvoering en bewaking van de batchload tot een absolute voorwaarde voor een adequate (en dus betrouwbare) verwerking van de informatiesystemen.
Voor het aanbrengen van wijzigingen in de schedulingdefinities zijn in een vroeg stadium procedures ontworpen, waarbij het daadwerkelijke implementeren van de wijzigingen gelegd was bij de werkvoorbereider, systeembeheerder of applicatiebeheerder. In grotere rekencentra zijn hier meerdere FTE's mee belast.
Een combinatie van procedurele en technische controls dwingt af dat alleen daartoe geautoriseerde jobs in de productie-workload meedraaien, zodat de totale bezetting van het productiesysteem tot op zekere hoogte voorspelbaar is of ten minste beheersbaar. Nieuwe ontwikkelingen zoals webservers die geïmplementeerd worden op OS/390, maken de workload minder goed voorspelbaar, hetgeen zowel door nieuwe hardware- als softwareconcepten voor een deel wordt opgelost. Te zijner tijd zal in een apart artikel nader op deze nieuwe ontwikkelingen worden ingegaan.

Security management

De implementatie van securitypakketten kwam eind zeventiger jaren goed op gang. In eerste instantie werden deze pakketten alleen gebruikt voor logon security, dus user-id- en passwordverificatie tijdens het aanloggen aan het systeem. Beveiliging van datasets tegen ongeautoriseerd gebruik volgde spoedig daarna, of werd bij sommige bedrijven vanaf het begin meegenomen bij de implementatie van het pakket. De pakketten zijn in de afgelopen twintig jaar sterk in functionaliteit verbeterd,

en beveiligen vandaag de dag praktisch elk resource dat maar enigszins beveiliging behoeft.
Andere beheertools hadden veelal zelf ook beveiligingsmogelijkheden ingebouwd (zogenaamde internal security). Vaak was dit vanwege een gebrek aan security-managementpakketten, niet alleen omdat ze nog niet bestonden of de benodigde functionaliteit nog niet boden, maar ook omdat lang niet elk rekencentrum een security-managementpakket had geïmplementeerd. Internal security zien we bijvoorbeeld bij:
* tapemanagementpakketten – bijvoorbeeld wie mag in de administratie handmatig een tape op scratch zetten?
* schedulingpakketten – bijvoorbeeld wie mag de definities van welke job aanpassen?
* transactiemanagers – bijvoorbeeld wie mag welke transacties uitvoeren?
* databasemanagers – bijvoorbeeld wie mag in welke tabel muteren?

Meer en meer wordt ernaar gestreefd om deze interne beveiliging onder te brengen in het security-managementpakket. Het laatste wordt in dat geval aangeduid als de external security manager. Beweegredenen hiervoor zijn onder meer:
* De beveiligingsdefinities zijn dan vastgelegd in robuuste, speciaal voor dit doel ontworpen software.
* Alle beveiligingsdefinities zijn vastgelegd in één database, waardoor het gemakkelijker wordt om (met rapportages of queries) eventuele doorbreking van gewenste functiescheidingen te signaleren.
* De beveiliging van de securitydatabase, back-ups van de securitydatabase en extracten ervan is in het algemeen goed geregeld; de vraag is legitiem of dit bij de her en der verspreide internal security faciliteiten van de verschillende tools ook het geval is.

Securitypakketten zoals RACF, ACF2 en Top Secret bieden globaal de volgende functies:
* Logon security, zowel tot het netwerk, als tot applicaties, transactiemanagers, maar ook de toegang tot Unix op OS/390 ('onder OS/390' mag ook: 't is maar net hoe je het tekent). Authenticatie vindt nog steeds grotendeels plaats op basis van een password, dat afhankelijk van de implementatie periodiek moet worden gewijzigd, speciale tekens moet bevatten, een minimale lengte moet hebben en binnen een bepaalde periode niet mag worden hergebruikt.
* Beveiliging van de toegang tot datasets, DASD-volumes, tapevolumes, CICS-transacties, IMS-transacties, DB2-resources, schermen en functies in beheertools, output in spool en honderden andere resourcetypen in de OS/390-omgeving.
* Program Access to Data Sets (PADS in RACF) of Program Pathing (in ACF2 en Top Secret): een bijzonder waardevolle control, die het mogelijk maakt persoonlijke user-id's of batch user-id's toegang te geven tot datasets *alleen* wanneer de toegang plaatsvindt via een met name genoemd programma uit een met name genoemde loadlibrary. Op deze manier wordt voorkomen dat applicatiecontrols kunnen worden omzeild.
* Gecontroleerd decentraal beheer van de beveiligingsdefinities, zodat het definiëren van nieuwe gebruikers en het autoriseren voor het gebruik van resources kan worden gelegd op die plek in de organisatie waar deze taken het meest effectief en efficiënt kunnen worden uitgevoerd.

In RACF-omgevingen wordt veelal Consul/RACF gebruikt als waardevolle aanvulling op de rapportagefaciliteiten van RACF en als een krachtig beheertool rond RACF.

Change management

Zorgvuldig change management is alom erkend als één van de belangrijkste controls bij de beheersing van de betrouwbaarheid van informatiesystemen en data. Change management in de softwareomgeving richt zich op het geheel van systeemsoftware, systeeminstellingen (zoals de eerdergenoemde SYS1.PARMLIB), applicatiesoftware, en de daarbijbehorende schermen (panels), job control language en databasegerelateerde definities. De uitdaging is daarbij het managen van de wijzigingen in de genoemde componenten in de test/ontwikkel-, acceptatie- en productieomgeving.

> Change management is één van de belangrijkste controls bij de beheersing van de betrouwbaarheid van informatiesystemen en data.

Voor systeemsoftware is hiervoor steeds een specifiek tool beschikbaar geweest, dat standaard met MVS en OS/390 wordt meegeleverd: het System Modification Program (SMP/E). Dit tool administreert van alle OS/390-componenten de versies en de afhankelijkheden van andere componenten en controleert alle wijzigingen die worden aangebracht op de systeemsoftware. Aanvullend dient het gebruik van het pakket omgeven te worden door procedurele voorzieningen, bijvoorbeeld bij de overdracht van de wijzigingen van het installatiesysteem naar de test-, acceptatie- en productiesystemen.
Voor het change management van applicatiesoftware en bijbehorende attributen bestaan twee categorieën tools:
* registratie en goedkeuring van wijzigingen (bijvoorbeeld Infoman/MVS en Netman);
* 'besturing' van source- en loadlibraries op basis van het change-managementproces (bijvoorbeeld Endevor, Librarian/CCF en Pan/APT).

'Besturing' houdt in deze context in dat alleen het pakket updates mag initiëren op de source- en loadlibraries van de applicatie in de verschillende omgevingen; de inhoud van die libraries staat volledig onder controle van het pakket.
Deze pakketten vormen een onmisbare ondersteuning voor het change-managementproces. Detectieve controls zijn echter eveneens onmisbaar omdat er veelal wel mogelijkheden zijn om wijzigingen aan te brengen buiten de controle van het change-managementproces om. Pakketten als Consul/Audit en CA-Examine zijn in staat als het ware een foto te maken van relevante datasets in de productieomgeving en op een later tijdstip de wijzigingen sinds het tijdstip van de 'foto' te signaleren. Het uitgangspunt is dat elke wijziging moet kunnen worden teruggevonden in de change-managementregistratie. Wijzigingen die daar niet aan voldoen kunnen dan nader worden onderzocht. Bij grote rekencentra wordt de

geschetste controle uitgevoerd door de change manager, interne controle of door compliance auditors.

Output balancing

Beta 91 en APCDDS zijn twee voorbeelden van een categorie vrij onbekende beheertools: output balancing of report balancing. Dit type pakketten stelde een grote Nederlandse bank begin jaren negentig in staat de afdeling waar dagelijks alle relevante output werd gecontroleerd, te verkleinen van veertig naar vijf medewerkers. Vooral de automatische integriteitscontroles die deze software kan uitvoeren maakt de toepassing ervan in potentie heel interessant.
Functies van dit soort pakketten zijn:
* controle of de inputbestanden van een job wel exact gelijk zijn aan de outputbestanden van de job die de bestanden heeft geproduceerd, dezelfde dag, of bijvoorbeeld een jaar geleden;
* controle of velden in input/outputrecords binnen vooraf bepaalde marges liggen;
* verbandscontroles tussen velden in printoutput;
* verbandscontroles tussen velden van de huidige run met velden van een voorgaande run.

Als het pakket afwijkingen vindt, dan kan een reeks van automatische vervolgacties worden geïnitieerd, variërend van het niet printen van de joboutput tot het volledig opnieuw uitvoeren van de job(s).
De overwegingen om een dergelijk pakket in te zetten voor controles die de applicatie ook zelf kan uitvoeren, zijn voor een deel dezelfde als de overwegingen om van internal security over te gaan naar external security: deze pakketten zijn gespecialiseerd in hun taak, er hoeft geen eigen onderhoud te worden uitgevoerd aan controlemechanismen in de applicatie, en als het pakket is geïmplementeerd dan kunnen alle applicaties er gebruik van maken. De laatste twee punten pleiten natuurlijk heel sterk voor een output-balancingpakket: anders moeten alle (unieke) controles in elke applicatie opnieuw worden geprogrammeerd of de controle moet aan de eindgebruikers worden overgelaten. Bij beide activiteiten kunnen fouten worden gemaakt.

Performance en capacity management

Deze managementdisciplines zijn bij uitstek de beheergebieden die een belangrijke rol spelen bij de beschikbaarheid van de systemen. Hiertoe behoort het monitoren van:
* wachttijden tijdens I/O's (met name naar DASD);
* overige wachttijden voor en tijdens de verwerking;
* I/O-responstijden;
* hit-ratio van cachegeheugen;
* trends in aantallen jobs, transacties, I/O's, etc.;
* trends in verwerkingstijden van jobs en transacties.

Hierbij wordt een onderscheid gemaakt tussen real-time monitoring en de dagelijkse of wekelijkse rapportages over de kengetallen. Real-time monitoring is een onmisbare activiteit bij de goede bewaking van de beschikbaarheid van de systemen, en wordt in toenemende mate uitgevoerd door intelligente software (zoals de OS/390-component WLM – Workload Manager) die zelfstandig acties onderneemt om de performance binnen de gestelde grenzen te houden.
Pakketten die invulling geven aan deze activiteiten zijn onder andere RMF, MICS, PM/MVS, The Monitor, Omegamon, Mainview, Patrol en Consul/SMS DSCAT. Tegenwoordig wordt ook DASD-management tot deze beheergebieden gerekend: het monitoren van het gedrag en de (momentane) bezetting van DASD-volumes en het zo nodig ingrijpen door bestanden te verplaatsen.
In de grotere rekencentra, waar het beheergebied performance en capacity management serieus is ingevuld, wordt hier zeker 1 FTE aan besteed. Het beheergebied heeft samen met scheduling een geheel nieuwe dimensie gekregen met het beschikbaar komen van Parallel Sysplex en de ondersteunende software, waarover zo dadelijk meer.

DASD-management

Wat doet een PC-gebruiker die iets te weinig vrije schijfruimte heeft? Er worden op de harde schijf wat bestanden gezipt, wat doorgaans een besparing van zeker zeventig procent oplevert. Als de ruimte dan weer te beperkt wordt, dan worden enkele van de zipfiles naar een floppy gekopieerd. Wanneer de gebruiker later bijvoorbeeld via een snelkoppeling (shortcut) refereert aan het gezipte en op floppy geplaatste bestand, dan verschijnt een grappig zoekend zaklantaarntje in beeld, en moet de gebruiker er zelf voor zorgen dat het bestand weer van de juiste floppy terug op schijf wordt geplaatst. In OS/390-omgevingen is de problematiek niet anders, wel de oplossingen ... gelukkig.
De voornaamste spelers op dit front zijn de beheertools DFSMSdss en DFSMShsm die samen de volgende functionaliteit bieden:
* back-up van datasets en complete DASD-volumes;
* op basis van vooraf bepaalde regels automatisch comprimeren van een dataset naar het zogenaamde migration level 1 (ML1) op DASD;
* na verloop van tijd, ook weer op basis van ingestelde regels, datasets automatisch verplaatsen van ML1 naar ML2 op tape;
* automatisch terugplaatsen op DASD van een bestand dat zich bevindt op ML1 of ML2 en dat weer nodig is; voor een dataset op ML1 neemt het terugplaatsen één tot enkele seconden in beslag; voor een dataset op ML2 duurt het bij gebruik van een taperobot circa vijftien seconden.

DASD-management speelt een belangrijke rol bij de beschikbaarheid van het systeem, zeker ook het proactieve DASD-management. Een standaard-rapportagetool dat hiervoor wordt gebruikt, is het bij OS/390 meegeleverde ISMF. Het beheergebied DASD-management wordt vandaag de dag gerekend tot performance en capacity management.

Consolemanagement

De operatorconsole in grote mainframeomgevingen krijgt per minuut zo'n honderd tot enkele honderden berichten te verwerken. Vele daarvan zijn slechts informatief en worden doorgaans dan ook onderdrukt. Evengoed verschijnen er op de console zo enorm veel berichten dat ze aan de aandacht van de operator kunnen ontsnappen. Overigens blijven de als zodanig gekenmerkte echt belangrijke berichten op de console staan, zodat ze wel moeten opvallen.

Dan blijft echter het punt: hoe snel, maar vooral *hoe* reageert de operator op het bericht? Zoals ook bij scheduling al werd belicht, is het wenselijk om de reacties reproduceerbaar te maken, zodat de operator, en daarmee het (informatie)systeem, in gegeven omstandigheden steeds hetzelfde reageert.
Consolemanagementpakketten zoals OPS/MVS, Automate/MVS, AutoOperator en Opera zijn erop gericht het reageren op consoleberichten te automatiseren. Op basis van:
* (combinaties van) consoleberichten;
* het tijdstip waarop het bericht op de console komt;
* de frequentie waarmee eenzelfde bericht steeds naar de console wordt geschreven;
* het al dan niet actief zijn van bepaalde jobs of on-line-systemen;
* de stand van vlaggetjes,

kan het pakket automatisch een actie of een reeks van acties initiëren, zoals
* het geven van een operatorcommando om bijvoorbeeld taken te stoppen of te starten;
* het geven van een netwerkcommando om netwerkverbindingen te stoppen of te starten;
* het aanpassen van de verwerkingskarakteristieken om bijvoorbeeld een bepaald type workload tijdelijk voorrang te geven;
* het doorgeven van een signaal (event/vlaggetje) aan het schedulingpakket;
* het versturen van een e-mail;
* het oppiepen van iemand.

In feite kan het consolemanagementpakket worden beschouwd als het verbindende element tussen operating system, subsystemen, beheertools en informatiesystemen en als het echt niet anders kan: de operator. Relatief simpele deelimplementaties van consolemanagementpakketten kunnen reeds een belangrijke bijdrage leveren aan de beschikbaarheid van het systeem.

Parallel Sysplex

Parallel Sysplex is een betrekkelijk jong begrip op het gebied van samenwerking tussen systemen. Het is geen tool en het is geen beheerdiscipline: het is de verzameling van hardwarefaciliteiten, software en microcode die samen invulling geven aan het begrip Parallel Sysplex.
Uitgangspunten bij het ontwerp van Parallel Sysplex waren:
* continue beschikbaarheid, door
  – minimaliseren van geplande onderbrekingen in de verwerking (bijvoorbeeld voor onderhoud, back-ups, reorganisaties);
  – minimaliseren van de gevolgen van niet-geplande onderbrekingen, zoals uitval van hardware of software die niet goed functioneert;
* vergroten van de schaalbaarheid: capaciteitsuitbreidingen moeten geleidelijk en zonder onderbrekingen kunnen worden geïmplementeerd;
* dynamic workload balancing: volledig automatische (her)verdeling van het werkaanbod over meerdere systemen ter ondersteuning van de bovenstaande uitgangspunten en om gewenste serviceniveaus (bijvoorbeeld responstijden en verwerkingstijden) te kunnen bereiken.

Kenmerkend voor een Parallel Sysplex is de Coupling Facility (CF) die specifieke voorzieningen biedt voor high performance data sharing. De systemen die deelnemen in het Parallel Sysplex zijn via optische verbindingen aangesloten op de CF, zodat communicatie met grote bandbreedte kan plaatsvinden.

Het is de verzameling van hardwarefaciliteiten, software en microcode die samen invulling geven aan het begrip Parallel Sysplex.

Het zijn de datasharingfaciliteiten in de CF die de basis vormen voor de samenwerking van beheertools, subsystemen, transactie- en databasemanagers over alle aangesloten systemen heen. Deze snelle en eenduidige manier van gegevensuitwisseling biedt geheel nieuwe mogelijkheden voor het betrouwbare gemeenschappelijke gebruik van (data)resources en om met elkaar te communiceren over workload, capaciteit, performance. Voorbeelden van toepassingen zijn:
* dynamische verdeling van CICS- en IMS-transacties over de systemen binnen hetzelfde Parallel Sysplex, waarbij de workload manager de actuele bezetting van de systemen betrekt in de beslissing welk systeem de transactie zal gaan uitvoeren;
* het behoud van de sessie tussen de gebruiker en een applicatie bij het uitvallen van bijvoorbeeld de applicatie, OS/390 of een hardwarecomponent;
* datasharing over alle aangesloten systemen voor DB2- en IMS-databases en VSAM-files.

De systemen hoeven overigens niet noodzakelijkerwijs in hetzelfde gebouw te staan, hetgeen nieuwe oplossingen biedt voor disaster recovery: bij uitval van de systemen op locatie A kunnen systemen op de locaties B en C de workload van locatie A automatisch overnemen.
In het Parallel Sysplex zijn hiervoor de volgende voorzieningen opgenomen:
* Sysplex Failure Management, hetgeen de mogelijkheid biedt om vooraf regels vast te leggen hoe het sysplex moet reageren bij het uitvallen van verbindingen of systemen, bijvoorbeeld het volledig automatisch verwijderen van een niet goed functionerend systeem uit het sysplex;
* Automatic Restart Manager, die ervoor zorgt dat batch jobs en started tasks automatisch – eventueel op een ander systeem – opnieuw worden gestart;
* de Workload Manager van OS/390, die er zorg voor draagt dat bij mindere resourcecapaciteit bedrijfskritische applicaties de juiste prioriteit krijgen.

Aldus vormt het Parallel Sysplex, samen met voldoende redundante hardware, *het* antwoord van IBM op de vraag naar 99,999% beschikbaarheid.

Ing. P. Mienes RE
is EDP audit manager in de unit Business Strategic IT binnen de business unit Technology & Assurance (T&A) van KPMG EDP Auditors. Onder zijn aandachtsgebieden vallen auditing en advisering in OS/390-omgevingen, zowel rond de beheerdisciplines als op diepgaand technisch niveau.

Conclusie

Eigenlijk is het niet zo vreemd: het OS/390-platform heeft inmiddels vijfendertig jaar de tijd gehad om zich te ontwikkelen, volwassen te worden en te stabiliseren. Het woord 'stabiliseren' moet echter niet verkeerd worden opgevat: zoals het hele automatiseringsspectrum is ook het OS/390-platform meer in beweging dan ooit tevoren. De aandacht voor ERP, Internet, e-business en groupwaretoepassingen leidde ook op dit platform tot nieuwe ontwikkelingen: OS/390 is zich de laatste tien jaar gaan openen voor de buitenwereld. Termen als Unix, TCP/IP, firewall, Baan, Oracle, SAP en webserver zijn binnengedrongen, niet alleen in de folders en handboeken, maar in tastbare implementaties op OS/390. Zo zal naar verwachting OS/390 als webserver een onmisbare plaats gaan innemen in e-commerce en e-business.

Wat de hardwarearchitectuur betreft was het platform vanaf het eerste begin al volwassen: een stringente scheiding tussen privileged en non-privileged instructies en tussen geheugenruimten van de verschillende gebruikers. Deze betrouwbare, robuuste basis is *de* kenmerkende eigenschap van het OS/390-platform en natuurlijk een essentiële voorwaarde voor de beheersbaarheid ervan.

Beheren om het beheersen ...
Beheersbaarheid is meer dan alleen een stabiel en betrouwbaar hardwareplatform. Het operating system draagt een belangrijke steen bij in de betrouwbaarheid en beschikbaarheid, maar het sluitstuk wordt gevormd door de beheertools. Zonder volwassen beheertools en beheerprocedures ontstaat al snel een niet goed beheersbare situatie.
De lange staat van dienst van het OS/390-platform brengt met zich mee dat de voorhanden zijnde beheertools grotendeels zijn uitgekristalliseerd. Mede door de opkomst van ITIL in de negentiger jaren zijn de beheertools ook effectief ingebed in de beheerprocedures. Het hardwareplatform, OS/390, de beheertools en beheerprocedures tezamen vormen de basis voor een bijzonder goed beheersbare omgeving voor bedrijfskritische applicaties.

Betrouwbaarheid was al langer een belangrijke voorwaarde voor dit soort applicaties; de *beschikbaarheidseisen* waren echter beperkt in die zin dat over het algemeen een beschikbaarheid van 5×10 uur werd verwacht. Steeds vaker zal echter door de gebruikers een beschikbaarheid van 7×24 uur worden geëist voor bijvoorbeeld webservers, firewalls en e-businesstoepassingen. Dit stelt bijzonder hoge eisen aan hardware, operating system, beheertools en beheerprocedures. Door dit artikel mag duidelijk zijn geworden waarom OS/390 hier zo'n uitstekende kandidaat voor is.

Literatuur

[Holl99]
Mw. J.A.M. Holla RE en drs. M.Th.J.M. Piels, *Beheer van IT vergt discipline*, in: Compact & ICT-auditing, 25 jaar Compact, jubileumuitgave.

[Mole99]
T. Molenaar, *Renpaard uit het Tertiair*, Computable, 10 december 1999.

[Paan93]
Prof. dr. ir. Ronald Paans en Marco H.E. Gianotten, *Management van computercentra; van chaos naar orde*, Giarte Publishing B.V., Amsterdam 1993.

[Piel99]
Drs. M.Th.J.M. Piels, *Grenzen aan het automatiseren van (het beheer van) de automatisering?*, Compact 1999/5.

Enterprise Systems Architecture/390: Principles of Operation (IBM SA22-7201-04).

OS/390: MVS Programming: Authorized Assembler Services Reference, Volume 3 (IBM GC28-1766-04).

MVS: Planning: Security (IBM GC28-1439-00).

OS/390: MVS Programming: Authorized Assembler Services Guide (IBM GC28-1763-05).

Continuous availability systems design guide (IBM SG24-2085-00).

Continuous availability S/390 technology guide (IBM SG24-2086-00).