



Faculteit Bedrijf en Organisatie

Een analyse van z/OS Health Checker en hoe deze te optimaliseren

Jonas Braem

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Thomas Pollet
Co-promotor:
Kevin Somers

Instelling: HCL Technologies

Academiejaar: 2019-2020

Tweede examenperiode

Faculteit Bedrijf en Organisatie

Een analyse van z/OS Health Checker en hoe deze te optimaliseren

Jonas Braem

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Thomas Pollet
Co-promotor:
Kevin Somers

Instelling: HCL Technologies

Academiejaar: 2019-2020

Tweede examenperiode

Woord vooraf

Deze proef is geschreven in het kader van het voltooien van de richting Toegepaste Informatica met keuzetraject Mainframe-beheer en Applicaties. Dit vond ik een interessant onderwerp omdat de mainframe me intrigeert en Health Checker was ook iets dat nog niet binnen de opleiding besproken was maar dat mij wel interesseerde. Ik vond deze proef ook zeer speciaal omdat mijn proef heeft bijgedragen aan een systeem dat nu geïmplementeerd is door de opdrachtgever. Ook het feit dat ik niks kende van het onderwerp was iets wat me aansprak om zo mijn kennis van de mainframe te verrijken. Deze proef is er wel niet zomaar gekomen en daarom wil ik nog enkele mensen bedanken voor hun bijdrage aan deze Bachelorproef.

De eerste persoon die ik wil bedanken is Kevin Somers voor de hulp en ondersteuning doorheen de gehele bachelorproef. Ik kon altijd terecht bij hem met vragen. En heb van hem ook nog veel geleerd over de Mainframe. Verder stond hij ook klaar om elke week te bellen om de proef te bespreken. Ik wil ook de Zweedse collega van Kevin, Bengt Gellingskog bedanken om samen met Kevin mij te ondersteunen met deze proef tijdens onze discussies over de Health Checker setup. Verder wil ik ook nog de andere collega's waaronder Louis Huysman en Alain de Waele van HCL te zuiderpoort bedanken en HCL Technologies BVBA zelf voor de mogelijkheid om deze proef te doen binnen hun infrastructuur.

Ik wil ook mijn promotor Thomas Pollet bedanken voor zijn feedback op de inhoud van mijn bachelorproef.

Tot slot wil ik ook mijn ouders bedanken voor de steun om mijn opleiding en mijn bachelorproef te voltooien. Waaronder de feedback op de schrijfwijze van mijn proef.

Ik wens u veel plezier toe bij het lezen van deze proef.

Samenvatting

De mainframe is een vitaal onderdeel van verscheidene belangrijke sectoren. Ze worden voornamelijk gebruikt door hun garantie van continue beschikbaarheid. Er is zowel software als hardware speciaal ontworpen voor dit doeleinde. z/OS Health checker is een voorbeeld van software dat hiervoor gebruikt wordt. Dit is een preventie tool die problemen probeert op te sporen voordat ze gebeuren. Maar de logging hiervan is niet zo efficiënt en de standaardopstelling van deze tool is ook niet efficiënt voor elke opstelling van een mainframe. Dit is ook het geval bij de opstelling HCL Technologies BVBA. Het doel van deze proef is de setup te optimaliseren en een veel efficiëntere manier van logging te bereiken. Er wordt eerst gekeken naar structuur binnen de mainframe waar z/OS health checker opereert. Dan naar de setup van z/OS Health Checker zelf en een analyse van die opstelling binnen HCL Technologies BVBA. Dan zal er een efficiëntere logging opgezet worden via JCL - jobs met als optie deze om te zetten naar een Web interface en hoe dit zou gebeuren.

Inhoudsopgave

1	Inleiding	15
1.1	Probleemstelling	16
1.2	Onderzoeksvraag	16
1.2.1	Deelonderzoeksvragen	16
1.3	Onderzoeksdoelstelling	16
1.4	Opzet van deze bachelorproef	17
2	Stand van zaken	19
2.1	De Mainframe	19
2.2	Logische partities en de Parallel Sysplex	20
2.2.1	Logische partitie of LPAR	20
2.2.2	Parallel Sysplex	21

2.3	z/OS	23
2.3.1	Storage gebruik van z/OS	23
2.4	Interactie met z/OS	25
2.4.1	Time Sharing Option/Extensions en Interactive System Productivity Facility	26
2.5	Job Control Language en Job Entry Subsystem	26
2.6	System Display and Search Facility	27
2.7	De Parmlib	28
2.8	z/OS Health Checker	28
2.8.1	Health Checker Framework	29
2.8.2	De Checks	29
3	Methodologie	33
3.1	z/OS Health Checker standaard setup	33
3.1.1	Analyse van huidige z/OS Health Checker Setup	33
3.1.2	Parmlib Members voor de standaard setup	36
3.2	Logging van z/OS Health Checker	37
3.2.1	JCL jobs voor het logging	38
3.2.2	Jobs schedulen met IWS	39
4	Vervolg op bachelorproef	41
4.1	Verwerken van SDSF output	41
4.1.1	REXX vs Java	41
4.2	Web Interface	42
5	Conclusie	43

A	Onderzoeksvoorstel	45
A.1	Introductie	45
A.2	State-of-the-art	45
A.3	Methodologie	46
A.4	Verwachte resultaten	46
A.5	Verwachte conclusies	46
A.6	Literatuurstudie	46
A.6.1	Inleiding	46
A.6.2	Mainframe	47
A.6.3	z/OS	47
A.6.4	LPAR of Logische Partitie	47
A.6.5	z/OS Health checker	48
B	Bijlagen	49
B.1	Check Output	49
B.2	Tabellen Standaard opstelling z/OS Health Checker	52
B.3	Parmlib members voor z/OS Health Checker	92
B.3.1	HZSPRM00	92
B.3.2	HZSPRMT1	95
B.3.3	HZSPRMT2	95
B.3.4	HZSPRMT3	98
B.3.5	HZSPRMT4	101
B.4	JCL jobs	104
B.4.1	FHZSVT11	104
B.4.2	FHZSVT21	105

B.4.3	FHZSVT31	105
B.4.4	FHZSVT41	106

Bibliografie	107
---------------------------	------------

Lijst van figuren

2.1	Mainframe	20
2.2	Logische Partities	21
2.3	Visualisatie van een Parallel Sysplex	22
2.4	z/OS binnen de mainframe omgeving	24
2.5	Visualisatie van het concept van virtuele storage	25
2.6	ISPF hoofdmenu	26
2.7	Visualisatie van JCL en JES	27
2.8	SDSF Hoofdmenu	28
2.9	z/OS Health Checker Framework	29
3.1	Health Checker Scherm binnen SDSF	34
3.2	ADDREPLACE statement	37
3.3	Plan voor de print jobs	39
3.4	Interval in IWS	39
3.5	Kalender IWS	40
3.6	Kalender IWS 2030	40
4.1	Websphere Application Server	42

Lijst van tabellen

3.1	Individuele check	34
3.2	Checks Per team	35
B.1	Health Checker Setup CICS team	53
B.2	Health Checker Communication team tabel 1	54
B.3	Health Checker Communication team tabel 2	55
B.4	Health Checker Communication team tabel 3	56
B.5	Health Checker Setup Print team	57
B.6	Health Checker Setup Automation team	58
B.7	Health Checker ROO team tabel 1	59
B.8	Health Checker ROO team tabel 2	60
B.9	Health Checker ROO team tabel 3	61
B.10	Health Checker ROO team tabel 4	62
B.11	Health Checker ROO team tabel 5	63
B.12	Health Checker ROO team tabel 6	64
B.13	Health Checker ROO team tabel 7	65
B.14	Health Checker ROO team tabel 8	66
B.15	Health Checker ROO team tabel 9	67

B.16	Health Checker ROO team tabel 10	68
B.17	Health Checker ROO team tabel 11	69
B.18	Health Checker ROO team tabel 12	70
B.19	Health Checker ROO team tabel 13	71
B.20	Health Checker ROO team tabel 14	72
B.21	Health Checker RTC team tabel 1	73
B.22	Health Checker RTC team tabel 2	74
B.23	Health Checker RTC team tabel 3	75
B.24	Health Checker RTC team tabel 4	76
B.25	Health Checker RTC team tabel 5	77
B.26	Health Checker Security team tabel 1	78
B.27	Health Checker Security team tabel 2	79
B.28	Health Checker SOE team tabel	80
B.29	Health Checker Storage team tabel 1	81
B.30	Health Checker Storage team tabel 2	82
B.31	Health Checker Storage team tabel 3	83
B.32	Health Checker Storage team tabel 4	84
B.33	Health Checker Storage team tabel 5	85
B.34	Health Checker Storage team tabel 6	86
B.35	Health Checker zOpen team tabel 1	87
B.36	Health Checker zOpen team tabel 2	88
B.37	Health Checker zOpen team tabel 3	89
B.38	Health Checker DB team tabel 1	90
B.39	Health Checker DB team tabel 2	91

1. Inleiding

Men zal het niet beseffen maar zonder de Mainframe zouden veel hedendaagse diensten wegvallen. Ook al is de mainframe voor velen iets uit het verleden, veel sectoren blijven er op vertrouwen. Een voorbeeld zijn de bankinstellingen, de kans is groot dat als je iets betaalt met je bankkaart dat de transactie verwerkt wordt door een mainframe. Een ander voorbeeld zijn de vliegmaatschappijen, als je eens een kijkje zou kunnen nemen naar de computer die gebruikt wordt op de luchthaven om je in te checken op je vlucht, zal je merken dat dit hoogstwaarschijnlijk een terminal is die aangesloten zit op een Mainframe. Verder wordt de mainframe ook nog gebruikt in andere sectoren zoals:

- Financiële Sector
- Magazijnbeheer
- Verzekeringen
- Ziekenzorg
- Overheid
- ...

Een uitval van een mainframe kan dus kritische diensten laten wegvallen. Bijvoorbeeld de mogelijkheid tot overschrijven via de bank. Daarom is een van de belangrijkste factoren van de mainframe de beschikbaarheid. De mainframe garandeert een uptime van 24 op 24, 7 op 7, 365 op 365 het hele jaar door dus. Hiervoor worden verscheidene technieken gebruikt zoals de "alles 2 regel". Je zult in een mainframe alles dubbel terugvinden. Ook de architectuur binnen de mainframe, de Parallel Sysplex (Meer hierover in hoofdstuk 2) zorgt dat bij een uitval van 1 partitie een andere zijn workload direct overneemt. En zijn er binnen die architectuur ook nog verschillende software componenten die hier voor zorgen. Een daarvan is degene waar deze proef zich op focust: z/OS Health Checker, deze tool werkt preventief. Het zal proberen om de problemen op te sporen alvorens die plaats vinden en

zal de System Administrator hiervan verwittigen. In hoofdstuk 2 zal de werking hiervan tot in detail worden uitgelegd.

1.1 Probleemstelling

De z/OS Health Checker opstelling van HCL Technologies is al lang niet meer veranderd en is ook niet gestandaardiseerd over de verschillende partities binnen de Mainframe. Bij eventuele uitbreiding van het systeem is er dus ook niet direct een standaardopstelling die men kan toepassen. Verder vind er ook een inefficiënte logging plaats. Deze is nu niet echt aanwezig, de bedoeling is dat de verantwoordelijke van elke partitie een log krijgt van alle fouten binnen zijn partitie.

1.2 Onderzoeksvraag

Deze proef zal zich focussen op de z/OS Health checker opstelling van HCL technologies met volgende onderzoeksvraag.

- Is het mogelijk een standaardopstelling te maken voor een z/OS Health Checker omgeving?

1.2.1 Deelonderzoeksvragen

Daarnaast is er in deze proef ook een focus op verdere efficiëntere logging van z/OS Health Checker en of deze kan via een webUI. Daaruit volgen de deelonderzoeksvragen:

- Kan er een efficiënt log-systeem opgezet worden voor de z/OS Health Checker output?

1.3 Onderzoeksdoelstelling

Het doel van deze proef is om een standaardopstelling te bekomen van z/OS Health Checker binnen de omgeving van HCL Technologies zodat deze bij uitbreiding van het systeem deze onmiddellijk kan implementeren op nieuwe partities.

Verder is het doel om ook een duidelijke logging op te stellen voor elke verantwoordelijke van elke logische partitie van de Mainframe omgeving van HCL, met als eventueel vervolg een mogelijkheid te vinden om die logging te laten gebeuren via een web interface.

1.4 Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie.

In Hoofdstuk 3 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de hoofdonderzoeksvraag en de 1ste deelonderzoeksvraag.

In hoofdstuk 4 word er gekeken naar eventuele vervolgen voor deze bachelor proef.

In Hoofdstuk 5, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Daarbij wordt ook een aanzet gegeven voor toekomstig onderzoek binnen dit domein.

2. Stand van zaken

Dit onderzoek zal zich focussen op z/OS Health Checker. Dit is een tool die draait in een mainframe omgeving. Vooraleer we kunnen beginnen met het bespreken van de oplossingsmethode van de onderzoeksvragen moeten we ons eerst verdiepen in de mainframe zelf en de omgeving waarin z/OS Health Checker draait om zo duidelijk te maken waarom deze tool zijn aanwezigheid belangrijk is en hoe deze werkt. Verder moeten we ook de basis begrijpen van andere tools en systemen binnen de mainframe zoals ISPF, SDS en JES. Om daarna te eindigen met JCL de taal die word gebruikt om de z/OS Health Checker logs op te stellen.

2.1 De Mainframe

De Mainframe speelt een centrale rol in de dagelijkse operaties bij de meeste grote bedrijven. De Ontwikkeling van de mainframe gaat terug tot de jaren '50. Ook al is er door de jaren heen veel veranderd aan de mainframe blijft het het meest stabiele, veilige en compatibele computing platform. Desondanks dat de mainframe een grote aanwezigheid heeft binnen de financiële wereld, blijft deze vrij onzichtbaar voor de grote menigte. Maar eigenlijk zijn we bijna allemaal indirect mainframe gebruikers ook al realiseren we het niet. (Ebbers, Kettner, O'Brien en Ogden, 2011)

De term mainframe kan vandaag het beste beschreven worden als een stijl van operaties, applicaties en besturing systeem faciliteiten. Een definitie hiervan zou zijn: "Een mainframe is hetgeen dat bedrijven gebruiken voor het hosten van commerciële databanken, transactie servers en applicaties die een hogere graad van security en availability nodig hebben dan die in machines van een kleinere schaal. De term mainframe is duidelijk

vervaagd met de jaren daarom wordt de term meestal geassocieerd met een systeem met volgende eigenschappen. (Ebbers e.a., 2011)

- Compatibiliteit met System z besturingssystemen, applicaties en data.
- Gecentraliseerde controle van resources
- Hardware en besturingssystemen die toegang kunnen delen tot disk drives met ander systemen met automatische locking en bescherming tegen destructief simultaan gebruik van data.
- Hardware en besturingssystemen die continu werken met honderden of duizenden simultane input-output operaties
- Clustering technologieën bevatten zoals de Parallel Sysplex die de mainframe flexibel en schaalbaar maken
- Geoptimaliseerd voor input-output business gerelateerde data processing applicaties



Figuur 2.1: 2 mainframes de System z Business Class en Enterprise class(Ebbers, Kettner, O'Brien en Ogden, 2011)

2.2 Logische partities en de Parallel Sysplex

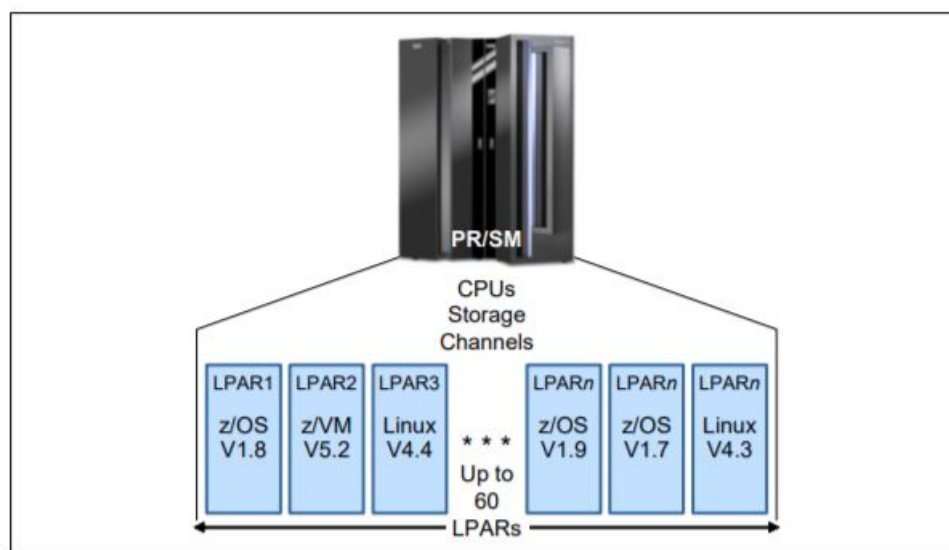
Nu men weet wat een mainframe is, is het belangrijk om te kijken naar de architectuur binnen het systeem zelf. Dit is aan de hand van allerlei gekoppelde logische partities die men als geheel de Parallel Sysplex noemt. Dit is ook de omgeving waarin z/OS Health Checker opereert. Daarom belichten we ook deze componenten in dit hoofdstuk.

2.2.1 Logische partitie of LPAR

De IBM mainframe kan verdeeld worden in verscheidene logische systemen. Tussen deze systemen kan men volgende resources verdelen:

- Memory
- Processors
- Input-Output devices.

Deze aparte systemen noemt men een logische partitie of LPAR. Al deze LPARs staan onder de controle van een hypervisor. De hypervisor is een software laag voor het beheer van meerdere besturingssystemen. De Verdeling van resources gebeurt door de Processor Resource/Systems Manager(PR/SM). De volledige definitie van een LPAR luidt als volgt: Een subset van de processor hardware dat gedefinieerd is voor het ondersteunen van een besturingssysteem. Meerdere LPARs zijn dus gelijkaardig aan verschillende aparte mainframes. Ze hebben elk hun eigen besturingssysteem en toegewezen hardware. (Ebbers e.a., 2011)



Figuur 2.2: Visualisatie van logische partities(Ebbers, Kettner, O'Brien en Ogden, 2011)

2.2.2 Parallel Sysplex

De Parallel Sysplex is een een techniek van clusteren. Waarmee men meerdere LPARs groepeer. z/OS Health Checker zal zowel opereren op aparte LPAR als op de gehele Parallel Sysplex door globale checks(meer hierover in sectie 2.8). Daarvoor gaan we ons ook verdiepen in deze clustering techniek.

Sysplex staat voor SYStems comPLEX dit is een of meerdere LPARs met z/OS, samengevoegd als 1 unit die gespecialiseerde hardware en software gebruikt. Het gebruikt unieke messaging services en kan bestandsstructuren delen in de couple facility(CF) datasets. Een sysplex is een instantie van een computer systeem dat draait op 1 of meerdere fysieke partities waarvan elke een andere release kan draaien van het z/OS besturingssysteem. Een sysplex is wel geïsoleerd tot 1 fysieke mainframe. De Parallel Sysplex anderzijds laat meerdere mainframes zich voordoen als 1 systeem. (Ebbers e.a., 2011)

Een Parallel Sysplex is een symmetrische sysplex die gebruik maakt van het delen van data

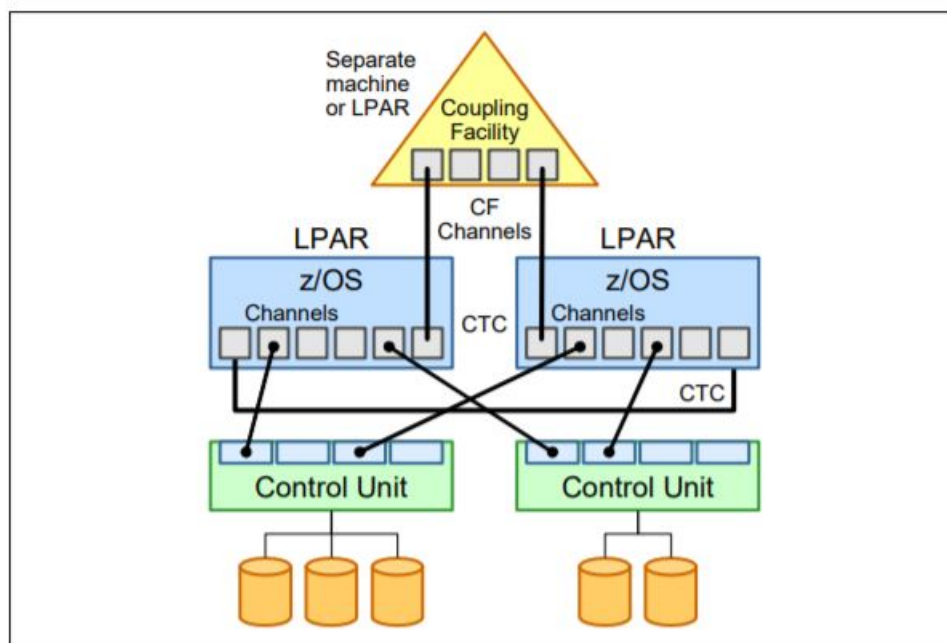
met meerdere systemen. Dit is dus de clustering van meerder mainframes. We bespreken ook enkele protocollen die de Parallel Sysplex gebruikt.

Server Time Protocol

Een belangrijk aspect van de Parallel Sysplex is het synchroniseren van de Time Of Day(TOD) klokken van de meerdere servers. Stel nu meerdere systemen hebben net in dezelfde database data aangepast, maar daarna gebeurt er een uitval. Dan zal men de databank reconstrueren met behulp van alle timestamps van alle aanpassingen. Hiervoor is het belangrijk dat de klokken van elke LPAR gesynchroniseerd zijn om zo de juiste data in de juiste volgorde te reconstrueren. Dit gebeurt vandaag met het Server Time Protocol(STP). (Ebbers e.a., 2011)

Coupling Facility

Sommige z/OS applicaties op verschillende LPARs hebben vaak toegang nodig tot dezelfde informatie. Hiervoor vertrouwt een Parallel Sysplex op een of meerder Coupling Facilities(CF). Een CF maakt het mogelijk om aan data sharing te doen met meerdere systemen. Een CF is ook een LPAR maar een speciale die andere LPARs toelaat data te delen. (Ebbers e.a., 2011)



Figuur 2.3: Visualisatie van een parallel sysplex met 2 LPARs en 1 Coupling Facility. Control Units controleren de logica voor bepaalde I/O-apparaten zoals printers of opslagfaciliteiten.(Ebbers, Kettner, O'Brien en Ogden, 2011)

Een goed geconfigureerde parallel sysplex cluster is zodanig ontworpen dat het een hoge beschikbaarheid biedt met een minimale onbeschikbaarheid. Dit is dus al een van technieken die de mainframe zijn hoge availability garanderen. Bijvoorbeeld: wanneer een

system uitvalt, dan kan een ander systeem dit direct opvangen omdat alle data en kritische applicaties gedeeld worden in de parallel sysplex. De taken van het uitvallende systeem worden overgenomen. Ondertussen kan het gefaalde systeem herstart worden. Dit zorgt er uiteindelijk voor dat een laag aantal single points of failure aanwezig is binnen de sysplex. (Ebbers e.a., 2011)

2.3 z/OS

Een ander belangrijk component van elk systeem is het besturingssysteem. In het geval van deze proef is dat z/OS. Dit niet het enige maar wel het meest gebruikte besturingssysteem op de mainframe. Zoals de naam het zelf zegt draait ook z/OS Health Checker op dit besturingssysteem. Daarom bespreken we ook dit besturingssysteem. Een besturingssysteem is eigenlijk een collectie van programma's die de interne werking van het computer systeem beheren. Een besturingssysteem is ontworpen om er voor te zorgen dat de resources van de computer optimaal gebruikt worden.

z/OS is vandaag een resultaat van tientallen jaren technologische vooruitgang door IBM. Het begon als een besturingssysteem dat maar 1 programma tegelijk kon afhandelen naar een dat vandaag duizenden programma's en gebruikers tegelijk kan afhandelen. Het besturingssysteem wordt uitgevoerd in de processor en bevindt zich ook in de processor storage. Mainframe hardware bestaat uit een aantal processors en gekoppelde toestellen zoals DASD(Direct Acces storage devices de mainframe term voor hard disks). Die worden dan allemaal aangestuurd vanuit de consoles gekoppeld aan de mainframe. De DASD's worden gebruikt voor systeem functies of door programma's van gebruikers die uitgevoerd worden door z/OS. (Ebbers e.a., 2011)

z/OS maakt het ook mogelijk om aan multiprocessing en multiprogramming te doen. Hierdoor maakt het z/OS geschikt voor het uitvoeren van programma's die veel input/output operaties nodig hebben.

Multiprocessing

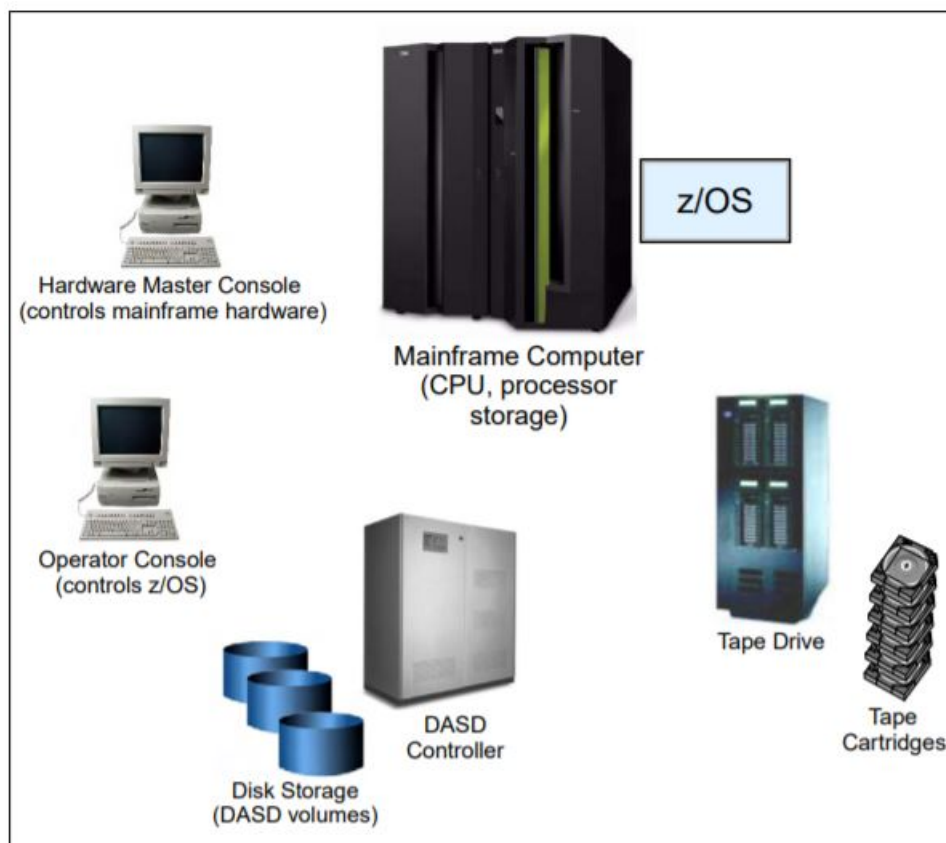
Dit is het simultaan opereren van meerdere processors die meerdere hardware resources delen zoals memory of externe opslag

Multiprogramming

Multiprogramming laat z/OS toe om duizenden programma's te draaien voor gebruikers die werken aan verschillende projecten waar men zich ook bevindt op de wereld.

2.3.1 Storage gebruik van z/OS

z/OS heeft ook zijn eigen manier voor het gebruik van opslag/storage. Dit is ook belangrijk, omdat de term 'address space' gebruikt zal worden binnen deze proef. En die term is de



Figuur 2.4: z/OS binnen de hardware omgeving bevindt zich op de mainframe. En wordt aangestuurd door aangesloten consoles/terminals. Om dan de data te bewerken op tape drives of DASD (term voor hard disk binnen de mainframe wereld) Ebbers, Kettner, O'Brien en Ogden, 2011

techniek die z/OS gebruikt binnen de storage omgeving.

Een mainframe en een gewone computer hebben 2 soorten fysieke opslag:

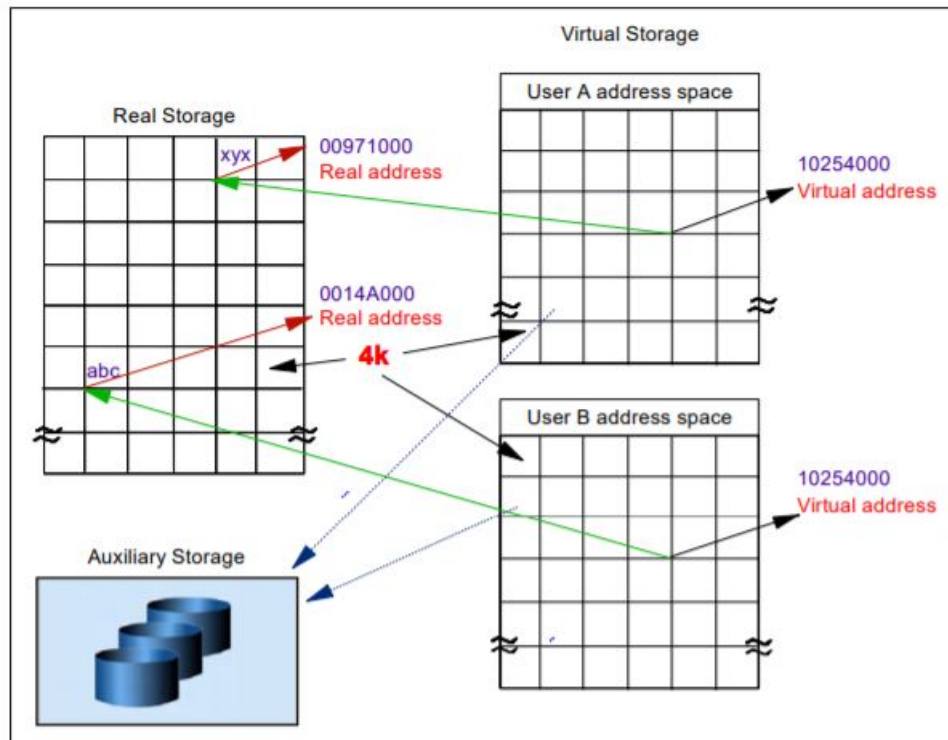
- Fysieke storage die zich bevindt op de mainframe processor zelf, ook wel 'real storage' genoemd te vergelijken met RAM op je laptop.
- Fysieke storage die zich buiten de mainframe bevindt op een tape of disk drive, dit wordt de auxiliary storage genoemd.

z/OS gebruikt deze 2 soorten storage om een andere soort te vormen namelijk virtual storage. Virtual storage is een combinatie van real en auxiliary storage. Het gebruikt een serie van tabellen en indexen om locaties binnen het real geheugen te associëren met locaties in de auxiliary storage.

Een address space is eigenlijk een range van virtuele adressen dat het besturingssysteem toekent aan een programma of gebruiker. Voor een gebruiker kan dit beschouwd worden als een container waar zijn data in zit. Door deze adres space moet z/OS niet een heel programma naar de real storage laden om het uit te voeren. Daarom zal men het programma in stukken (ook gekend als pages) van de auxiliary storage naar de real storage verplaatsen

in de volgorde die nodig is om het programma uit te voeren. Eens dat een page niet meer nodig is kan men het terugschrijven naar de auxiliary storage. Dit laat z/OS toe om meer programma's simultaan uit te voeren.

De fysieke opslag is daarom opgedeeld in verschillende stukken die elke hun eigen adres hebben maar de pages worden opgevraagd met hun virtueel adres. Het proces om een virtueel adres te vertalen in een real address noemt men Dynamic Address translation(DAT). (Ebbers e.a., 2011)

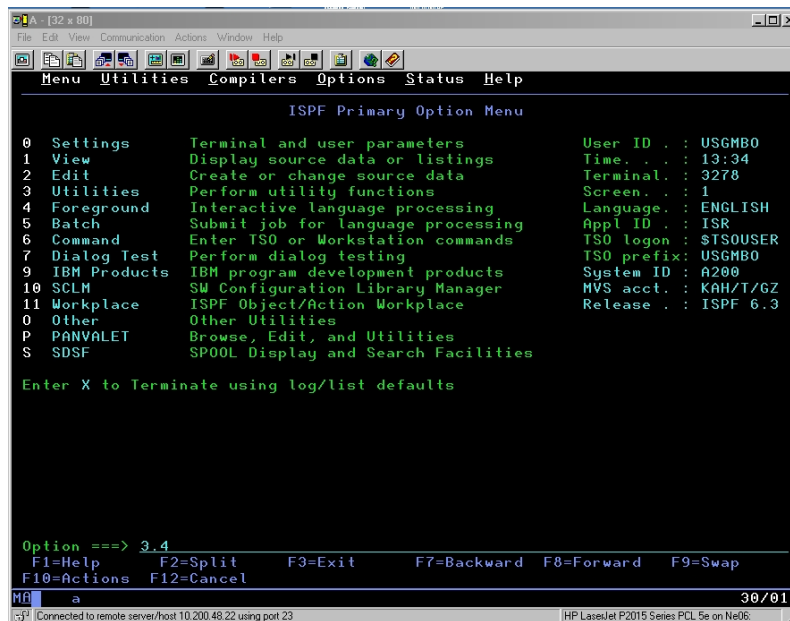


Figuur 2.5: Bijna alle programma's gebruiken virtuele adressen als ze refereren naar data in de real storage. Maar als een aangevraagd adres zich niet in de real storage bevindt zal er een onderbreking plaatsvinden en zal men de nodige data uit auxiliary storage naar de real storage laden(ook wel paging genoemd)(Ebbers, Kettner, O'Brien en Ogden, 2011)

Een andere manier om te denken aan een address space is eigenlijk een soort van kaart voor de programmeur waarmee hij al zijn code en data kan opvragen.

2.4 Interactie met z/OS

Om een systeem te beheren moeten we er natuurlijk ook mee kunnen werken. Voor deze interactie gebruikt men bijvoorbeeld een TSO/E commando om aan te loggen en dan ISPF, een collectie van menu's en panels die brede range van functies aanbied.



Figuur 2.6: Dit is het hoofdmenu van ISPF vanaf hier kan je de verschillende panelen gebruiken

2.4.1 Time Sharing Option/Extensions en Interactive System Productivity Facility

TSO/E of Time Sharing Option/Extensions laat gebruikers toe om een interactieve sessie te maken met een z/OS systeem. Hierdoor kunnen ze aanloggen op het z/OS systeem en gebruik maken van een command prompt interface. Maar omdat de command prompt niet echt handig is wordt er meestal gebruik gemaakt van de Interactive System Productivity Facility (ISPF). Dit is een collectie van menu's en panelen die een wijde range van functies aanbied voor het bewerken van data in z/OS. Zo bied ISPF onder andere een tekst editor aan en functies voor het vinden en ophoofden van bestanden. (Parziale, Fadel en Jon, 2017)

De bestanden binnen z/OS worden ook wel 'data sets' genoemd, dit is de term die ook verder gebruikt zal worden als we over bestanden spreken binnen z/OS. Er zijn 2 soorten data sets waarmee gewerkt wordt in deze proef.

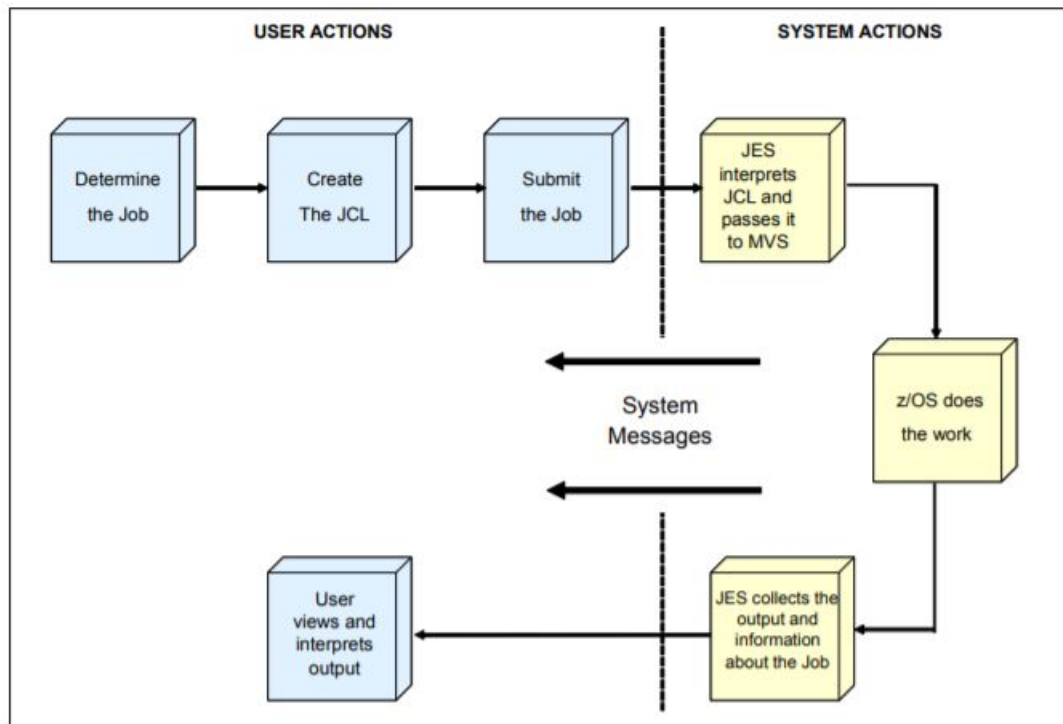
- Sequential data set: dit zijn data sets waarvan de individuele records georganiseerd zijn op hun fysieke volgorde binnen de dataset.
- Partitioned data set (PDS): dit is een data set die verdeeld is in partities ook wel 'members' deze kunnen een programma bevatten of gewoon data. Eigenlijks is een PDS een collectie van sequentiële data sets. Je zou het dus kunnen vergelijken met een folder met bestanden.

2.5 Job Control Language en Job Entry Subsystem

In deze proef worden ook Jobs uitgevoerd voor het opstellen van de logs van z/OS Health Checker. Hiervoor wordt de Job Control Language (JCL) gebruikt. Want om een

programma uit te voeren moet het verwerkt worden door z/OS.

Vooraleer het z/OS systeem een programma voor je kan uitvoeren moet je een paar dingen doen. Je moet eerst natuurlijk beschrijven welk programma je wil uitvoeren maar ook welke resources er gebruikt zullen worden (bv: eventuele input). Dit doe je met een JCL job. Deze jobs 'submit' je dan naar het Job Entry Subsystem (JES) deze zal de jobs inplannen en hun output verwerken. (Cosimo en Kuehner, 2018)



Figuur 2.7: De gebruiker definieert, maakt en zal een job submitten. Deze wordt dan verwerkt door het Job Entry Subsystem en zal de output van de job teruggeven als resultaat.

2.6 System Display and Search Facility

We willen natuurlijk de output van onze jobs kunnen zien daarvoor wordt er gebruikt gemaakt van de System Display and Search Facility of SDSF. Dit is niet het enige dat we kunnen zien in SDSF. Voor z/OS Health Checker zullen we ook deze tool gebruiken om alle info van het systeem op te vragen. Deze functie kan je bereiken via het hoofdmenu van ISPF met de optie 'S' (zie figuur 2.6)

In SDSF zijn er veel panelen en soorten output die je kan opvragen maar de enige die wij zullen gebruiken is namelijk die van z/OS Health Checker die bereik je binnen SDSF via de 'ck' optie. Verder zal je bij het schrijven van jobs het 'Status of jobs' paneel nodig hebben. Dit paneel toont de output van uitgevoerde jobs, dit paneel bereik je met de 'st' optie.

```

Display Filter View Print Options Search Help
-----
HQX77A0 ----- SDSF PRIMARY OPTION MENU -----
COMMAND INPUT ==> DA_ SCROLL ==> PAGE

DA  Active users          INIT  Initiators
I   Input queue           PR   Printers
O   Output queue          PUN  Punches
H   Held output queue     RDR  Readers
ST  Status of jobs        LINE  Lines
JG  Job groups            NODE  Nodes
SYM  System symbols       SO   Spool offload
LOG  System log           SP   Spool volumes
SR  System requests       NS   Network servers
MAS  Members in the MAS   NC   Network connections
JC   Job classes          RM   Resource monitor
SE   Scheduling environments CK  Health checker
RES  WLM resources        LNK  Link list data sets
ENC  Enclaves             LPA  Link pack data sets
PS   Processes            APF  APF data sets
SYS  System information   PAG  Page data sets
ENQ  Enqueues             PARM Parmlib data sets
END  Exit SDSF            ULOG User session log

```

Figuur 2.8: Hier zie je het hoofdmenu van SDSF en onder andere ook de 2 opties die gebruikt worden doorheen de proef namelijk 'ck' en 'st'

2.7 De Parmlib

z/OS Health Checker heeft ook enkele datasets in de parmlib die veranderd worden in deze proef daarom moeten we ook begrijpen wat die parmlib eigenlijk is. Elk z/OS systeem heeft een PDS met members die worden meegegeven door IBM. Deze PDS is SYS1.PARMLIB. De members zijn allemaal systeem en applicatie parameters die het systeem nodig heeft bij het opstarten (Initial Program Load (IPL) in mainframe term). Bij de IPL wordt de parmlib gelezen om het systeem op te zetten. Deze PDS wordt later ook nog gelezen door andere componenten en programma's. (Cosimo en Kuehner, 2018)

Een van deze componenten is z/OS Health Checker. In de parmlib zitten members die definiëren welke controles (Checks) er zullen uitgevoerd worden en welke niet.

2.8 z/OS Health Checker

Na een analyse naar de oorzaken van de verschillende uitvallen kwam men tot de conclusie dat veel hiervan perfect vermeden had kunnen worden. Vele uitvallen kwamen door slechte configuraties die leiden tot single points of failure. Hierdoor is z/OS Health Checker ontwikkeld door IBM. (Walle, 2013)

IBM Health Checker voor z/OS is een tool die helpt om potentiële problemen op te sporen in de configuratie van het systeem. Deze problemen zouden een grote impact kunnen hebben op het systeem of zouden zelfs een uitval kunnen veroorzaken. Health Checker kijkt de huidige instellingen van z/OS en de Sysplex na en vergelijkt deze met instellingen die door IBM aangeraden worden. Bij eventuele problemen zal Health Checker een

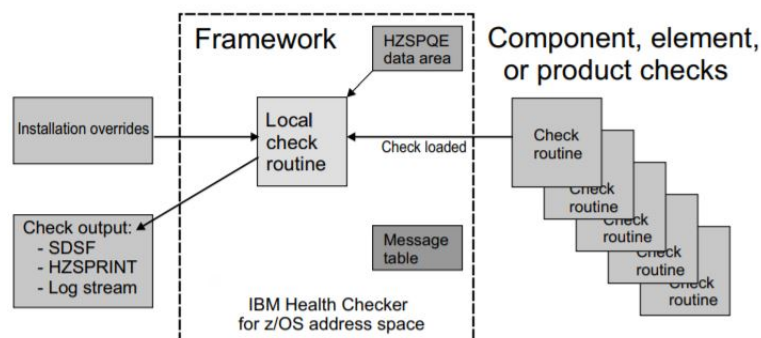
output genereren met gedetailleerde info over het probleem zelf en op welke manier je het probleem het beste oplost. Wel belangrijk is dat Health Checker eerder een preventieve tool is en geen monitoring tool. (Bezzi, O'Conner, Philips en Thiemann, 2010)

z/OS Health Checker bestaat uit 2 delen

- De checks
- Het framework

2.8.1 Health Checker Framework

Het framework van z/OS Health Checker is de interface die je gebruikt om checks uit te voeren en te beheren. Deze ondersteunt niet enkele checks van IBM maar ook die van software producenten zoals Computer Associates en Compuware. Binnen het framework zit de HZSPQE data, hierin zit alle informatie die een check nodig zou hebben zoals de default parameters. Het framework bevat ook de Message table die de data bijhoudt van de output van de checks. (IBMCorporation, 2019)



Figuur 2.9: Het z/OS Health Checker Framework dat het mogelijk maakt om checks uit te voeren en de output hiervan door te geven aan SDSF en andere tools.(IBMCorporation, 2019)

2.8.2 De Checks

De Checks zelf zijn programma's die componenten en instellingen evalueren en dan eventuele problemen melden. Deze checks worden voornamelijk aangeboden door IBM zelf maar deze kunnen ook van andere bedrijven zijn. Je kan ook zelf Checks schrijven met System REXX. Een check vergelijkt de instellingen van 1 bepaald component met een set van instellingen die aangeraden worden door de eigenaar van de check. De aangeraden instellingen zijn geen verplichte instellingen maar eerder een best practice van IBM. Je kan deze ook zelf definiëren of aanpassen. (Bezzi e.a., 2010)

Een check wordt gedefinieerd met 3 waarden

Check Owner

Elke check heeft zijn eigenaar. De naam van de check is meestal verbonden met het component waarvoor hij draait. In deze naam zit ook bijna altijd een verwijzing naar het bedrijf of product die de check gemaakt heeft. De checks van IBM beginnen dan ook allemaal met IBM. Een check owner is een string van maximum 16 karakters. Een voorbeeld van een check owner is bijvoorbeeld IBMXCF. Deze check is dus geschreven door IBM en zal iets te maken hebben met XCF(Cross-system Coupling Facility). Een check owner heeft meerdere checks zelf. (Bezzi e.a., 2010)

Check Name

De check zelf heeft natuurlijk ook een naam. En is uniek voor elke check. Deze zal maximum 32 karakters lang zijn. Een voorbeeld van een check naam: "XCF_CDS_MAXSYSTEM".

Check Values

Een check bezit verder ook nog enkele vooraf gedefinieerde waarden

1. Een interval die definieert hoe vaak en wanneer de check uitgevoerd wordt.
2. De ernst(Severity) van een check deze definieert de check output. Hoe grotere het potentiële probleem hoe hoger de severity.

Check Types

Verder zijn er ook 3 types van checks: Local, remote en REXX checks. In deze proef komen enkel de local checks aan bod. De lokale check is geschreven in ASSEMBLER. En draaien binnen de Health Checker address space. Deze worden opgeroepen met parameters die verwijzen naar de HZSPQE data hierin zit alle data die de check routine nodig heeft. Het verschil met een Remote check is dat een remote check niet binnen de adres space van Health Checker draait.

Check Output

De check output is ook zeer belangrijk deze duid namelijk aan of een check al dan niet succesvol was. Er zijn verschillende soorten check messages.

- Information Message: Deze message krijg je wanneer een check succesvol is of wanneer deze niet kan draaien in de huidige omgeving(Bijvoorbeeld wanneer het component dat de check evalueert niet aanwezig is op het systeem).
- Exception Message: Deze message krijg je wanneer de check onsuccesvol was en deze een potentieel probleem heeft gevonden. Dit noemt een exception. Dit bericht bevat de severity van het probleem samen met suggesties om het probleem op te lossen
- Report: Bij een exception zal er ook een extra report bijgevoegd zijn met extra

informatie over het probleem.

- **Debug:** Sommige checks kan je in debug mode uitvoeren. Dit wordt vooral enkel gebruikt bij het ontwikkelen van een eigen check.

Voor een voorbeeld van check output zie bijlage B.1

Check Status

Een check heeft ook een bepaalde status. Deze status duid aan of dat een check al dan niet geactiveerd is. Volgende statussen zijn mogelijk. (IBM, 2019)

- **ACTIVE:** Dit specificeert dat de check actief is en draait.
- **INACTIVE:** Dit specificeert dat de check niet actief is en deze niet zal draaien.
- Ookal is een check actief of inactief kan deze ook nog 2 opties hebben namelijk Enabled & Disabled. Wanneer een check enabled is betekent dat dat deze kan draaien binnen de huidige omgeving. Als een check disabled is kan dit betekenen dat deze niet kan draaien omdat er condities binnen de huidige parallel sysplex omgeving zijn die dat verhinderen. Bijvoorbeeld een check van DB2 die disabled staat omdat er om een bepaalde LPAR geen DB2 product is.
- **GLOBAL:** Dit betekent dat de check globaal is en daarom maar op 1 LPAR mag draaien van de gehele sysplex. Maar de check zal wel instellingen controleren voor de hele parallel sysplex.

Verder is er nog 1 speciaal soort check. Namelijk de migration check. Deze check helpen je bij het plannen voor een overschakelen van z/OS versie. Deze checks kan je uitvoeren na een migratie om te kijken of deze succesvol was. Deze checks beginnen altijd met 'ZOSMIG'. (IBMCorporation, 2019)

3. Methodologie

Na de stand van zaken die gebaseerd is op de literatuurstudie volgt de volgorde van stappen die ondernomen zijn om deze proef te voltooien.

3.1 z/OS Health Checker standaard setup

De eerste onderzoeksvraag was of er mogelijkheid was tot een standaardopstelling binnen de z/OS Health Checker omgeving van HCL Technologies. Maar eerst moet er een analyse plaatsvinden op de huidige opstelling van Health Checker. Om deze te optimaliseren

3.1.1 Analyse van huidige z/OS Health Checker Setup

De opstelling die in deze proef geanalyseerd werd bevindt zich binnen een parallel sysplex. En deze parallel sysplex werken we met 4 LPARS: VT1, VT2, VT3 en VT4. Elke LPAR heeft verschillende checks. Maar na de 4 LPARS te overlopen was het duidelijk dat de meeste checks op VT1 draaien. Daarom is de analyse gefocust op VT1.

De analyse is gemaakt met de check data uit SDSF deze kan je bereiken door bij het ISPF hoofdmenu volgende optie te geven 's;ck'. Dit is S voor SDSF met als volgende optie CK voor Health Checker deze ziet er zo uit.

Na het overlopen van alle checks op VT1 hebben we de SDSF output samengevat in volgende tabel. Met de naam van de check. De status, deze beschrijft of de check aanstaat of niet. De outcome, deze beschrijft of de check succesvol was of niet. En de reason, dit is de reden waarvoor de check draait. Een voorbeeld:

```

SDSF HEALTH CHECKER DISPLAY VT01 LINE 1-34 (300)
COMMAND INPUT == SCROLL ==> PAGE
PREFIX=6350842* DEST=(ALL) OWNER=* SYSNAME= CheckOwner
NAME State Statu
ACF2_AUTO_START_CHECK CA_ACF2 ACTIVE(ENABLED) SUCCE
ACF2_CHECK_DATABASES CA_ACF2 ACTIVE(ENABLED) SUCCE
ACF2_CHECK_EXITS CA_ACF2 INACTIVE(ENABLED) INACT
ACF2_CHECK_JES2_EXITS CA_ACF2 ACTIVE(ENABLED) SUCCE
ACF2_SAFDEF_NOARF_CHECK CA_ACF2 INACTIVE(ENABLED) INACT
ALLOC_ALLC_OFFLN_POLICY IBMALLOC ACTIVE(ENABLED) SUCCE
ALLOC_SPEC_WAIT_POLICY IBMALLOC ACTIVE(ENABLED) SUCCE
ALLOC_TAFELTB_PREF IBMALLOC INACTIVE(ENABLED) INACT
ALLOC_TIOT_SIZE IBMALLOC ACTIVE(ENABLED) SUCCE
ASM_LOCAL_SLOT_USAGE IBMASM INACTIVE(ENABLED) INACT
ASM_NUMBER_LOCAL_DATASETS IBMASM INACTIVE(ENABLED) INACT
ASM_PAGE_ADD IBMASM INACTIVE(ENABLED) INACT
ASM_PLPA_COMMON_SIZE IBMASM INACTIVE(ENABLED) INACT
ASM_PLPA_COMMON_USAGE IBMASM INACTIVE(ENABLED) INACT
CATALOG_ATTRIBUTE_CHECK IBMCATALOG ACTIVE(ENABLED) SUCCE
CATALOG_IMBED_REPLICATE IBMCATALOG INACTIVE(ENABLED) INACT
CATALOG_RAILS IBMCATALOG ACTIVE(ENABLED) SUCCE
CCS_ENF_SCREEN_VALIDITY CA_CCS ACTIVE(ENABLED) SUCCE
CICS_CDRA_ACCESS IBMCIICS INACTIVE(ENABLED) INACT
CICS_JOB SUB_SPOOL IBMCIICS INACTIVE(ENABLED) INACT
CICS_JOB SUB_TDQINTRDR IBMCIICS INACTIVE(ENABLED) INACT
CNZ_AHPE_EVENTUAL_ACTION_MSGS IBMCNZ ACTIVE(ENABLED) SUCCE
CNZ_CONSOLE_MASTERAUTH_CHDSYS IBMCNZ ACTIVE(ENABLED) SUCCE
CNZ_CONSOLE_MSCOPE_AND_ROUTECODE IBMCNZ INACTIVE(ENABLED) INACT
CNZ_CONSOLE_ROUTECODE_11 IBMCNZ ACTIVE(ENABLED) SUCCE
CNZ_EMCS_HARDCOPY_MSCOPE IBMCNZ INACTIVE(ENABLED) INACT
CNZ_EMCS_INACTIVE_CONSOLES IBMCNZ ACTIVE(ENABLED) SUCCE
CNZ_OBSOLETE_MSPFLD_AUTOMATION IBMCNZ ACTIVE(ENABLED) SUCCE
CNZ_SYSCONS_ALLOWCMD IBMCNZ INACTIVE(ENABLED) INACT
CNZ_SYSCONS_MSCOPE IBMCNZ ACTIVE(ENABLED) SUCCE
CNZ_SYSCONS_PD_MODE IBMCNZ ACTIVE(ENABLED) SUCCE
CNZ_SYSCONS_ROUTECODE IBMCNZ ACTIVE(ENABLED) SUCCE
CNZ_TASK_TABLE IBMCNZ ACTIVE(ENABLED) SUCCE
OSAPP_FTPD_ANONYMOUS_JES IBMCS ACTIVE(ENABLED) SUCCE
F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE
*ISFPCU4

```

Figuur 3.1: Dit is het Health Checker paneel binnen SDSF hier kunnen we de output van de laatste keren dat een check is uitgevoerd

Name	Status	Outcome	Reason
XCF_CDS_MAXSYSTEM	ACTIVE(ENABLED)	SUCCE	CDS MAXSYSTEM value across all CDS types should be at least equal to the value in the primary sysplex CDS.

Tabel 3.1: Waarden van de check die bij de eerste analyse werden samengevat

Voor de volledige tabel zie bijlage

Hier uit bleek dat er verscheidene checks een exception hadden. Deze zijn eerst gecontroleerd of dat men deze zelf kon oplossen zonder ondersteuning van de collega's die het product beheren. Bij deze analyse bleek ook dat er veel checks zijn die onnodig draaiden omdat het product van de check bijvoorbeeld niet aanwezig was. Niet alle checks moeten in alle LPARs draaien en sommige moesten aangepast worden. Maar daarvoor heb je nog de belangrijke vraag of een wijziging van een check doorgevoerd moet worden naar alle LPARs of maar naar 1 LPAR binnen de Parallel Sysplex. Met deze informatie is naar feedback gevraagd van alle Mainframe Teams binnen HCL Technologies. Want men kan checks niet aanpassen van bijvoorbeeld DB2 zonder feedback van het DB team. Hiervoor zijn alle checks gegroepeerd per owner daarna nog eens gegroepeerd per mainframe team en uiteindelijk werd tabel 3.2 bereikt. Deze tabellen zijn er gekomen na veel overleg om zo tot de beste setup te bekomen.

Nog een kleine verduidelijking voor alle teams die belang hebben bij de z/OS Health Checker setup.

- CICS: Dit team is verantwoordelijk voor de IBM CICS Software¹
- Communication: Dit team is verantwoordelijk voor alle communicatie-gerelateerd software zoals TCP/IP, FTP², etc.

¹Customer Information Control System dit is software voor het verwerken van transacties (Rayns, 2011)

²File Transfer Protocol

- **Print:** Dit team is verantwoordelijk voor alle output software zoals CA-View³, VPS, etc.
- **Automation:** Dit team is verantwoordelijk voor alles wat met automatisering te maken heeft. Vooral software zoals System Automation, NetView en GDPS. Dit team is het meest betrokken bij monitoring.
- **ROO:** Roll-Out & Operate, dit team beheert de 'basics' van z/OS. Dit team voert ook de upgrades uit.
- **RTC:** RunTime Control, dit team is verantwoordelijk voor performance en rapportering.
- **Security:** Dit team spreekt voor zich het beheert toels zoals RACF/ACF2/Top Secret dit zijn allemaal tool die extra beveiliging aanbieden voor de mainframe.
- **SOE:** Standard Operating Environment, dit team doet alle installaties van software voor de mainframe.
- **Storage:** Dit team beheert alles wat te maken heeft met opslag en ook de software die erbij hoort.
- **zOPEN:** Dit team beheert alles van zLinux⁴ samen met Unix System Services⁵ en Websphere⁶
- **DB:** Database team, beheert alles database systemen waaronder DB2.

Alle Check owners gegroepeerd per team.

Team	Checks(Owner)
Cics	IBMCICS
Communication	IBMCS, CA_TPX
Print	CA_DLVR, CA_SPOOL, CA_VIEW
Rollout and Opperate(ROO)	CA_CSS, IBMCNZ, IMBCTRACE, IBMDAE, IBMISPF, IBMXLOGR, IBMJES(2), IBMRRS, IBMRTM, IBMSDSF, IBMSDUMP, IBMSLIP, IBMSVA, IBMSYSTRACE, IBMTIMER, IBMTSOE, IBMXCF, IBMGRS
Run Time Control(RTC)	IBMVLF, CA_PMO, IBMRCE, IBMASM, IBMRSM, IBMSUP, IBMVSM, CPWR_THRUPUT_MGR
Security	CA_ACF2, IBMICSF
SOE	All checks starting with ZOSMIG
Storage	CA_DISK, IBMALLOC, IBMCATALOG, IBMDMO, IBMHSM, IMBIOS, IBMOCE, IMBPDSE, IBMSMS, IBMVSAM(RLS), CA_VANTAGE
zOpen	IBMSSH, IBMUSS, IBMZFS
Databse(DB)	CA_DB2, CA_DTCM, CA_IDMS
Automation	IBMGDPS

Tabel 3.2: Alle checkowners gegroepeerd per team binnen HCL

³Tool van Computer Associates

⁴Een linux besturingsysteem voor mainframe

⁵UNIX besturingsysteem implementatie voor de mainframe

⁶Software product gericht op web-technologie

Nadat alle checks per team werden gegroepeerd, is er eerst zelf een voorstel gemaakt voor de opstelling van hun checks waar zij feedback op konden geven. Wanneer er geen feedback was is de voorgestelde opstelling geïmplementeerd. Als een check gewijzigd werd moet dit gedefinieerd worden in de parmlib. Maar men moet ook weten of een wijziging in alle LPARs moet gebeuren of maar in 1 LPAR. Daarom hebben we bij de tabellen van de gehele setup nog 2 extra kolommen.

- Run: Dit definieert dat de check uitgevoerd moet worden(Yes of No). Of dat deze check aangepast moet worden(MOD).
- 00/&SUF: Dit definieert dat de check voor alle LPARs wordt uitgeschakeld (00) of maar voor 1 LPAR(&SUF⁷). Wanneer er N/A(not applicable) staat betekent dit dat er niets verandert zal worden aan de check.

Verder was er nog 1 uitzondering voor het beheren van de checks. Het SOE team beheert alle migratie checks en deze zullen ongewijzigd blijven in de parmlibs. De migratie checks zelf worden maar 1 keer geactiveerd.

De volledige health Checker setup zoals ze na deze proef is bevind zich in bijlage B.2. Dit is een samenvatting van hoe HCL wilt dat de setup eruit ziet.

3.1.2 Parmlib Members voor de standaard setup

Om dit door te voeren moeten we de parmlib members wijzigen. Als een check over het gehele systeem wordt uitgeschakeld gebeurt dit in HZSPRM00. Is het voor een enkele LPAR dan moet de wijziging gedefinieerd worden in de parmlib member van die LPAR. Er zijn 5 parmlib members voor z/OS Health Checker binnn de setup van HCL Technologies.

1. HZSPRM00: Wijzigingen in deze member gelden voor alle LPARs binnen de parallel sysplex.
2. HZSPRMT1: Wijzigingen in deze member gelden enkel voor VT1
3. HZSPRMT2: Wijzigingen in deze member gelden enkel voor VT2
4. HZSPRMT3: Wijzigingen in deze member gelden enkel voor VT3
5. HZSPRMT4: Wijzigingen in deze member gelden enkel voor VT4

Deze wijzigingen gebeuren met een ADDREPLACE-statement. De statement in figuur 3.2 doet het volgende:

- Statement(NOPLEX) specificeert de naam van de wijziging
- Update specificeert dat de check zal blijven draaien maar met andere parameters. Dit kan ook 'inactive' zijn voor de check te deactiveren of 'Delete' om de check te verwijderen.
- Check(,) definieert de check owner en de check zelf die gewijzigd wordt.
- Parm('NOPLEX') dit is de parameter die meegegeven wordt voor de wijziging.
- Reason() dit is een zelf in te vullen reden voor het wijzigen van de check.
- Date(20090701) dit is de datum van de wijziging in yyyyymmdd formaat. Bezzi e.a.,

⁷Dit is een omgevings variabele dat de huidige LPAR specificeerd

2010

```

BROWSE    SYS1.PARMLIB(HZSPRM00) - 01.06          Line 00000130 Col 001 080
Command ==>                                         Scroll ==> CSR
/*
ADDREPLACE
  POLICY(ZOS1A0) STATEMENT(NOPLEX)
  UPDATE
    CHECK(IBMUS, USS_FILESYS_CONFIG)
    PARM('NOPLEX')
    REASON('This baby aint no SYSPLEX')
    DATE(20090701)
  /*

```

Figuur 3.2: Voorbeeld van een ADDREPLACE Statement (IBM Corporation, 2019)

Bij de analyse van de setup bleek ook dat er veel checks waren die de GLOBAL status hadden. Maar deze draaiden verspreid over de 4 LPARs. Voor de nieuwe standaard setup van z/OS Health Checker wenste HCL Technologies dat alle GLOBAL checks draaide op de VT1 LPAR, hiervoor moesten dus alle GLOBAL checks inactief gezet worden op alle LPARs buiten VT1. Dit is mogelijk door dat te doen in volgende parmlib members: HZSPRMT2, HZSPRMT3 en HZSPRMT4. Met deze requirements zijn de 4 parmlib members aangemaakt. Deze bevinden zich in bijlage B.3

Nadat de parmlib members aangepast werden, moeten ze nog door het systeem geïmplementeerd worden. Dit doet men met volgende commando's in SDSF.

- /F HZSCHK,REPLACE,PARMLIB=(00,T1) voor LPAR VT1.
- /F HZSCHK,REPLACE,PARMLIB=(00,T2) voor LPAR VT2.
- /F HZSCHK,REPLACE,PARMLIB=(00,T3) voor LPAR VT3.
- /F HZSCHK,REPLACE,PARMLIB=(00,T4) voor LPAR VT4.

Verder is er nu ook een HSPRMTK member dit is een kopie van HZSPRMT4. Als het systeem zou uitbreiden met een LPAR, zal deze member gebruikt worden om z/OS Health checker in te stellen voor de nieuwe LPAR. de HZSPRMT4 member is gekozen omdat deze het meest geschikt is voor nieuwe LPARs.

3.2 Logging van z/OS Health Checker

Nu er een standaard setup van z/OS Health Checker tot stand is gekomen. Is er nog een systeem nodig dat de potentiële problemen kan loggen. Want SDSF zal geen berichten sturen naar gebruikers dat er een probleem is. Via SDSF zal je enkel zelf kunnen kijken tussen alle checks om te zoeken welke een exception hebben. Daarom zijn er JCL jobs geschreven, die voor elke LPAR een mail gaan sturen met alle checks die in exception staan. Deze mail gaat naar een team in India dat ervoor zal zorgen dat de problemen opgelost worden. Omdat checks meerdere keren draaien zal er met IBM Tivoli Workload

scheduler een plan gemaakt worden. Met dit plan zullen de jobs automatische elke vrijdag draaien.

3.2.1 JCL jobs voor het logging

De jobs zelf bevinden zich in bijlage B.4. De JCL job in bijlage B.4.1 zal gebruikt worden om de werking van de job uit te leggen.

De eerste regels van de job definiëren zijn naam en zijn eigenschappen. Deze job wordt dan gebonden aan VT01 omdat hij voor deze LPAR draait.

```
//FHZSVT11 JOB (610VV110000,4352,,1800),'RTN=FHZSPRVT',CLASS=D,
// MSGCLASS=A,MSGLEVEL=(1,1)
/*ROUTE XEQ NJEVT
/*ROUTE PRINT NJOVT
/*+JBS BIND VT01
/*
```

Dan wordt het HZSPRINT programma opgeroepen. Met de parameters zoals hieronder, roep je alle checks aan van alle owners met een exception. Er worden hier voor wildcards gebruikt(*).

```
//HZSPRINT EXEC PGM=HZSPRNT,
// PARM='CHECK(*,*),EXCEPTIONS'
```

De output van het HZSPRINT programma word in een tijdelijk bestand gestopt als volgt.

```
//SYSOUT DD DISP=(NEW,PASS),DSN=&&OUT,LRECL=256,RECFM=FB,
// DATACLAS=PSN
/*
```

Dan wordt er met het SMTPAPIX programma een mail opgezet via noreply@volvo.com. Deze gebruikt het tijdelijke bestand als input(MEMO statement), die input verzend het dan naar de email adressen waarnaar verwezen word in de job.

```
//SNDMEM EXEC PGM=SMTPAPIX,PARM='MSGVL=4,
// FROM=NOREPLY@VOLVO.COM',
// COND=(0400,EQ,HZSPRINT)
//APIFILE DD *
)SEND
ETITLE VT01: Health Checker exceptions
OPTION FORCE
DEST jonas.braem@supplier.volvo.com
DEST david.westbrand@hcl.com
DEST kevin.somers@hcl.com
DEST bengt.gellingskog@hcl.com
MEMO FILEA
)END
```


Als laatste zal het tijdelijk bestand verwijderd worden met volgende code.

```
//FILEA DD DISP=(OLD,DELETE),DSN=&&OUT
```

De mail zelf zal nu de check output bevatten van alle checks met een exception zoals in bijlage 2.8.2

3.2.2 Jobs schedulen met IWS

Eens de jobs klaar zijn moeten ze gepland worden. In de setup van HCL zullen de jobs elke vrijdag moeten draaien. Dit zal gebeuren met IBM Tivoli Workload Scheduler(IWS). Met deze tool kunnen we jobs automatisch laten uitvoeren met een interval. Je kan deze tool vergelijken met CRON op linux. Om de scheduler te initialiseren wordt volgend commando gebruikt vanuit het ISPF hoofdmenu

```
TSO OPCA
```

Hier is dan een plan opgesteld met de 4 print jobs als volgt. De FHZRS is een 'Routine start' en de FHZSRE een 'Routine exit' om zo het begin en einde van het plan af te bakenen.

```
----- BROWSING OPERATIONS ----- Row 1 to 6 of 6
Command ==> _ Scroll ==> PAGE
Enter the PRED command above to include predecessors in this list, or,
enter the GRAPH command above to view operations graphically.
Enter the row command S to select the details of an operation.
Enter the row command J to browse the JCL

Application      : FHZSPRVT      HC print report VT LPARs

Row  cmd  Oper  Duration  Job name  Operation text
ws  no.    HH.MM.SS
-----
..... CPNT 005  00.00.01  FHZSRS
..... CPUT 025  00.00.01  FHZSVT11
..... CPUT 050  00.00.01  FHZSVT21
..... CPUT 075  00.00.01  FHZSVT31
..... CPUT 100  00.00.01  FHZSVT41
..... CPNT 255  00.00.01  FHZSRE
***** Bottom of data *****
```

Figuur 3.3: Het plan met de 4 JCL print jobs samen met een routine start en routine end.

Dan moet men het interval instellen dat bepaald wanneer het plan uitgevoerd word. Dit moet elke vrijdag gebeuren. Dit stel je in als volgt.

```
Enter the GENDAYS command to display the dates generated by this rule
Enter the E command to specify EVERY options

Application      : FHZSPRVT      HC print report VT LPARs
Rule             : WEEK

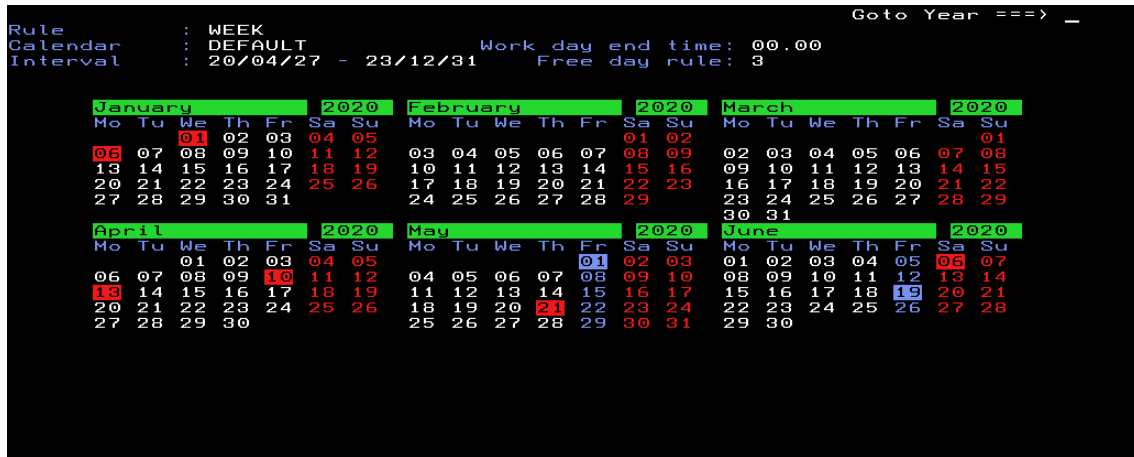
--- Frequency ---      --- Day ---      --- Cycle Specification ---
Only                  - Day -            S Week
S Every               - Free day          - January - July
- First              - Work day           - Month   - August
- Second             - Monday            - Year    - March   - September
- Third              - Tuesday           -         - April   - October
- Fourth             - Wednesday          -         - May    - November
- Fifth              - Thursday           -         - June   - December
                    - Friday
                    - Saturday
                    - Sunday

Week number          _____
Period/RG            _____
name                 _____
Shift default origin by _____ days
```

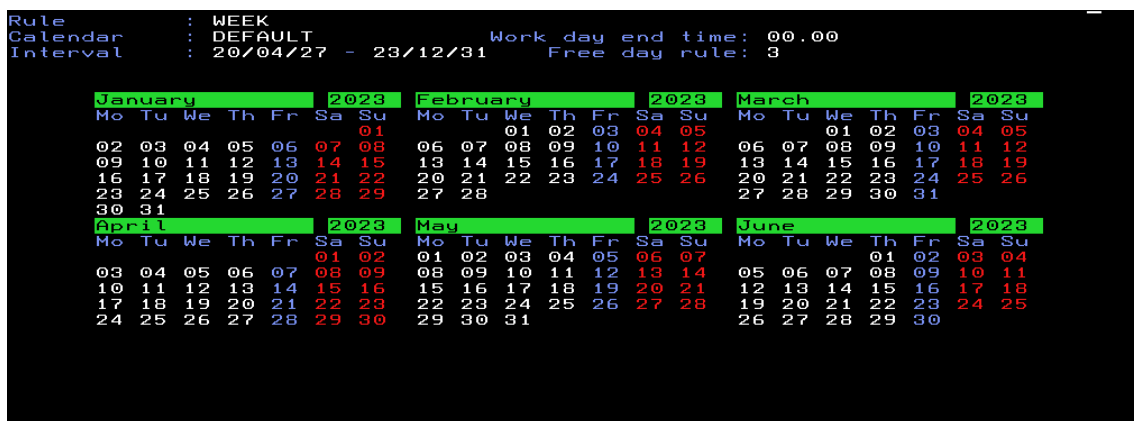
Figuur 3.4: Het interval is ingesteld om het plan elke vrijdag van elke week uit te voeren.

Om na te kijken of het interval goed ingesteld is kunnen we via het GENDAYS commando via het scherm op figuur 3.4 naar een kalender te gaan. Deze kalender duidt alle dagen dat

het plan zal uitvoeren aan in het blauw. En op figuur 3.5 is te zien dat dit alle vrijdagen zijn vanaf 01/05/20. Dit is de datum wanneer het plan voor de eerste keer werd uitgevoerd. Dit plan staat gepland voor de komende 10 jaar zoals te zien is op figuur



Figuur 3.5: De kalender duid datums aan met een blauw lettertype. Op deze dagen zal het plan uitgevoerd worden



Figuur 3.6: Het interval van het plan staat alvast gepland tot 2031

Zoals te zien op de kalenders zijn de JCL jobs nu geprogrammeerd en zal een team elke vrijdag een log ontvangen met alle exceptions van alle LPARs. Het team zal dan de verantwoordelijke van de check met een exception aansporen om deze op te lossen. Door deze werking zal er nu efficiënter gewerkt worden met z/OS Health Checker en is er nog minder kans dat er problemen over het hoofd worden gezien.

4. Vervolg op bachelorproef

4.1 Verwerken van SDSF output

Een interessant vervolg van deze proef is het maken van applicaties/programma's voor het beheer van SDSF. En voornamelijk het beheer van het Health Checker panel. Een vervolg op deze proef zou een vergelijkende studie kunnen zijn tussen Java en REXX en de manier waarop ze met SDSF data kunnen werken.

4.1.1 REXX vs Java

REXX(Restructured Extending Executor) is een programmeertaal ontworpen door IBM. Deze taal wordt ook gebruikt op de mainframe. Maar wat ook belangrijk is is dat deze taal gebruikt kan worden op programma's te schrijven die werken met SDSF. Dit doe je met de REXX met SDSF interface. Hiermee kan je via REXX meerdere panelen aanroepen en bepaalde kolommen tonen. REXX kan zelf gebruikt worden om grafieken te maken van data uit SDSF. Of om jobs uit te voeren. (**Parziale2007**)

Java is nog een andere taal die gebruikt kan worden. Met de SDSF Java API kan je SDSF data verwerken via een java programma. De API laat je toe om de verschillende SDSF panelen en hun bijhorende data op te vragen. Met java kan je ook acties uitvoeren zoals de kolom commando's in SDSF. Hiermee zou je een check mee kunnen starten bijvoorbeeld. (**Wood2012**)

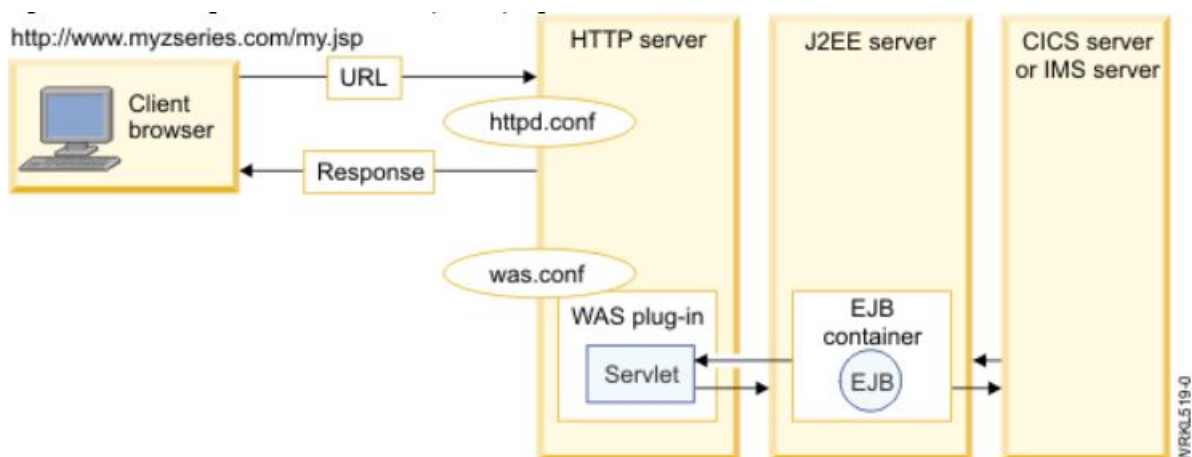
Een vervolg op de proef zou een studie kunnen zijn die onderzoekt welke van de 2 opties het meest geschikt zou zijn voor het verwerken van SDSF Data. En de taal die het meest geschikt zou zijn voor interactie met z/OS Health checker via SDSF

4.2 Web Interface

HCL Technologies hun logging is gewenst via mail. Maar het zou interessant zijn om de logging van z/OS Health Checker uit te breiden naar een WebUI. Daarom worden er in dit hoofdstuk nog methoden bekeken om de SDSF output te loggen op een webpagina. Om webpagina's te hosten op z/OS zijn enkele mogelijkheden. Dit kan zowel voor statische webpagina's maar dynamische webpagina's zijn degene die nodig zijn. Een van de manieren voor een dynamische website te hosten op z/OS is met volgende setup.

De setup bevat een z/OS HTTP web server met een Webpsphere application Server plug-in. In deze plugin zit een java servlet dit is een java programma dan interactie heeft met de EJB. Dit staat voor Enterprise JavaBeans dit is een server side software component. Deze draait dan op een J2EE server. (IBM, 2010)

Deze server biedt een platform aan voor server side applicaties. En aangezien interactie mogelijk is tussen Java en SDSF met de SDSF Java API. Zou je dus zo de SDSF output op een webpagina krijgen. En zo een logging systeem maken met een webUI. (Vilaghy2002)



Figuur 4.1: Visualisatie van de omgeving nodig voor het draaien van dynamische webpagina's op z/OS(IBM, 2010)

Een vervolg van deze proef kan dus een studie zijn naar het opzetten van deze architectuur en de mogelijkheden die ze biedt voor het werken met SDSF via een webUI.

5. Conclusie

De eerste onderzoeksvraag was of het mogelijk is om een standaardopstelling te maken voor z/OS Health Checker. Deze standaard setup is bereikt en wordt nu gebruikt binnen de mainframe omgeving van HCL technologies via de parmlib members in bijlage B.3 die daar nu geïmplementeerd zijn. Als HCL Technologies wenst om hun parallel sysplex uit te breiden, kunnen ze de nieuwe parmlib members gebruiken om z/OS Health Checker te configureren voor de nieuwe LPAR. Een van de requirements voor HCL Technologies was een centraal beheer van de tool op 1 LPAR, dit was VT1. Nadat de globale checks eerst verspreid draaiden op alle LPARs, zullen ze nu allemaal vanuit VT1 draaien. En is er een centraal overzicht en beheer van deze checks via SDSF op VT1.

Een andere requirement was een logging systeem zodat men de exceptions van bepaalde checks direct kan oplossen. Nadat het systeem met JCL jobs & IBM Tivoli Workload Scheduler is opgezet, heeft men de verantwoordelijkheid voor het opvolgen van deze logging aan een team in India doorgegeven. Deze zal nu bij exceptions de verantwoordelijke van het product aansporen om het probleem op te lossen. Zij beheren nu ook de standaardopstelling zoals gedefinieerd in bijlage B.2. Als er nieuwe checks zijn zullen zij deze ook toevoegen aan deze opstelling.

Voor dit onderzoek was ik niet zeker of de standaardopstelling bereikt ging worden aangezien ik nog niks van z/OS Health Checker wist. Maar na een tijd werd duidelijk dat dit zeker ging lukken. Voor het logging systeem had ik wel al verwacht dat deze ging lukken aangezien er al documentatie beschikbaar was hoe je deze kan doen met HZSPRINT.

Dit onderzoek biedt zeker een meerwaarde voor bedrijven zoals HCL Technologies. Na deze proef is er een algemeen model opgesteld van hun opstelling voor Health Checker.

Men zal met dit model problemen sneller kunnen opsporen en oplossen alvorens deze plaatsvinden. En door de opstelling zullen zij ook makkelijker hun Parallel Sysplex kunnen uitbreiden. De opstelling in de bijlage is dan ook in bezit van HCL Technologies en deze wordt nu ook gebruikt in hun mainframe testomgeving.

Verder studie zou het loggen kunnen optimaliseren door dit te doen via een web interface. Een andere optie is een vergelijkende studie tussen REXX en Java. Die studie kan dan een analyse maken welke van de 2 de meeste opties aanbied voor interactie met SDSF. Deze vervolgen worden uitgebreid besproken in hoofdstuk 4

A. Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

A.1 Introductie

Mainframes staan bekend om hun continue toegankelijkheid doorheen hun levensduur. Een van de tools die de toegankelijkheid garandeert is z/OS Health Checker. In dit onderzoek gaat men de werking van z/OS Health checker onderzoeken en volgens met methodes het de mainframe weerhoudt op een eventuele uitval. Daarna wordt de huidige opstelling van deze Tool geanalyseerd binnen de mainframe infrastructuur van HCL. Om deze daarna dan te verbeteren en efficiënter te maken.

A.2 State-of-the-art

De mainframe is en blijft nog altijd een belangrijk systeem binnen onze samenleving. Er wordt vaak gesuggereerd dat de mainframe iets is uit het verleden, maar vele instellingen (bv. Banken) vertrouwen er nog op. Dit is omdat de mainframe garandeert 24 op 24, 365 op 365 operationeel te zijn. Maar voor die garantie heeft het bepaalde tools. Een van deze tools is z/OS Health Checker. Deze tool is geen diagnostische of toezichhoudende tool. Maar ze is eerder een continu lopend preventie tool. Dat potentiële problemen voor de Mainframe probeert te zoeken en te melden aan de systeem administrator. De tool geeft niet enkel het probleem maar ook een aanbevolen actie die de systeem administrator kan nemen.

A.3 Methodologie

Eerst zal er een studie zijn naar de tool op deze te begrijpen en correct te hanteren. Dan pas zullen de opstellingen van z/OS Health Checker op alle systemen van HCL geanalyseerd worden. Met die analyse zal er een vergelijking opgesteld worden om te kunnen onderscheiden op welke systemen de opstelling goed is en op welke deze nog efficiënter kan. Van hieruit wordt er een proof of concept uitgewerkt. Die proberen we dan te implementeren op alle systemen van toepassing.

A.4 Verwachte resultaten

Een duidelijker beeld van hoe een Mainframe zijn toegankelijkheid garandeert met een tool zoals z/OS Health checker. Een andere verwachting is ook een beter begrip van de werking van de z/OS Health Checker tool en hoe deze de mainframe de garantie geeft om 365 op 365 te blijven draaien zonder problemen. Verder zal men ook het belang van een tool zoals deze kunnen begrijpen.

A.5 Verwachte conclusies

Dat z/OS health checker een belangrijk onderdeel is van het mainframe systeem. Dat men na de analyse en vergelijking van de huidige opstelling een betere en efficiëntere heeft ontwikkeld. En dat deze ook geïmplementeerd is en zo de systemen nog betrouwbaarder zijn dan voordien.

A.6 Literatuurstudie

A.6.1 Inleiding

De mainframe is een computer dat bekend staat voor zijn continue toegankelijkheid. Door deze availability wordt de mainframe soms nog verkozen boven de gewone servers. Een mainframe is ontworpen om heel lang operationeel te blijven zonder onderbreking. Dit gebeurt natuurlijk niet zomaar. Een van de tools dat IBM hiervoor heeft gemaakt is Health Checker. Een preventietool die probeert problemen op te sporen voordat ze zich voordoen. Maar deze heeft geen efficiënte manier om de problemen te loggen. Dit is ook bij de opdrachtgever HCL Technologies. Deze heeft een setup die al jaren niet meer veranderd is. En ook niet uniform is op de verschillende Logische partities.

De bedoeling van deze proef is:

- Nieuw log systeem van eventuele problemen opgemerkt door Health Checker
- Gecentraliseerd beheer van de tool & checks op 1 LPAR

- Aanpassen van checks naar wensen van HCL

A.6.2 Mainframe

De mainframe is een type computer gespecialiseerd in het afhandelen van input & output requests ook gekend als transacties. Een mainframe is ook veel groter dan een gewone computer en heeft zo ook veel meer processing vermogen. De eerste mainframe is ontworpen in de jaren 50 en tot nu toe blijft ze een belangrijke verwerkers van workloads terwijl ze nog steeds applicaties draaien gemaakt in 1970. De term mainframe begint door nieuwe technologieën te vervagen. Tegenwoordig kan men volgens IBM (Ebbers e.a., 2011) beter beschrijven als een systeem dat bedrijven gebruiken voor het beheren van commerciële databanken, transactie servers en applicaties die een hogere graad van veiligheid en toegankelijkheid nodig hebben in tegenstelling tot die gevonden in kleiner geschaalde machines.

De mainframe ziet zijn gebruik vooral in volgende sectors: banken, financiën, zieken zorg, verzekeringen overheden en veel meer. De mainframe blijft een fundamenteel onderdeel van deze sectoren.

Men mag een mainframe ook niet verwarren met een supercomputer. Een supercomputer is namelijk eerder gespecialiseerd in het uitvoeren van calculaties en niet in het afhandelen van transacties in tegenstelling tot de mainframe.

A.6.3 z/OS

Is het besturingssysteem voor de IBM mainframe. z/OS is het resultaat van tientallen jaren van ontwikkeling dat is begonnen bij het eerste besturingssysteem voor IBM mainframes. Het begon als een systeem dat maar 1 programma tegelijk kon draaien naar een dat er duizenden tegelijk aankan. z/OS bied volgens IBM (Ebbers e.a., 2011) een stabiele veilige schaalbare en continu toegankelijke omgeving.

A.6.4 LPAR of Logische Partitie

LPARs of logische partities zijn eigenlijk op zichzelf aparte mainframes die hun eigen besturingssysteem hebben. Dit hoeft niet z/OS te zijn maar kan ook z/VM zijn of Linux on Z. Volgens IBM kan het er zo'n 60 hebben op 1 fysieke mainframe Ebbers e.a., 2011. Elke LPAR heeft zijn eigen toegewezen aantal processors & hoeveelheid geheugen. LPARS kunnen wel met elkaar communiceren via de SYSPLEX. Een architectuur van verscheiden LPARS die met elkaar communiceren wordt een parallel sysplex genoemd. Op elke LPAR zal er een instantie van Health Checker aanwezig zijn.

A.6.5 z/OS Health checker

Health Checker is een preventieve tool. Die standaard in elke IBM mainframe meegegeven word elke nieuwe versie van z/OS. De tool is ontworpen doordat de meest voorkomende reden van een uitval een single point of failure was in de instellingen van het systeem volgens IBM Bezzi e.a., 2010 of door performance bottlenecks Walle, 2013. Ze wordt dus gebruikt om single points of failures te zoeken in slecht opgestelde instellingen van een systeem. De tool doet dat door de huidige systeeminstellingen te vergelijken met instellingen die door de eigenaar van een Check worden gedefinieerd of door zelf gedefinieerde waarden. Nogmaals moet de nadruk gelegd worden op dat health checker een preventieve tool is, dit betekent dat het dus zelfs niks zal doen buiten het rapporteren van problemen. De tool geeft bij de rapportering van een probleem wel mee welke acties de verscheidene administrators van de mainframe moeten ondernemen. Het oplossen van de problemen moet dus gebeuren door de administrators. Health Checker bestaat uit 2 delen die samen een geheel vormen:

- Health Checker Framework
- Checks

Health Checker Framework

Dit is de interface die de checks beheert onder andere hun logging, het plannen van de uitvoering van checks. Ze ondersteunt checks gemaakt door IBM maar ook checks door externe bedrijven of checks die je zelf schrijft.

Checks

De check zelf is een programma of routine die een bepaald component of instelling evalueert om zo potentiële problemen te zoeken op een systeem. Ze zijn op zichzelf onafhankelijk van het framework.

B. Bijlagen

B.1 Check Output

Check output bij exception van de XCF_CF_STR_POLICIYSIZE check.

```
***** TOP OF DATA *****
CHECK(IBMxcf,xcf\_cf\_str\_policiysize)
SYSPLEX:    PLEXVT    SYSTEM: VT01
START TIME: 05/12/2020 09:42:52.954735
CHECK DATE: 20090707  CHECK SEVERITY: MEDIUM
```

IXCH0923I Coupling facility structure policy sizing is summarized by the following report:

An asterisk (*) before a structure name indicates an exception condition. When the qualification is "Alter not allowed", an exception condition is when INITSIZE is specified not equal to SIZE. Otherwise an exception condition is when INITSIZE is less than half of SIZE.

Structure Name	INITSIZE	Max SIZE	Alter Qualification
*ISGLOCK	20 M	30 M	Alter not allowed
*MQTOAPPL1	256 M	1 G	Alter supported
*MQTOAPPL3	256 M	1 G	Alter supported
*MQTOAPPL4	256 M	1 G	No connections defined

*MQTOAPPL5 256 M 1 G Alter supported

* Medium Severity Exception *

IXCH0255E A CFRM policy structure specification has too large a difference between the INITSIZE and SIZE values.

Explanation: A specification of INITSIZE in the active or pending CFRM policy indicates an initial structure size that is too small for the maximum structure size (as determined by the SIZE specification). Either a structure has an initial size specified as less than half the maximum size, or a structure whose users do not allow structure alter has an initial size specified different than the maximum size.

When allocating the structure initially, whether INITSIZE is specified or not, the system attempts to build all control structures that will be required to support the maximum size of the structure. These control structures are built in the control storage allocation of the structure. For structures whose users do not allow structure alter, the control storage allocated to accommodate larger sizes is wasted. An INITSIZE value substantially smaller than the SIZE value might cause the following:

- It might be impossible to allocate a structure at a size of INITSIZE, because the amount of control storage required to support the SIZE value might actually be larger than INITSIZE.
- If the allocation succeeds, it might result in a structure with a proportionally large amount of its storage allotted to structure controls, leaving too few structure objects to be exploited usefully by the associated application.

System Action: The system continues processing.

Operator Response: N/A

System Programmer Response: IBM suggests that the INITSIZE and SIZE specification for structures be determined by the CfSizer (Coupling Facility Structure Sizer) tool:

<http://www.ibm.com/systems/z/cfsizer>

Use the CfSizer tool to determine the INITSIZE and SIZE parameters for structures with an exception condition. Update the CFRM policy (or policies) with the new parameters. The new parameters should not

have an INITSIZE value for a structure less than half the SIZE value for that structure. If alter is not allowed by users of a structure, INITSIZE should not be specified for that structure. Start an updated policy with the following system command:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=polname
```

The policy changes will become pending and affect only future structure allocations, not currently allocated structures. REBUILD or REALLOCATE can be used to activate the pending changes for currently allocated structures. For example, a REALLOCATE process can be started with the following system command:

```
SETXCF START,REALLOCATE
```

Problem Determination: See IXCH0923I in the message buffer that identifies the coupling facility structures with an exception condition.

The problem may have occurred because a CFRM policy structure SIZE value was adjusted without also adjusting the INITSIZE value (or vice versa).

The following system command can be used to determine the name of the active or pending CFRM policy:

```
DISPLAY XCF,POLICY,TYPE=CFRM
```

Source: Parallel Sysplex (XCF)

Reference Documentation: For more information on planning and activating CFRM policies, see z/OS MVS Setting Up a Sysplex.

For the syntax of the SETXCF START command, see "SETXCF Start Command" in z/OS MVS System Commands.

Automation: N/A

Check Reason: Too large a difference between INITSIZE and SIZE may waste coupling facility space or prevent structure allocation.

END TIME: 05/12/2020 09:42:53.031335 STATUS: EXCEPTION-MED

***** BOTTOM OF DATA *****

B.2 Tabellen Standaard opstelling z/OS Health Checker

Dit zijn de tabellen van alle checks gegroepeerd per team dat beheer heeft over de checks.

In deze bijlage bevinden zich de tabellen met de standaardopstelling van de checks van volgende teams in deze volgorde

1. CICS
2. Communication
3. Print
4. Automation
5. Rollout & Operate
6. Runtime Control
7. Security
8. SOE
9. Storage
10. zOPEN
11. DB

De tabellen staan horizontaal omdat ze verticaal niet op een pagina passen.

Name	Status	Outcome	Reason	Run	00/00/SUF.
CICS_CEDA_ACCES	INACTIVE(ENABLED)	INACT	CEDA can be used by unauthenticated users	No	00
CICS_JOBSUB_SPOOL	INACTIVE(ENABLED)	INACT	Jobs can be run with regionid authority by unauthenticated users using the SPOOL	No	00
CICS_JOBSUB_TDQINTRDR	INACTIVE(ENABLED)	INACT	Jobs can be run with regionid authority by unauthenticated users using a TDQ	No	00

Tabel B.1: Setup van Health Checker voor het CICS team

Name	Status	Outcome	Reason	Run	00/&SUF.
CSAPP_FTPD_ANONYMOUS_JES	ACTIVE(ENABLED)	SUCCESS	CHECK FTPD ANONYMOUS JES STATEMENT IS IN USE	Yes	N/A
CSAPP_FTPDMVRSHD_RHOSTS_DATA	ACTIVE(ENABLED)	SUCCESS	CHECK MVRSHD RHOSTS DATA IS IN USE	Yes	N/A
CSAPP_SNMPAGENT_PUBLIC_COMMUNITY	ACTIVE(ENABLED)	SUCCESS	CHECK SNMP AGENT PUBLIC COMMUNITY IS IN USE	Yes	N/A
CSRES_AUTOQ_GLOBALTCPIPDATE	ACTIVE(ENABLED)	SUCCESS	CHECK THAT GLOBALTCPIPDATE IS NOT SPECIFIED WHEN AUTOQUIESCENCE IS SPECIFIED	Yes	N/A
CSRES_AUTOQ_RESOLVEVIA	ACTIVE(ENABLED)	SUCCESS	CHECK RESOLVEVIA VALUE WHEN THE AUTONOMIC QUIESCING OF UNRESPONSIVE NAME SERVERS FUNCTION IS ACTIVE	Yes	N/A
CSRES_AUTOQ_TIMEOUT	ACTIVE(ENABLED)	SUCCESS	CHECK RESOLVEVIA VALUE WHEN THE AUTONOUTONOMIC QUIESCING OF UNRESPONSIVE NAME SERVERS FUNCTION IS ACTIVE	Yes	N/A

Tabel B.2: Setup van Health Checker voor het Communication team tabel 1

Name	Status	Outcome	Reason	Run	00/&SUF.
CSTCP-IPMAXRT4_TCPIP*	ACTIVE(ENABLED)	SUCCESS	CHECK FOR TC- P/IP IPV4 INDIRECT ROUTES MAXIMUM THRESHOLD	Yes	N/A
CSTCP-IPMAXRT6_TCPIP*	ACTIVE(ENABLED)	SUCCESS	CHECK FOR TC- P/IP IPV6 INDIRECT ROUTES MAXIMUM THRESHOLD	Yes	N/A
CSTCP_IWQ_IPSEC_TCPIP*	ACTIVE(ENABLED)	SUCCESS	ENSURE SUFFICIENT FIXED STORAGE IS AVAILABLE FOR IWQ IPSEC	Yes	N/A
CSTCP_SYSPLEXMON_RECOV_TCPIP*	ACTIVE(ENABLED)	SUCCESS	CHECK THAT SYSPLEX- MONITOR RECOVERY IS SPECIFIED WHEN DY- NAMICXCF IS SPECI- FIED	Yes	N/A
CSTCP_SYSTCPIP_CTRACE_TCPIP*	ACTIVE(ENABLED)	SUCCESS	CHECK FOR TCP/IP SYSTCPIP CTRACE WITH NONDEFAULT OPTION	Yes	N/A
CSTCP_TCPMAXRCVBUFSIZE_TCPIP*	ACTIVE(ENABLED)	SUCCESS	ENSURE TCP RECEIVE BUFFER SIZE IS SUFFI- CIENT FOR FTP SER- VER	Yes	N/A
CSVMTAM_CSM_STG_LIMIT	ACTIVE(ENABLED)	SUCCESS	CHECK CSM STORAGE LIMIT VTV0022 REXX?	Yes	N/A
CSVMTAM_T1BUF_T2BUF_EE	ACTIVE(ENABLED)	SUCCESS	CHECK T1BUF/T2BUF ALLOCATIONS WITH EE	Yes	N/A

Tabel B.3: Setup van Health Checker voor het Communication team tabel 2

Name	Status	Outcome	Reason	Run	00/&SUF.
CSV TAM_T1BUF_T2BUF_NOEE	ACTIVE(DISABLED)	ENV N	CHECK T1BUF/T2BUF ALLOCATIONS WIT-HOUT EE	Yes	N/A
CSV TAM_VIT_OPT_ALL	ACTIVE(ENABLED)	SUCCESS	CHECK VIT OPT=ALL IS NOT SPECIFIED	Yes	N/A
CSV TAM_VIT_OPT_STDOPTS	ACTIVE(ENABLED)	SUCCESS	CHECK VIT STDOPTS OPTION IS ACTIVE	Yes	N/A
TPX_ACB_INACTIVE_@TPX*	ACTIVE(ENABLED)	SUCCESS	TPX ACB is inactive	Yes	N/A
TPX_CF_UNCONNECTED_@TPX*	ACTIVE(ENABLED)	SUCCESS	TPX coupling facility is not connected	Yes	N/A
TPX_STORAGE_OVERFLOW_@TPX*	ACTIVE(ENABLED)	SUCCESS	TPX storage overflow beyond boundaries	Yes	N/A
CSTCP_CINENT_PORTRNG_RSV_TCPIP*	INACTIVE(ENABLED)	INACT	CHECK THAT CINET INADDRANYPORT RANGE IS RESERVED FOR OMVS ON THE TCP/IP STACK	No	00

Tabel B.4: Setup van Health Checker voor het Communication team tabel 3

Name	Status	Outcome	Reason	Run	00/&SUF.
DLVR_CKPT_USAGE@RMOVT1	ACTIVE(ENABLED)	SUCCE	Monitor CA Deliver checkpoint utilization.	Yes	N/A
DLVR_OPT_HDETAIL@RMOVT1	ACTIVE(ENABLED)	SUCCE	Insure the HDETAIL option is only activated when necessary	Yes	N/A
DLVR_PREFM_PQE@RMOVT1	ACTIVE(ENABLED)	SUCCE	Monitor CA Deliver application work queues.	Yes	N/A
SPOOL_CKPT_ACT@ESF	ACTIVE(ENABLED)	SUCCE	To check CA Spool checkpoint working	Yes	N/A
SPOOL_FILE_QUEUE_PCT@ESF	ACTIVE(ENABLED)	SUCCE	To check CA Spool file queue utilization	Yes	N/A
SPOOL_SPACE_PCT@ESF	ACTIVE(ENABLED)	SUCCE	To check CA Spool data set space utilization	Yes	N/A
SPOOL_TCP_ACT@ESF	ACTIVE(ENABLED)	SUCCE	To check CA Spool TCP/IP printer subtasks	Yes	N/A
SPOOL_TRANSFRM_ACT@ESF	ACTIVE(ENABLED)	SUCCE	To check CA Spool Transfrmr subtasks	Yes	N/A
VIEW_OPT_CLFMDST@SAR*	ACTIVE(ENABLED)	SUCCE	Alert datacenter staff that CA View is configured to archive every report in the spool	Yes	N/A
VIEW_OPT_MASTER@SAR*	ACTIVE(ENABLED)	SUCCE	Insure that all CA View users do not have access to ADMINISTRATIVE functions	Yes	N/A
VIEW_OPT_TBACKUP@SAR*	ACTIVE(ENABLED)	SUCCE	Insure that TBACKUP is set to a value that allows the database to be restored	Yes	N/A

Tabel B.5: Setup van Health Checker voor het Print team

Name	Status	Outcome	Reason	Run	00/&SUF.
GDPS_CHECK_CONFIG	ACTIVE(ENABLED)	EXCEPTION	GDPS check verifies configurations	Yes	N/A
GDPS_CHECK_CONSOLE	ACTIVE(ENABLED)	SUCCESS	GDPS check CONSOLE information	Yes	N/A
GDPS_CHECK_DASDMIH	ACTIVE(ENABLED)	SUCCESS	GDPS check DASD MIH	Yes	N/A
GDPS_CHECK_DEVICE	ACTIVE(ENABLED)	EXCEPTION	GDPS check verifies device configuration	Yes	N/A
GDPS_CHECK_GRS	ACTIVE(DISABLED)	ENV N	GDPS check verifies GRS definitions	Yes	N/A
GDPS_CHECK_JOBS	ACTIVE(ENABLED)	SUCCESS	GDPS check JOBS	Yes	N/A
GDPS_CHECK_K_SYS_LPAR	ACTIVE(DISABLED)	ENV N	GDPS check K SYS LPAR informationx	Yes	N/A
GDPS_CHECK_LOGR	ACTIVE(ENABLED)	SUCCESS	GDPS check verifies LOGR configuration	Yes	N/A
GDPS_CHECK_MAXSYS	ACTIVE(ENABLED)	SUCCESS	GDPS check MAXSYS information	Yes	N/A
GDPS_CHECK_NUMUCBS	ACTIVE(ENABLED)	SUCCESS	GDPS max devices generated	Yes	N/A
GDPS_CHECK_SPOF	ACTIVE(ENABLED)	EXCEPTION	GDPS check connectivity Single Point Of Failure	No	00
GDPS_CHECK_STATE	ACTIVE(ENABLED)	SUCCESS	GDPS check STATE	Yes	N/A
GDPS_CHECK_XCF	ACTIVE(ENABLED)	SUCCESS	GDPS check verifies XCF definitions	Yes	N/A
GDPS_CHECK_XCF_CDS	ACTIVE(ENABLED)	SUCCESS	GDPS check verify XCF CDS datasets	Yes	N/A

Tabel B.6: Setup van Health Checker voor het Automation team

Name	Status	Outcome	Reason	Run	00/&SUF.
CSS_ENF_SCREEN_VALIDITY	ACTIVE (ENABLED)	SUCCESS	ENF screen command check	Yes	N/A
CNZ_ARMF_EVENTUAL_ACTION_MSGS	ACTIVE (ENABLED)	SUCCESS	Avoid the loss of critical action messages	Yes	N/A
CNZ_CONSOLE_MASTERAUTH_CMDSYS	ACTIVE (ENABLED)	SUCCESS	Ability to control this system	Yes	N/A
CNZ_CONSOLE_MSCOPE_AND_ROUTE_CODE	ACTIVE (ENABLED)	EXCEPTION	Reduces the number of messages sent to a console in the sysplex	No	00
CNZ_CONSOLE_ROUTE_CODE_11	ACTIVE (ENABLED)	SUCCESS	To reduce message traffic by removing messages that are only intended for the problem programmer	Yes	N/A
CNZ_EMCS_HARDCOPY_MSCOPE	ACTIVE (ENABLED)	EXCEPTION	To avoid having EMCS consoles process an excessive number of messages	No	00
CNZ_EMCS_INACTIVE_CONSOLES	ACTIVE (DISABLED)	SUCCESS	Reduces the time a system takes to join a sysplex	Yes	&SUF.
CNZ_OBSOLETE_MSGFLD_AUTOMATION	ACTIVE (ENABLED)	SUCCESS	Obsolete versions of Message Flood Automation must be removed from the system	Yes	N/A
CNZ_SYSCONS_ALLOWCMD	INACTIVE (ENABLED)	EXCEPTION	To ensure that the system console is as available as possible to act as a console of last resort	No	N/A

Tabel B.7: Setup van Health Checker voor het ROO team tabel 1

Name	Status	Outcome	Reason	Run	00/&SUF.
CNZ_SYSCONS_MSCOPE	ACTIVE (ENABLED)	SUCCESS	To ensure the system console is configured to receive messages from only the local system	Yes	N/A
CNZ_SYSCONS_PD_MODE	ACTIVE (ENABLED)	SUCCESS	Should only run in Problem Determination mode when there is a problem	Yes	N/A
CNZ_SYSCONS_ROUTECD	ACTIVE (ENABLED)	SUCCESS	To ensure the system console is configured to receive important messages	Yes	N/A
CNZ_TASK_TABLE	ACTIVE (ENABLED)	SUCCESS	Check Consoles Task Table	Yes	N/A
CTRACE_DEFAULT_OR_MIN	ACTIVE (ENABLED)	SUCCESS	A component trace that is tracing more than the degraded fault can cause degraded system performance.	Yes	N/A
DAE_SHAREDSDN	ACTIVE (ENABLED)	SUCCESS	To ensure DAE data set is shared between systems in sysplex	Yes	N/A
DAE_SUPPRESSING	ACTIVE (ENABLED)	SUCCESS	To ensure DAE subsystem is active and suppressing dumps	Yes	N/A
ISPF_WSA	INACTIVE (ENABLED)	INACT/SUCCESS	CHECK IF ISPF WORK-STATION AGENT IS IN USE	No	N/A

Tabel B.8: Setup van Health Checker voor het ROO team tabel 2

Name	Status	Outcome	Reason	Run	00/&SUF.
IXGLOGR_ENTRYTHRESHOLD	ACTIVE (ENABLED)	SUCCESS	Logger entry threshold reached conditions should be investigated to determine if applications performance is being impacted	Yes	N/A
IXGLOGR_STAGINGDSFULL	ACTIVE (ENABLED)	SUCCESS	Logger staging dataset full conditions should be investigated to determine if application performance is being impacted	Yes	N/A
IXGLOGR_STRUCTUREFULL	ACTIVE (ENABLED)	SUCCESS	Logger structure full conditions should be investigated to determine if application performance is being impacted	Yes	N/A
JES_NJE_SECURITY	ACTIVE (ENABLED)	UNEXPECTED ERROR	Verify primary JES NJE security settings	YES	N/A
JES2_UPGRADE_CKPT_LEVEL_JES2	ACTIVE (ENABLED)	SUCCESS	JES2 checkpoint level upgrade checks	Yes	N/A
RRS_ARCHIVECFSTRUCTURE	ACTIVE (ENABLED)	SUCCESS	RRS Archive log stream is sharing its coupling facility structure with another log stream. This is not recommended.	Yes	N/A
RRS_DELAYED_TRANSACTIONS	INACTIVE (ENABLED)	INACT	The following list of RRS Transactions are active in SyncPoint but have not yet completed in the allotted amount of time.	Yes	N/A

Tabel B.9: Setup van Health Checker voor het ROO team tabel 3

Name	Status	Outcome	Reason	Run	00/&SUF.
RRS_DUROFFLOADSIZE	ACTIVE (ENABLED)	SUCCESS	Delayed UR log offload dataset size should be at least as large as the space allocated for the log stream in the structure.	Yes	N/A
RRS_MUROFFLOADSIZE	ACTIVE (ENABLED)	SUCCESS	Main UR log offload dataset size should be at least as large as the space allocated for the log stream in the structure.	Yes	N/A
RRS_RMDATALOGDUPLEXMODE	ACTIVE (ENABLED)	SUCCESS	RM Data log should use a better duplexing scheme than local buffer duplexing.	Yes	N/A
RRS_RMDOFFLOADSIZE	ACTIVE (ENABLED)	SUCCESS	RM Data log offload dataset size should be at least as large as the space allocated for the log stream in the structure.	Yes	N/A
RRS_RSTOFFLOADSIZE	ACTIVE (ENABLED)	SUCCESS	Restart log offload dataset size should be at least as large as the space allocated for the log stream in the structure.	Yes	N/A
RRS_STORAGE_NUMLARGELOGBLKS	ACTIVE (ENABLED)	SUCCESS	If the count of large log buffers in RRS grows too big then RRS might encounter a terminating failure.	Yes	N/A

Tabel B.10: Setup van Health Checker voor het ROO team tabel 4

Name	Status	Outcome	Reason	Run	00/&SUF.
RRS_STORAGE_NUMLARGMSGBLKS	ACTIVE (ENABLED)	SUCCESS	If the count of large message blocks in RRS grows too big then RRS might encounter a terminating failure.	Yes	N/A
RRS_STORAGE_NUMSERVERREQS	ACTIVE (ENABLED)	SUCCESS	If the count of server requests within RRS grows too big then RRS might be encountering a hang situation.	Yes	N/A
RRS_STORAGE_NUMTRANSBLKS	ACTIVE (ENABLED)	SUCCESS	If the count of transactions that RRS is managing grows too big then RRS might encounter a terminating failure.	Yes	N/A
RTM_IEAVTRML	ACTIVE (ENABLED)	SUCCESS	Placing resource manager names into IEAVTRML can hurt system performance.	Yes	N/A
SDSF_CLASS_SDSF_ACTIVE	ACTIVE (ENABLED)	SUCCESS	Verify SAF class SDSF is active	Yes	N/A
SDSF_ISFPARMS_IN_USE	ACTIVE (ENABLED)	SUCCESS	Check source of SDSF configuration parameters	Yes	N/A
SDUMP_AUTO_ALLOCATION	ACTIVE (ENABLED)	SUCCESS	Automatic allocation of dump data sets efficiently writes the dump from virtual storage to DASD.	Yes	N/A

Tabel B.11: Setup van Health Checker voor het ROO team tabel 5

Name	Status	Outcome	Reason	Run	00/&SUF.
SDUMP_AVAILABLE	ACTIVE (ENABLED)	SUCCESS	The SDUMP setup should ensure a dump can be generated when system problems occur.	Yes	N/A
SLIP_PER	ACTIVE (ENABLED)	SUCCESS	An active, but not needed, SLIP PER trap can cause degraded system performance.	Yes	N/A
SVA_AUTOIPL_DEFINED	ACTIVE (DISABLED)	ENV N	To ensure AutoIPL function is in use when available.	Yes	N/A
SVA_AUTOIPL_DEV_VALIDATION	ACTIVE (DISABLED)	ENV N	To ensure IPL device(s) defined in AutoIPL policy are valid.	Yes	N/A
SYSTRACE_BRANCH	ACTIVE (ENABLED)	SUCCESS	An active, but not needed, SYSTRC BR=ON option can cause degraded system performance.	Yes	N/A
SYSTRACE_MODE	ACTIVE (ENABLED)	SUCCESS	An active, but not needed, SYSTRC MODE=ON option can cause degraded system performance.	Yes	N/A
TTOE_OPERSEWAIT_SETTING	ACTIVE (ENABLED)	SUCCESS	Verify OPERSEWAIT is the preferred setting.	Yes	N/A
TTOE_PARM LIB_ERROR	ACTIVE (ENABLED)	SUCCESS	PARMLIB errors may have occurred during IPL.	Yes	N/A
TTOE_USERLOGS	ACTIVE (ENABLED)	SUCCESS	User logs should be in use.	Yes	N/A

Tabel B.12: Setup van Health Checker voor het ROO team tabel 6

Name	Status	Outcome	Reason	Run	00/&SUF.
XCF_CDS_MAXSYSTEM	ACTIVE (ENABLED)	SUCCESS	CDS MAXSYSTEM value across all CDS types should be at least equal to the value in the primary simplex CDS.	Yes	&SUF
XCF_CDS_SEPARATION	ACTIVE (ENABLED)	SUCCESS	Ensure that CDS separation has been maintained.	Yes	N/A
XCF_CDS_SPOF	INACTIVE (ENABLED)	INACT	Ensure that couple data sets are configured without single points of failure.	No	00
XCF_CF_ALLOCATION_PERMITTED	ACTIVE (DISABLED)	SUCCESS	Coupling facilities should have allocation permitted.	Yes	&SUF
XCF_CF_CONNECTIVITY	ACTIVE (ENABLED)	SUCCESS	Avoid problems with CF Connectivity.	Yes	N/A
XCF_CF_MEMORY_UTILIZATION	ACTIVE (ENABLED)	SUCCESS	Coupling facility memory should allow for expansion and failover.	Yes	N/A
XCF_CF_PROCESSORS	ACTIVE (ENABLED)	SUCCESS	Coupling facility performance as measured in service time is degraded with shared coupling facility central processors.	Yes	N/A

Tabel B.13: Setup van Health Checker voor het ROO team tabel 7

Name	Status	Outcome	Reason	Run	00/&SUF.
XCF_CF_SCM_UTILIZATION	ACTIVE (DISAB- LED)	ENV N	Coupling facility storage-class memory should be available to provide relief for coupling facility real storage constraints.	Yes	N/A
XCF_CF_STR_AVAILABILITY	ACTIVE (DISAB- LED)	EXCEPTION	Each structure preference list definition should have two usable coupling facilities which are in different CECs.	Yes	&SUF
XCF_CF_STR_DUPLEX	ACTIVE (DISAB- LED)	SUCCESS	Allocated structures with active CFRM policy values of DUPLEX(ALLOWED) or (ENABLED) should be duplexed in the preferred mode.	Yes	&SUF
XCF_CF_STR_ENCRYPT	ACTIVE (DISAB- LED)	SUCCESS	Structure data is consistent with the ENCRYPT parameter and cryptographic encryption key for the structure in the CFRM policy.	Yes	&SUF
XCF_CF_STR_EXCLLIST	ACTIVE (DISAB- LED)	SUCCESS	Check Structure exclusion lists.	Yes	&SUF

Tabel B.14: Setup van Health Checker voor het ROO team tabel 8

Name	Status	Outcome	Reason	Run	00/&SUF.
XCF_CF_STR_MAXCSM	ACTIVE (DISAB- LED)	ENV N	Storage-class memory (SCM) eligible for use by structures should not exceed the total SCM available to the CF	Yes	N/A
XCF_CF_STR_MAXSPACE	ACTIVE (ENABLED)	SUCCESS	CF real storage eligible for use by structures should not exceed the total space available to the CF	Yes	N/A
XCF_CF_STR_NONVOLATILE	ACTIVE (DISAB- LED)	EXCEPTION	Coupling facility structure non-volatility and failure isolation from connectors should be provided when requested.	Yes	&SUF
XCF_CF_STR_POLICYSIZE	ACTIVE (DISAB- LED)	EXCEPTION	Too large a difference between INITSIZE and SIZE may waste coupling facility space or prevent structure allocation.	Yes	&SUF
XCF_CF_STR_PREFLIST	ACTIVE (DISAB- LED)	SUCCESS	Check Structure preference lists.	Yes	&SUF

Tabel B.15: Setup van Health Checker voor het ROO team tabel 9

Name	Status	Outcome	Reason	Run	00/&SUF.
XCF_CF_STR_SCM_AUGMENTED	ACTIVE (ENABLED)	SUCCESS	Residual augmented space prevents alter processing from dynamically adjusting CF structure storage usage.	Yes	N/A
XCF_CF_STR_SCM_MINCOUNTS	ACTIVE (ENABLED)	SUCCESS	Violating a minimum structure object count causes inconsistency with defined SCMMAXSIZE when duplexed.	Yes	N/A
XCF_CF_STR_SCM_UTILIZATION	ACTIVE (DISABLED)	ENV N	Structure assigned SCM is being used which indicates the structure size is insufficient for the application workload	Yes	N/A
XCF_CF_STR_SCHMAXSIZE	ACTIVE (ENABLED)	SUCCESS	Storage-Class Memory (SCM) assigned to a structure should be consistent with the structure defined SCMMAXSIZE	Yes	N/A
XCF_CF_SYSPLEX_CONNECTIVITY	ACTIVE (DISABLED)	SUCCESS	Multiple coupling facilities should be connected to all systems.	Yes	&SUF
XCF_CFRM_MSGBASED	ACTIVE (DISABLED)	SUCCESS	CFRM message-based event management improves recovery time for users of CF structures.	Yes	&SUF

Tabel B.16: Setup van Health Checker voor het ROO team tabel 10

Name	Status	Outcome	Reason	Run	00/&SUF.
XCF_CLEANUP_VALUE	ACTIVE (ENABLED)	SUCCESS	Quick removal of a partitioned system from the SYSPLEX.	Yes	N/A
XCF_DEFAULT_MAXMSG	ACTIVE (ENABLED)	SUCCESS	Avoid problems with XCF signalling.	Yes	N/A
XCF_FDI	ACTIVE (ENABLED)	SUCCESS	Allow adequate time for the system to recover before other systems in the sysplex take action to remove it.	Yes	N/A
XCF_MAXMSG_NUMBUF_RATIO	ACTIVE (ENABLED)	SUCCESS	Avoid problems with XCF signalling.	Yes	N/A
XCF_SFM_ACTIVE	ACTIVE (DISABLED)	SUCCESS	An SFM policy provides better failure management.	Yes	&SUF
XCF_SFM_CFSTRHANGTIME	ACTIVE (ENABLED)	SUCCESS	Allow the system to take automatic action to relieve a hang when an expected connector response remains overdue too long	Yes	N/A
XCF_SFM_CONNFAIL	ACTIVE (DISABLED)	SUCCESS	Allow SFM to reconfigure the sysplex when one or more systems lose signalling connectivity.	Yes	&SUF

Tabel B.17: Setup van Health Checker voor het ROO team tabel 11

Name	Status	Outcome	Reason	Run	00/&SUF.
XCF_SFM_SSUMLIMIT	ACTIVE (ENABLED)	SUCCESS	Allow SFM to initiate status update missing processing when a system remains in a degraded state too long.	Yes	N/A
XCF_SFM_SUM_ACTION	ACTIVE (ENABLED)	SUCCESS	Allow SFM to fence and partition a system without operator intervention and without undue delay.	Yes	N/A
XCF_SIG_CONNECTIVITY	ACTIVE (ENABLED)	SUCCESS	Avoid problems with XCF signalling.	Yes	N/A
XCF_SIG_PATH_SEPARATION	ACTIVE (ENABLED)	SUCCESS	XCF signaling connections should have no single point of failure.	Yes	N/A
XCF_SIG_STR_SIZE	ACTIVE (ENABLED)	SUCCESS	XCF signaling structures should be of sufficient size to support all systems in the target sysplex.	Yes	N/A
XCF_SYSPLEX_CDS_CAPACITY	ACTIVE (ENABLED)	SUCCESS	Check sysplex CDS capacities.	Yes	N/A

Tabel B.18: Setup van Health Checker voor het ROO team tabel 12

Name	Status	Outcome	Reason	Run	00/&SUF.
XCF_SYSTATDET_PARTITIONING	ACTIVE (ENABLED)	EXCEPTION	Verify best-practice recommendation of the SSD partitioning protocol being enabled for expeditious removal of failed systems.	Yes	N/A
XCF_TCLASS_CLASSLEN	ACTIVE (ENABLED)	SUCCESS	Avoid problems with XCF signalling.	Yes	N/A
XCF_TCLASS_CONNECTIVITY	ACTIVE (ENABLED)	SUCCESS	Avoid problems with XCF signalling.	Yes	N/A
XCF_TCLASS_HAS_UNDESIG	ACTIVE (ENABLED)	SUCCESS	Avoid problems with XCF signalling.	Yes	N/A
GRS_AUTHQLVL_SETTINGS	ACTIVE (ENABLED)	SUCCESS	If the AUTHQLVL parameter is not set to the maximum level, certain requests may be susceptible to denial of service attacks.	Yes	N/A
GRS_CONVERT_RESERVES	ACTIVE (ENABLED)	SUCCESS	When in STAR mode, converting RESERVES can help improve performance and avoid deadlock.	Yes	N/A

Tabel B.19: Setup van Health Checker voor het ROO team tabel 13

Name	Status	Outcome	Reason	Run	00/&SUUF.
GRS_EXIT_PERFORMANCE	ACTIVE (ENABLED)	SUCCESS	Certain exits may negatively impact system performance.	Yes	N/A
GRS_GRSQ_SETTING	ACTIVE (ENABLED)	SUCCESS	IBM recommends a GRSQ setting of contention.	Yes	N/A
GRS_MODE	ACTIVE (ENABLED)	SUCCESS	GRS should run in STAR mode to improve performance.	Yes	N/A
GRS_RNL_IGNORED_CONV	ACTIVE (ENABLED)	SUCCESS	If an entry in the RE-SERVE Conversion performance RNL is superseded by an entry in the SYSTEMS Exclusion RNL, it will be ignored.	Yes	N/A
GRS_SYNCHRES	ACTIVE (ENABLED)	SUCCESS	GRS synchronous RE-SERVE processing should be enabled to avoid deadlock conditions.	Yes	N/A
GRS_CONVERT_RESERVES	ACTIVE (ENABLED)	SUCCESS	When in STAR mode, converting RESERVES can help improve performance and avoid deadlock.	Yes	N/A

Tabel B.20: Setup van Health Checker voor het ROO team tabel 14

Name	Status	Outcome	Reason	Run	00/&SUF.
MS_MHS_TM_HOLDS	ACTIVE (ENABLED))	EXCEPTION	ThruPut Manager MHS_TM Held Jobs.	No	&SUF
MS_TM_CLASSES	ACTIVE (ENABLED))	SUCCE	ThruPut Manager Job Clas- ses.	Yes	N/A
MS_TM_JES2_EXITS	ACTIVE (ENABLED))	SUCCE	ThruPut Manager JES2 Exit Status.	Yes	N/A
MS_TM_LICENSE	ACTIVE (ENABLED))	SUCCE	ThruPut Manager License Validation.	Yes	N/A
MS_TM_OPTIONS	ACTIVE (ENABLED))	SUCCE	ThruPut Manager Options.	Yes	N/A
MS_TM_SETTINGS	ACTIVE (ENABLED))	SUCCE	ThruPut Manager Support Options.	Yes	N/A
MS_TM_SMF	ACTIVE (ENABLED))	SUCCE	ThruPut Manager SMF Exits.	Yes	N/A
MS_TMSS_ACTIVE	ACTIVE (ENABLED))	SUCCE	ThruPut Manager Started Task Status.	Yes	N/A

Tabel B.21: Setup van Health Checker voor het RTC team tabel 1

Name	Status	Outcome	Reason	Run	00/&SUF.
ASM_LOCAL_SLOT_USAGE	ACTIVE (ENABLED)	SUCCESS	To check the local page data set utilization	No	00
ASM_NUMBER_LOCAL_DATASETS	ACTIVE (ENABLED)	SUCCESS	To ensure that the recommended number of local page data sets are in use	No	00
ASM_PAGE_ADD	ACTIVE (ENABLED)	SUCCESS	To ensure that additional page data sets can be added dynamically	No	00
ASM_PLPA_COMMON_SIZE	ACTIVE (DISABLED)	ENV N	To ensure that the PLPA and common page data sets are sized correctly	No	00
ASM_PLPA_COMMON_USAGE	ACTIVE (DISABLED)	ENV N	To check on PLPA and common page data sets utilization	No	00
RSM_AFQ	ACTIVE (ENABLED)	SUCCESS	System may not recover in time if set too low	Yes	N/A
RSM_HVCOMMON	ACTIVE (ENABLED)	SUCCESS	Availability may be impacted	Yes	N/A
RSM_HVSHARE	ACTIVE (ENABLED)	SUCCESS	Availability may be impacted	Yes	N/A
RSM_INCLUDEIMACF	ACTIVE (ENABLED)	SUCCESS	Obsolete keyword has no effect	Yes	N/A
RSM_MAXCADS	ACTIVE (ENABLED)	SUCCESS	Performance may be impacted	Yes	N/A
RSM_MEMLIMIT	ACTIVE (ENABLED)	SUCCESS	Performance may be impacted	Yes	N/A

Tabel B.22: Setup van Health Checker voor het RTC team tabel 2

Name	Status	Outcome	Reason	Run	00/&SUF.
RSM_MINIMUM_REAL	ACTIVE (ENABLED)	SUCCESS	Availability may be impacted	Yes	N/A
RSM_REAL	ACTIVE (ENABLED)	SUCCESS	Performance may be impacted	Yes	N/A
RSM_RSU	ACTIVE (ENABLED)	SUCCESS	Performance may be impacted	Yes	N/A
IEA_ASIDS	ACTIVE (ENABLED)	SUCCESS	ASIDs are a finite resource. It is important to know how many remain available.	Yes	N/A
IEA_LXS	ACTIVE (ENABLED)	SUCCESS	LXs are a finite resource. It is important to know how many remain available.	Yes	N/A
SUP_HIPERDISPATCH	ACTIVE (ENABLED)	SUCCESS	A system runs with optimal performance when HyperDispatch is enabled.	Yes	N/A
SUP_HIPERDISPATCHCPUCONFIG	ACTIVE (ENABLED)	SUCCESS	HyperDispatch disabled can only use CPU ID 0 through 3Fx.	No	00
RCF_PCCA_ABOVE_16M	ACTIVE (ENABLED)	SUCCESS	More storage is available below 16M when the PC-CAs are RMODE 31.	Yes	N/A
SUP_LCCA_ABOVE_16M	ACTIVE (ENABLED)	SUCCESS	More storage is available below 16M when the LC-CAs are RMODE 31.	Yes	N/A

Tabel B.23: Setup van Health Checker voor het RTC team tabel 3

Name	Status	Outcome	Reason	Run	00/&SUF.
SUP_SYSTEM_SYMBOL_TABLE_SIZE	ACTIVE (ENABLED)	SUCCESS	Monitor the size of the system symbol table	Yes	N/A
VSM_ALLOWUSERKEYCSA	INACTIVE (ENABLED)	INACT	Validate the AllowUser-KeyCSA DIAGxx Setting	No	00
VSM_CSA_CHANGE	ACTIVE (ENABLED)	SUCCESS	Audit reductions in size of private virtual storage	Yes	N/A
VSM_CSA_LARGEST_FREE	ACTIVE (ENABLED)	SUCCESS	Audit the size of the largest free CSA block	Yes	N/A
VSM_CSA_LIMIT	ACTIVE (ENABLED)	SUCCESS	Audit the size of CSA	Yes	N/A
VSM_CSA_THRESHOLD	ACTIVE (ENABLED)	SUCCESS	Monitor the allocation of CSA	Yes	N/A
VSM_PVT_LIMIT	ACTIVE (ENABLED)	SUCCESS	Audit the size of private storage	Yes	N/A
VSM_SQA_LIMIT	ACTIVE (ENABLED)	SUCCESS	Audit the size of SQA	Yes	N/A
VSM_SQA_THRESHOLD	ACTIVE (ENABLED)	SUCCESS	Monitor the allocation of SQA	Yes	N/A

Tabel B.24: Setup van Health Checker voor het RTC team tabel 4

Name	Status	Outcome	Reason	Run	00/&SUF.
PMO_LNKLST_MEM_NOT_FNF	ACTIVE (ENABLED)	SUCCESS	Ensure that CA PMO is adequately assisting LN- KLST searches.	Yes	N/A
PMO_NUMADDED_EXCEEDED_PCT	ACTIVE (ENABLED)	SUCCESS	Ensure there are enough en- tries for private libraries.	Yes	N/A
PMO_TIME_BETWEEN_SORTS	ACTIVE (ENABLED)	SUCCESS	Ensure that the managed list is big enough to pro- vide best performance for directory searches of LN- KLST members with CA PMO.	Yes	N/A
VLF_MAXVIRT	ACTIVE (ENABLED)	SUCCESS	Help to optimize VLF data space usage.	Yes	N/A

Tabel B.25: Setup van Health Checker voor het RTC team tabel 5

Name	Status	Outcome	Reason	Run	00/0&SUF.
ACF2_AUTO_START_CHECK	ACTIVE (ENABLED)	SUCCESS	Check AFC2 auto start setting	Yes	N/A
ACF2_CHECK_DATABASES	ACTIVE (ENABLED)	SUCCESS	Check AFC2 Databases	Yes	N/A
ACF2_CHECK_EXITS	ACTIVE (ENABLED)	SUCCESS	Check AFC2 Exits (Was inactive?)	No	N/A
ACF2_CHECK_JES2_EXITS	ACTIVE (ENABLED)	SUCCESS	Check AFC2 JES2 exits	Yes	N/A
ACF2_SAFDEF_NOAPF_CHECK	ACTIVE (ENABLED)	EXCEPTION	Check AFC2 SAFDEFS for NOAPF CHECK (Was inactive?)	No	N/A
ICSF_COPROCESSOR_STATE_NEGCHANGE	ACTIVE (ENABLED)	SUCCESS	Detects degradation in coprocessor state.	Yes	N/A
ICSF_DEPRECATED_SERV_WARNINGS	ACTIVE (ENABLED)	SUCCESS	Detects use of deprecated callable service.	Yes	N/A

Tabel B.26: Setup van Health Checker voor het Security team tabel 1

Name	Status	Outcome	Reason	Run	00/&SUF.
ICSF_KEY_EXPIRATION	ACTIVE (ENABLED)	SUCCESS	Detects keys and objects that are expiring.	Yes	N/A
ICSF_MASTER_KEY_CONSISTENCY	ACTIVE (ENABLED)	SUCCESS	Detects Master Key state inconsistency.	Yes	N/A
ICSF_OPTIONS_CHECKS	ACTIVE (ENABLED)	SUCCESS	Detects ICSF options not set to specified values.	Yes	N/A
ICSF_PKCS_PSS_SUPPORT	ACTIVE (ENABLED)	SUCCESS	Detects PKCS-PSS capability.	Yes	N/A
ICSF_UNSUPPORTED_CCA_KEYS	ACTIVE (ENABLED)	SUCCESS	Detects CCA keys that are unsupported.	Yes	N/A
ICSF_WEAK_CCA_KEYS	ACTIVE (ENABLED)	SUCCESS	Detects CCA keys that are weak.	Yes	N/A
ICSFMIG7731_ICSF_RETAINED_RSAKEY	INACTIVE (ENABLED)	INACT	Detects use of retained RSA private keys.	No/Mod	&SUF.

Tabel B.27: Setup van Health Checker voor het Security team tabel 2

Name	Status	Outcome	Reason	Run	00/&SUF.
ZOSMIGV2R4_NEXT_CS_OSIMGMT	ACTIVE (ENABLED)	SUCCESS	CHECK IF OSIMGMT IS IN USE	Yes	N/A
ZOSMIGV2R4_CS_IWQSC_TCPIP*	INACTIVE (ENABLED)	INACT	ENSURE SUFFICIENT FIXED STORAGE FOR IWQ IPSEC ENABLING	No	N/A
ZOSMIGREC_ROOT_FS_SIZE	INACTIVE (ENABLED)	INACT	Verify size accommodation for the z/OS root file sys- tem to prevent halt on in- stallation.	No	N/A
ZOSMIGV2R3_NEXT_USS_SMB_DETECTED	INACTIVE (ENABLED)	INACT	z/OS V2.3 is planned to be the last release of the opera- ting system to support the DFS/SMB functionality.	No	N/A
ZOSMIGV2R3_NEXT_VSM_USERKEYCOMM	INACTIVE (ENABLED)	INACT		No	N/A
ZOSMIGV2R4_SSH_CONFIG	INACTIVE (ENABLED)	INACT	A migration check for OpenSSH config file	No	N/A
ZOSMIGV2R4_SSHD_CONFIG	INACTIVE (ENABLED)	INACT	A migration check for OpenSSH Daemon config file	No	N/A
ZOSMIGREC_ROOT_FS_SIZE	INACTIVE (ENABLED)	INACT	Verify size accommodation for the z/OS root file sys- tem to prevent halt on in- stallation.	No	N/A
ZOSMIGV2R3_NEXT_USS_SMB_DETECTED	INACTIVE (ENABLED)	INACT	z/OS V2.3 is planned to be the last release of the opera- ting system to support the DFS/SMB functionality.	No	N/A
ZOSMIGREC_SUP_TIMER_INUSE	INACTIVE (ENABLED)	INACT	Verify STP is in use when applicable	No	N/A

Tabel B.28: Setup van Health Checker voor het SOE team tabel

Name	Status	Outcome	Reason	Run	00/8SUF.
DISK_SYSPARM_CARTCALC_N	ACTIVE (ENABLED)	SUCCESS	Ensure CARTCALC is set to Y if using Cartridge tapes	Yes	N/A
DISK_SYSPARM_DSNDUPVL_N	ACTIVE (ENABLED)	SUCCESS	Ensure DSNDUPVL is set to Y if using volume pooling	Yes	N/A
DISK_SYSPARM_IOCHNBLK_1	ACTIVE (ENABLED)	SUCCESS	IOCHNBLK is set greater than 1 if using tape cartridges.	Yes	N/A
DISK_SYSPARM_IOTRACKS_1	ACTIVE (ENABLED)	SUCCESS	Ensure IOTRACKS is set to the number of tracks on a cylinder.	Yes	N/A
DISK_SYSPARM_IXMQTIME_0	ACTIVE (ENABLED)	SUCCESS	Ensure IXMQTIME is set greater than 0	Yes	N/A
DISK_SYSPARM_MERGSORT_S	ACTIVE (ENABLED)	SUCCESS	Ensure MERGSORT is set to S	Yes	N/A
ALLOC_ALLC_OFFLN_POLICY	ACTIVE (ENABLED)	SUCCESS	Check ALLC_OFFLN Policy	Yes	N/A
ALLOC_SPEC_WAIT_POLICY	ACTIVE (ENABLED)	SUCCESS	CHECK SPEC_WAIT Policy	Yes	N/A
ALLOC_TAPELIB_PREF	INACTIVE (ENABLED)	INACT	Check TAPELIB_PREF	No	00
ALLOC_TIOT_SIZE	ACTIVE (ENABLED)	SUCCESS	CHECK TIOT SIZE	Yes	N/A

Tabel B.29: Setup van Health Checker voor het storage team tabel 1

Name	Status	Outcome	Reason	Run	00/&SUF.
CATALOG_ATTRIBUTE_CHECK	ACTIVE (ENABLED)	SUCCESS	To detect and report catalog DASD status and SHAREOPTIONS if they are inconsistent.	Yes	N/A
CATALOG_IMBED_REPLICATE	INACTIVE (ENABLED)	INACT	IMBED and REPLICATE attributes are no longer allowed on the define of a new catalog	No	00
CATALOG_RNLS	ACTIVE (ENABLED)	SUCCESS	All Catalog RESERVES should be converted to SYSTEMS ENQUEUEs unless catalogs are shared outside the sysplex	Yes	N/A
DMO_REFUCB	ACTIVE (ENABLED)	SUCCESS	Reports a REFUCB function in PARMLIB	Yes	N/A
DMO_TAPE_LIBRARY_INIT_ERRORS	ACTIVE (ENABLED)	SUCCESS	Reports any tape library devices that failed to be initialized during IPL	Yes	N/A
HSM_CDSB_BACKUP_COPIES	ACTIVE (ENABLED)	SUCCESS	Ensure critical level of HSM CDS backups specified.	Mod	&SUF
HSM_CDSB_DASD_BACKUP	ACTIVE (ENABLED)	SUCCESS	Ensure required DASD data sets exist for HSM CDS backups.	No	00
HSM_CDSB_VALID_BACKUPS	ACTIVE (ENABLED)	SUCCESS	Ensure critical level of valid HSM CDS backups exist.	Mod	&SUF

Tabel B.30: Setup van Health Checker voor het Storage team tabel 2

Name	Status	Outcome	Reason	Run	00/&SUF.
IOS_BUFFER_CREDITS	ACTIVE (ENABLED)	SUCCESS	Insufficient Buffer Credits	Yes	N/A
IOS_CAPTUCB_PROTECT	ACTIVE (ENABLED)	SUCCESS	Captured UCB protection state verification.	Yes	N/A
IOS_CMRTIME_MONITOR	ACTIVE (ENABLED)	SUCCESS	Command Response Time Monitor for Channel Paths	Yes	N/A
IOS_DYNAMIC_ROUTING	ACTIVE (ENABLED)	SUCCESS	Dynamic Routing verification.	Yes	N/A
IOS_FABRIC_MONITOR	ACTIVE (ENABLED)	SUCCESS	Fabric Health Monitor for CUP Diagnostics	Yes	N/A
IOS_IORATE_MONITOR	ACTIVE (ENABLED)	SUCCESS	I/O Rate Monitor for Channel Paths	Mod	00
IOS_MIDAW	ACTIVE (ENABLED)	SUCCESS	The MIDAW facility state verification.	Yes	N/A
IOS_PORT_SPEED	ACTIVE (ENABLED)	SUCCESS	Port Speed Mismatch for Links	Yes	N/A
IOS_STORAGE_IOSBLKS	ACTIVE (ENABLED)	SUCCESS	31-bit storage for IOS blocks state verification.	Yes	N/A

Tabel B.31: Setup van Health Checker voor het Storage team tabel 3

Name	Status	Outcome	Reason	Run	00/&SUF.
OCE_XTIOT_CHECK	ACTIVE (ENABLED)	SUCCE	Check whether XTIOs for non VSAM is enabled.	Yes	N/A
PDSE_SMSPDSE1	ACTIVE (ENABLED)	SUCCE	Check for the presence of SMSPDSE1.	Yes	N/A
SMS_CDS_REUSE_OPTION	ACTIVE (ENABLED)	SUCCE	Avoid ACDS or COMMDS running into space problems	Yes	N/A
SMS_CDS_SEPARATE	ACTIVE (ENABLED)	SUCCE	Prevent single point of failure for ACDS and COMMDS	Yes	N/A
VSAM_CA_RECLAIM	ACTIVE (ENABLED)	SUCCE	IBM recommends running with CA reclaim enabled to provide improved DASD space usage and performance enhancements.	Yes	N/A
VSAM-INDEX-TRAP	INACTIVE (ENABLED)	SUCCE	VSAM Index Trap provides better first failure data capture	Yes	00
VSAMRLS_CFCACHE_MINIMUM_SIZE	ACTIVE (ENABLED)	SUCCE	Reduces performance impact due to small CF cache size	Yes	&SUF
VSAMRLS_CFLS_FALSE_CONTENTION	ACTIVE (DISABLED)	SUCCE	Prevents performance degradation due to false lock contention	Mod	&SUF
VSAMRLS_DIAG_CONTENTION	ACTIVE (ENABLED)	SUCCE	Gives the customer ability to monitor VSAM RLS contention	Yes	N/A

Tabel B.32: Setup van Health Checker voor het Storage team tabel 4

Name	Status	Outcome	Reason	Run	00/&SUF.
VSAMRLS_QUIESCE_STATUS	ACTIVE (ENABLED)	SUCCESS	Unresponsive CICS regions for QUIESCE and UNQUIESCE could indicate a problem	Yes	N/A
VSAMRLS_SHCDS_CONSISTENCY	ACTIVE (DISABLED)	SUCCESS	Optimizes space utilization	Mod	&SUF
VSAMRLS_SHCDS_MINIMUM_SIZE	ACTIVE (DISABLED)	SUCCESS	Prevents performance impact due to small SHCDS size	Mod	&SUF
VSAMRLS_SINGLE_POINT_FAILURE	ACTIVE (DISABLED)	SUCCESS	Prevents a single point of failure due to a lost volumes.	Mod	&SUF
VSAMRLS_TVS_ENABLED	ACTIVE (DISABLED)	ENV N	Verify that DFSMSStvs is enabled.	No	00
VANTAGE_HELP_DD_IN_USE@*	ACTIVE (ENABLED)	SUCCESS	CA Vantage should use a //HELP DD statement, not the HELP sysparm.	Yes	N/A
VANTAGE_LOG_THRSH@*	ACTIVE (ENABLED)	SUCCESS	Verify CA Vantage is configured to run with enough free space for log processing.	Yes	N/A
VANTAGE_LOGBREAK_USAGE@*	ACTIVE (ENABLED)	SUCCESS	Verify CA Vantage script use of the script keyword SET_LOGBREAK value.	Yes	N/A

Tabel B.33: Setup van Health Checker voor het Storage team tabel 5

Name	Status	Outcome	Reason	Run	00/&SUF.
VANTAGE_MEM_ABOVE_THRSH@*	ACTIVE (ENABLED)	SUCCESS	Check if running out of virtual memory	Yes	N/A
VANTAGE_MEM_BELOW_THRSH@*	ACTIVE (ENABLED)	SUCCESS	Check if running out of virtual memory	Yes	N/A
VANTAGE_OLD_AUT_INACT@*	ACTIVE (ENABLED)	SUCCESS	Verify CA Vantage is configured to run with the old threshold automation inactive.	Yes	N/A
VANTAGE_SCRIPT_ACCUM@*	ACTIVE (ENABLED)	SUCCESS	Check if scripts accumulate forever.	Yes	N/A
VANTAGE_STMON_ACT@*	ACTIVE (ENABLED)	SUCCESS	The Internal Status Monitor is recommended to be active to take advantage of the Internal Status Monitor functions.	Yes	N/A
VANTAGE_VANTREL_SYSPARM@*	ACTIVE (ENABLED)	SUCCESS	Verify CA Vantage is configured to run with the VANTREL sysparm default setting	Yes	N/A
VANTAGE_WARMSTART_ACT@*	ACTIVE (ENABLED)	SUCCESS	Verify CA Vantage is configured to run with the Warmstart feature active.	Yes	N/A
VANTAGE_ZIIP_ENABLED@*	ACTIVE (ENABLED)	SUCCESS	It is recommended that you make CA Vantage code eligible to be dispatched on zIIP - set sysparm ZIIPIT to (Y).	Yes	N/A

Tabel B.34: Setup van Health Checker voor het Storage team tabel 6

Name	Status	Outcome	Reason	Run	00/&SUF.
USS_AUTOMOUNT_DELAY	ACTIVE (ENABLED)	SUCCESS	Low automount delay times in a SYSPLEX can cause the system to hang, continually trying to unmount file systems and failing.	Yes	N/A
USS_CLIENT_MOUNTS	ACTIVE (ENABLED)	SUCCESS	File systems should not function ship if they can be mounted locally. Performance is not optimal in this situation.	Yes	N/A
USS_FILESYS_CONFIG	ACTIVE (ENABLED)	SUCCESS	Avoid performance problems in Unix System services and improve recovery after an outage	Yes	N/A
USS_HFS_DETECTED	ACTIVE (ENABLED)	EXCEPTION	HFS file systems are no longer the strategic file system. All HFS file systems should be migrated to zFS	Yes	N/A
USS_INETD_UNSECURE_SERVICES	ACTIVE (ENABLED)	SUCCESS	Check if inetd daemon is configured with insecure services which may result in a security exposure.	Yes	N/A
USS_KERNEL_PVTSTG_THRESHOLD	ACTIVE (ENABLED)	SUCCESS	If no private storage is available in the kernel some UNIX System Services syscalls will fail. This can result in an outage.	Yes	N/A

Tabel B.35: Setup van Health Checker voor het zOPEN team tabel 1

Name	Status	Outcome	Reason	Run	00/&SUF.
USS_KERNEL_RESOURCES_THRESHOLD	ACTIVE (DISABLED)	ENV N	Running out of z/OS UNIX System Services resources can cause system calls to start failing.	Yes	N/A
USS_KERNEL_STACKS_THRESHOLD	ACTIVE (ENABLED)	SUCCESS	If no kernel stack cells are available some UNIX System Services syscalls will fail. This can result in an outage.	Yes	N/A
USS_MAXSOCKETS_MAXFILEPROC	ACTIVE (ENABLED)	EXCEPTION	If MAXSOCKETS or MAXFILEPROC are set too low you can run out of usable sockets or file descriptors respectively.	Mod	00
USS_PARMLIB	ACTIVE (ENABLED)	EXCEPTION	Reconfiguration settings should be kept in a permanent location so they are available the next time z/OS UNIX is initialized.	Yes	N/A
USS_PARMLIB_MOUNTS	ACTIVE (ENABLED)	EXCEPTION	BPXPRMxx parmlib mount failures can cause outages if not handled in a timely manner.	Yes	N/A
USS_SUPERUSER	ACTIVE (ENABLED)	SUCCESS	The userID value for SUPERUSER must be defined to the security product and have an OMVS segment with a UID of 0.	Yes	N/A

Tabel B.36: Setup van Health Checker voor het zOPEN team tabel 2

Name	Status	Outcome	Reason	Run	00/&SUF.
ZFS_CACHE_REMOVALS	ACTIVE(ENAB	SEEDCES	Verifies whether the default meta cache/transaction/client cache sizes are used	Yes	N/A
ZFS_VERIFY-CAHCE SIZE	ACTIVE(ENAB	SEEDCES	Verifies whether the default cache sizes are used	Yes	N/A
ZFS_VERIFY_COMPRESSION_HEALTH	ACTIVE(ENAB	SEEDCES	Verifies that all user cache pages are reistered with the zEDC Express service when there are compressed aggregates.	Yes	N/A

Tabel B.37: Setup van Health Checker voor het zOPEN team tabel 3

Name	Status	Outcome	Reason	Run	00/&SUF.
DB2_OFS_CREATE_VIEW@PTX*	ACTIVE (ENABLED)	SUCCESS	Validate value that is specified for the GENE-RATE_COLUMN _NAME_WITH_SELECT(*)	Yes	N/A
DB2_PDA_BP_LOOKUP@PTX*	ACTIVE (ENABLED)	SUCCESS	Validate value specified for the DB2_BUFFER_CHECKING parameter	Yes	N/A
DB2_EXCP_ACCESS@PTX*	ACTIVE (ENABLED)	SUCCESS	Validate value specified for the EXCP_READ_ACCESS parameter	Yes	N/A
DB2_PPA_EXPLAIN_RC1@*	ACTIVE (ENABLED)	SUCCESS	Ensure Plan Analyzer EXPLRC1 is set to a non-zero return code.	Yes	N/A
DB2_PPA_EXPLAIN_RC2@*	ACTIVE (ENABLED)	SUCCESS	Ensure Plan Analyzer EXPLRC2 is set to a non-zero return code.	Yes	N/A
DB2_PPA_EXPLAIN_RC3@*	ACTIVE (ENABLED)	SUCCESS	Ensure Plan Analyzer EXPLRC3 is set to a non-zero return code.	Yes	N/A
DB2_PRA_#LOAD_LAST_RUN@PTX*	INACTIVE (ENABLED)	INACT	Check when PRA#LOAD was last run	No	00
DB2_PRA_PARM_LEVEL@PTX*	ACTIVE (ENABLED)	SUCCESS	Ensure the parm level and program level are the same	Yes	N/A
DB2_PRA_TAPE_RETENTION@PTX*	INACTIVE (ENABLED)	INACT	Determine if tape retention or expiration is set to a value other than default	No	00

Tabel B.38: Setup van Health Checker voor het DB team tabel 13

Name	Status	Outcome	Reason	Run	00/&SUF.
DB2_RCM_FULLTREE_CHECK@PTX*	INACTIVE (ENABLED)	INACT	Validate that the optimum FULLTREE option is used for R12	No	00
DTCM_DB_FORCE_CHKPT@*	ACTIVE (ENABLED)	SUCCESS	Ensure no forced check points	Yes	N/A
DTCM_DB_MAX_TSK@*	ACTIVE (ENABLED)	SUCCESS	Too few tasks for concurrent work can cause application delays or failures	Yes	N/A
IDMS_CHANGE_TRACKING@SYST*	ACTIVE (ENABLED)	EXCEPTION	Verify whether CA IDMS Change Tracking is enabled.	Yes	N/A
IDMS_CPU_EFFECTIVENESS@SYST*	ACTIVE (ENABLED)	SUCCESS	Calculate the ratio of CPU time received versus CPU time required to provide a measure of involuntary waits for CPU time.	Yes	N/A
IDMS_SCRATCH_IN_MEMORY@SYST*	ACTIVE (ENABLED)	SUCCESS	Verify if scratch is in memory.	Yes	N/A
IDMS_ZIIP_USAGE@SYST*	ACTIVE (ENABLED)	SUCCESS	Verify whether CA IDMS exploits zIIP processors.	Yes	N/A

Tabel B.39: Setup van Health Checker voor het DB team tabel 2

B.3 Parmlib members voor z/OS Health Checker

In deze bijlagen bevindt zich de Parmlib members die aangemaakt zijn voor de standaard-opstelling. HZSPRMT4 is de member die voortaan gebruikt zal worden bij het uitbreiden van de parallel sysplex.

B.3.1 HZSPRM00

```

LOGGER=ON,LOGSTREAMNAME=HZS.HEALTH.CHECKER.HISTORY
/*                                                                    */
/* STATEMENTNAME use MOD00nn (nn sequence number)                    */
/* if override is needed in an LPAR use same name in LPAR specific  */
/* HZSPRMnn                                                            */
ADDREPLACE POLICY,STATEMENTNAME(MOD0001)
    UPDATE CHECK(IBMUSS,USS_MAXSOCKETS_MAXFILEPROC)
    PARM('35000,64000')
    DATE(20070608)
    REASON('HCL default value in VT')
ADDREPLACE POLICY,STATEMENTNAME(MOD0002)
    UPDATE CHECK(IBMVSM,VSM_ALLOWUSERKEYCSA)
    inactive
    DATE(20190130)
    REASON('HCL have default YES')
ADDREPLACE POLICY,STATEMENTNAME(MOD0003)
    UPDATE CHECK(IBMUSS,USS_FILESYS_CONFIG)
    SEVERITY(HI)
    INTERVAL(06:00)
    PARM('SYSPLEX')
    DATE(20110128)
    REASON('SYSPLEX.ROOT must be mounted AUTOMOVE')
ADDREPLACE POLICY,STATEMENTNAME(MOD0004)
    UPDATE CHECK(IBMxcf,XCF_CDS_SPOF)
    inactive
    DATE(20111119)
    REASON('sufficient separation')
ADDREPLACE POLICY,STATEMENTNAME(MOD0005)
    UPDATE CHECK(IBMios,ios_iorate_monitor)
    INTERVAL(ONETIME)
    SEVERITY(MEDIUM)
    DATE(20131219)
    REASON('Check at IPL only')
ADDREPLACE POLICY,STATEMENTNAME(MOD0006)
    UPDATE CHECK(CA_DB2,DB2_PRA_#LOAD_LAST_RUN@PTXMAN*)
    inactive
    DATE(20190130)

```

```
    REASON(' N/A ')
ADDREPLACE POLICY,STATEMENTNAME(MOD0007)
    UPDATE CHECK(CA_DB2,DB2_PRA_TAPE_RETENTION@PTXMAN*)
    inactive
    DATE(20190130)
    REASON(' N/A ')
ADDREPLACE POLICY,STATEMENTNAME(MOD0008)
    UPDATE CHECK(CA_DB2,DB2_RCM_FULLTREE_CHECK@PTXMAN*)
    inactive
    DATE(20190130)
    REASON(' N/A ')
ADDREPLACE POLICY,STATEMENTNAME(MOD0009)
    UPDATE CHECK(IBM_CATALOG,CATALOG_IMBED_REPLICATE)
    inactive
    DATE(20151001)
    REASON(' N/A ')
ADDREPLACE POLICY,STATEMENTNAME(MOD0010)
    UPDATE CHECK(IBMCS,CSTCP_CINET_PORTRNG_RSV_TCP*)
    inactive
    DATE(20170504)
    REASON('Check not possible in HCL')
ADDREPLACE POLICY,STATEMENTNAME(MOD0011)
    UPDATE CHECK(IBM_CICS,CICS_CEDA_ACCESS)
    INACTIVE
    DATE(20180607)
    REASON('Functionality not used')
ADDREPLACE POLICY,STATEMENTNAME(MOD0012)
    UPDATE CHECK(IBM_CICS,CICS_JOBSUB_SPOOL)
    INACTIVE
    DATE(20180607)
    REASON('Functionality not used')
ADDREPLACE POLICY,STATEMENTNAME(MOD0013)
    UPDATE CHECK(IBM_CICS,CICS_JOBSUB_TDQINTRDR)
    INACTIVE
    DATE(20180607)
    REASON('Functionality not used')
ADDREPLACE POLICY,STATEMENTNAME(MOD0014)
    UPDATE CHECK(IBMIXGLOGR,IXGLOGR_ENTRYTHRESHOLD)
    ACTIVE
    PARM('ALL')
    DATE(20181204)
    REASON('Standard value')
ADDREPLACE POLICY,STATEMENTNAME(MOD0015)
    UPDATE CHECK(IBMALLOC,ALLOC_TAPELIB_PREF)
    INACTIVE
    DATE(20190130)
```

```

    REASON('Storage teams active choice')
ADDREPLACE POLICY,STATEMENTNAME(MOD0016)
    UPDATE CHECK(IBMASM,ASM_LOCAL_SLOT_USAGE)
    INACTIVE
    DATE(20200427)
    REASON('Page datasets on SCM/VFM')
ADDREPLACE POLICY,STATEMENTNAME(MOD0017)
    UPDATE CHECK(IBMASM,ASM_NUMBER_LOCAL_DATASETS)
    INACTIVE
    DATE(20200427)
    REASON('Page datasets on SCM/VFM')
ADDREPLACE POLICY,STATEMENTNAME(MOD0018)
    UPDATE CHECK(IBMASM,ASM_PAGE_ADD)
    INACTIVE
    DATE(20200427)
    REASON('Page datasets on SCM/VFM')
ADDREPLACE POLICY,STATEMENTNAME(MOD0019)
    UPDATE CHECK(IBMASM,ASM_PLPA_COMMON_SIZE)
    INACTIVE
    DATE(20200427)
    REASON('Page datasets on SCM/VFM')
ADDREPLACE POLICY,STATEMENTNAME(MOD0020)
    UPDATE CHECK(IBMASM,ASM_PLPA_COMMON_USAGE)
    INACTIVE
    DATE(20200427)
    REASON('Page datasets on SCM/VFM')
ADDREPLACE POLICY,STATEMENTNAME(MOD0021)
    UPDATE CHECK(IBMHSB,HSM_CDSB_DASD_BACKUP)
    INACTIVE
    DATE(20200427)
    REASON('There is always a preallocated DASD dataset for CDS backups')
ADDREPLACE POLICY,STATEMENTNAME(MOD0022)
    UPDATE CHECK(IBMVSAMRLS,VSAMRLS_TVS_ENABLED)
    INACTIVE
    DATE(20200427)
    REASON('Feature not present in HCL setup')
ADDREPLACE POLICY,STATEMENTNAME(MOD0023)
    UPDATE CHECK(IBMCMZ,CNZ_CONSOLE_MSCOPE_AND_ROUTCODE)
    INACTIVE
    DATE(20200427)
    REASON('Uncontrolled Consoles')
ADDREPLACE POLICY,STATEMENTNAME(MOD0024)
    UPDATE CHECK(IBMCMZ,CNZ_EMCS_HARDCOPY_MSCOPE)
    INACTIVE
    DATE(20200427)
    REASON('Uncontrolled Consoles')

```



```
ADDREPLACE POLICY,STATEMENTNAME(MOD0025)
  UPDATE CHECK(IBMGDPS,GDPS_CHECK_SPOF)
  INACTIVE
  DATE(20200427)
  REASON('HCL setup')
```

B.3.2 HZSPRMT1

```
ADDREPLACE POLICY,STATEMENTNAME(MODT101)
  UPDATE CHECK(CPWR_THRUPUT_MGR,MS_MHS_TM_HOLDS)
  INACTIVE
  DATE(20200504)
  REASON('HCL CONFIG')
ADDREPLACE POLICY,STATEMENTNAME(MODT102)
  UPDATE CHECK(IBMHSM,HSM_CDSB_BACKUP_COPIES)
  SEVERITY(MED) INTERVAL(24:00) PARM('CRITVAL(2)')
  DATE(20200427)
  REASON('Ensure critical value of HSM CDS backup copies exist')
ADDREPLACE POLICY,STATEMENTNAME(MODT103)
  UPDATE CHECK(IBMHSM,HSM_CDSB_VALID_BACKUPS)
  SEVERITY(MED) INTERVAL(24:00) PARM('CRITVAL(2)')
  DATE(20200427)
  REASON('Ensure critical value of HSM CDS backup exist')
```

B.3.3 HZSPRMT2

```
LOGGER=OFF
ADDREPLACE POLICY,STATEMENTNAME(MODT201)
  UPDATE CHECK(IBMUSS,USS_FILESYS_CONFIG)
  INACTIVE
  DATE(20070608)
  REASON('VT02 always run with RW')
ADDREPLACE POLICY,STATEMENTNAME(MODT202)
  UPDATE CHECK(IBMHSM,HSM_CDSB_BACKUP_COPIES)
  SEVERITY(MED) INTERVAL(24:00) PARM('CRITVAL(2)')
  DATE(20200427)
  REASON('Ensure critical value of HSM CDS backup copies exist')
ADDREPLACE POLICY,STATEMENTNAME(MODT203)
  UPDATE CHECK(IBMHSM,HSM_CDSB_BACKUP_COPIES)
  SEVERITY(MED) INTERVAL(24:00) PARM('CRITVAL(2)')
  DATE(20200427)
  REASON('Ensure critical value of HSM CDS Backups exist')
ADDREPLACE POLICY,STATEMENTNAME(MODT204)
  UPDATE CHECK(IBM CNZ,CNZ_EMCS_INACTIVE_CONSOLES)
  INACTIVE
```

```
DATE(20200427)
REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT205)
UPDATE CHECK(IBMxcf,XCF_CDS_MAXSYSTEM)
INACTIVE
DATE(20200427)
REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT206)
UPDATE CHECK(IBMxcf,XCF_CF_ALLOCATION_PERMITTED)
INACTIVE
DATE(20200427)
REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT207)
UPDATE CHECK(IBMxcf,XCF_CF_STR_AVAILABILITY)
INACTIVE
DATE(20200427)
REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT208)
UPDATE CHECK(IBMxcf,XCF_CF_STR_DUPLEX)
INACTIVE
DATE(20200427)
REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT209)
UPDATE CHECK(IBMxcf,XCF_CF_STR_ENCRYPT)
INACTIVE
DATE(20200427)
REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT210)
UPDATE CHECK(IBMxcf,XCF_CF_STR_EXCLLIST)
INACTIVE
DATE(20200427)
REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT211)
UPDATE CHECK(IBMxcf,XCF_CF_STR_NONVOLATILE)
INACTIVE
DATE(20200427)
REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT212)
UPDATE CHECK(IBMxcf,XCF_CF_STR_POLICYSIZE)
INACTIVE
DATE(20200427)
REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT213)
UPDATE CHECK(IBMxcf,XCF_CF_STR_PREFLIST)
INACTIVE
DATE(20200427)
```

```
        REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT214)
        UPDATE CHECK(IBMxcf,XCF_CF_SYSPLEX_CONNECTIVITY)
        INACTIVE
        DATE(20200427)
        REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT215)
        UPDATE CHECK(IBMxcf,XCF_CFRM_MSGBASED)
        INACTIVE
        DATE(20200427)
        REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT216)
        UPDATE CHECK(IBMxcf,XCF_SFM_ACTIVE)
        INACTIVE
        DATE(20200427)
        REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT217)
        UPDATE CHECK(IBMxcf,XCF_SFM_CONNFAIL)
        INACTIVE
        DATE(20200427)
        REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT218)
        UPDATE CHECK(IBMGRS,GRS_CONVERT_RESERVES)
        INACTIVE
        DATE(20200427)
        REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT219)
        UPDATE CHECK(IBMGRS,GRS_MODE)
        INACTIVE
        DATE(20200427)
        REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT220)
        UPDATE CHECK(IBMGRS,GRS_RNL_IGNORED_CONV)
        INACTIVE
        DATE(20200427)
        REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT221)
        UPDATE CHECK(IBMVSAMRLS,VSAMRLS_CFCACHE_MINIMUM_SIZE)
        INACTIVE
        DATE(20200427)
        REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT222)
        UPDATE CHECK(IBMVSAMRLS,VSAMRLS_CFLS_FALSE_CONTENTION)
        INACTIVE
        DATE(20200427)
        REASON('Force global to run on VT1')
```

```

ADDREPLACE POLICY,STATEMENTNAME(MODT223)
    UPDATE CHECK(IBMVSAMRLS,VSAMRLS_SHCDS_CONSISTENCY)
        INACTIVE
        DATE(20200427)
        REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT224)
    UPDATE CHECK(IBMVSAMRLS,VSAMRLS_SHCDS_MINIMUM_SIZE)
        INACTIVE
        DATE(20200427)
        REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT225)
    UPDATE CHECK(IBMVSAMRLS,VSAMRLS_SINGLE_POINT_FAILURE)
        INACTIVE
        DATE(20200427)
        REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT226)
    UPDATE CHECK(CPWR_THRUPUT_MGR,MS_MHS_TM_HOLDS)
        INACTIVE
        DATE(20200504)
        REASON('HCL Config')

```

B.3.4 HZSPRMT3

```

ADDREPLACE POLICY,STATEMENTNAME(MODT302)
    UPDATE CHECK(IBMHSM,HSM_CDSB_BACKUP_COPIES)
        SEVERITY(MED) INTERVAL(24:00) PARM('CRITVAL(2)')
        DATE(20200427)
        REASON('Ensure critical value of HSM CDS backup copies exist')
ADDREPLACE POLICY,STATEMENTNAME(MODT303)
    UPDATE CHECK(IBMHSM,HSM_CDSB_BACKUP_COPIES)
        SEVERITY(MED) INTERVAL(24:00) PARM('CRITVAL(2)')
        DATE(20200427)
        REASON('Ensure critical value of HSM CDS Backups exist')
ADDREPLACE POLICY,STATEMENTNAME(MODT304)
    UPDATE CHECK(IBMCMNZ,CNZ_EMCS_INACTIVE_CONSOLES)
        INACTIVE
        DATE(20200427)
        REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT305)
    UPDATE CHECK(IBMxcf,XCF_CDS_MAXSYSTEM)
        INACTIVE
        DATE(20200427)
        REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT306)
    UPDATE CHECK(IBMxcf,XCF_CF_ALLOCATION_PERMITTED)

```

```
    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT307)
    UPDATE CHECK(IBMxcf,XCF_CF_STR_AVAILABILITY)
    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT308)
    UPDATE CHECK(IBMxcf,XCF_CF_STR_DUPLEX)
    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT309)
    UPDATE CHECK(IBMxcf,XCF_CF_STR_ENCRYPT)
    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT310)
    UPDATE CHECK(IBMxcf,XCF_CF_STR_EXCLLIST)
    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT311)
    UPDATE CHECK(IBMxcf,XCF_CF_STR_NONVOLATILE)
    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT312)
    UPDATE CHECK(IBMxcf,XCF_CF_STR_POLICYSIZE)
    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT313)
    UPDATE CHECK(IBMxcf,XCF_CF_STR_PREFLIST)
    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT314)
    UPDATE CHECK(IBMxcf,XCF_CF_SYSPLEX_CONNECTIVITY)
    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT315)
    UPDATE CHECK(IBMxcf,XCF_CFRM_MSGBASED)
    INACTIVE
```

```
DATE(20200427)
REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT316)
UPDATE CHECK(IBMxcf,XCF_SFM_ACTIVE)
INACTIVE
DATE(20200427)
REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT317)
UPDATE CHECK(IBMxcf,XCF_SFM_CONNFAIL)
INACTIVE
DATE(20200427)
REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT318)
UPDATE CHECK(IBMGRS,GRS_CONVERT_RESERVES)
INACTIVE
DATE(20200427)
REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT319)
UPDATE CHECK(IBMGRS,GRS_MODE)
INACTIVE
DATE(20200427)
REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT320)
UPDATE CHECK(IBMGRS,GRS_RNL_IGNORED_CONV)
INACTIVE
DATE(20200427)
REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT321)
UPDATE CHECK(IBMVSAMRLS,VSAMRLS_CFCACHE_MINIMUM_SIZE)
INACTIVE
DATE(20200427)
REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT322)
UPDATE CHECK(IBMVSAMRLS,VSAMRLS_CFLS_FALSE_CONTENTION)
INACTIVE
DATE(20200427)
REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT323)
UPDATE CHECK(IBMVSAMRLS,VSAMRLS_SHCDS_CONSISTENCY)
INACTIVE
DATE(20200427)
REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT324)
UPDATE CHECK(IBMVSAMRLS,VSAMRLS_SHCDS_MINIMUM_SIZE)
INACTIVE
DATE(20200427)
```

```

        REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT325)
    UPDATE CHECK(IBMVSAMRLS,VSAMRLS_SINGLE_POINT_FAILURE)
        INACTIVE
        DATE(20200427)
        REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT326)
    UPDATE CHECK(CPWR_THRUPUT_MGR,MS_MHS_TM_HOLDS)
        INACTIVE
        DATE(20200503)
        REASON('HCL Config')

```

B.3.5 HZSPRMT4

```

LOGGER=OFF
ADDREPLACE POLICY,STATEMENTNAME(MODT401)
    UPDATE CHECK(IBMUSS,USS_FILESYS_CONFIG)
        INACTIVE
        DATE(20070608)
        REASON('VT02 always run with RW')
ADDREPLACE POLICY,STATEMENTNAME(MODT402)
    UPDATE CHECK(IBMHSM,HSM_CDSB_BACKUP_COPIES)
        SEVERITY(MED) INTERVAL(24:00) PARM('CRITVAL(2)')
        DATE(20200427)
        REASON('Ensure critical value of HSM CDS backup copies exist')
ADDREPLACE POLICY,STATEMENTNAME(MODT403)
    UPDATE CHECK(IBMHSM,HSM_CDSB_BACKUP_COPIES)
        SEVERITY(MED) INTERVAL(24:00) PARM('CRITVAL(2)')
        DATE(20200427)
        REASON('Ensure critical value of HSM CDS Backups exist')
ADDREPLACE POLICY,STATEMENTNAME(MODT404)
    UPDATE CHECK(IBM CNZ,CNZ_EMCS_INACTIVE_CONSOLES)
        INACTIVE
        DATE(20200427)
        REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT405)
    UPDATE CHECK(IBM XCF,XCF_CDS_MAXSYSTEM)
        INACTIVE
        DATE(20200427)
        REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT406)
    UPDATE CHECK(IBM XCF,XCF_CF_ALLOCATION_PERMITTED)
        INACTIVE
        DATE(20200427)
        REASON('Force global to run on VT1')

```

```
ADDREPLACE POLICY,STATEMENTNAME(MODT407)
  UPDATE CHECK(IBMxcf,XCF_CF_STR_AVAILABILITY)
    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT408)
  UPDATE CHECK(IBMxcf,XCF_CF_STR_DUPLEX)
    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT409)
  UPDATE CHECK(IBMxcf,XCF_CF_STR_ENCRYPT)
    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT410)
  UPDATE CHECK(IBMxcf,XCF_CF_STR_EXCLLIST)
    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT411)
  UPDATE CHECK(IBMxcf,XCF_CF_STR_NONVOLATILE)
    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT412)
  UPDATE CHECK(IBMxcf,XCF_CF_STR_POLICYSIZE)
    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT413)
  UPDATE CHECK(IBMxcf,XCF_CF_STR_PREFLIST)
    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT414)
  UPDATE CHECK(IBMxcf,XCF_CF_SYSplex_CONNECTIVITY)
    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT415)
  UPDATE CHECK(IBMxcf,XCF_CFRM_MSGBASED)
    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT416)
```



```
UPDATE CHECK(IBMxcf,XCF_SFM_ACTIVE)
  INACTIVE
  DATE(20200427)
  REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT417)
UPDATE CHECK(IBMxcf,XCF_SFM_CONNFAIL)
  INACTIVE
  DATE(20200427)
  REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT418)
UPDATE CHECK(IBMGRS,GRS_CONVERT_RESERVES)
  INACTIVE
  DATE(20200427)
  REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT419)
UPDATE CHECK(IBMGRS,GRS_MODE)
  INACTIVE
  DATE(20200427)
  REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT420)
UPDATE CHECK(IBMGRS,GRS_RNL_IGNORED_CONV)
  INACTIVE
  DATE(20200427)
  REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT421)
UPDATE CHECK(IBMVSAMRLS,VSAMRLS_CFCACHE_MINIMUM_SIZE)
  INACTIVE
  DATE(20200427)
  REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT422)
UPDATE CHECK(IBMVSAMRLS,VSAMRLS_CFLS_FALSE_CONTENTION)
  INACTIVE
  DATE(20200427)
  REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT423)
UPDATE CHECK(IBMVSAMRLS,VSAMRLS_SHCDS_CONSISTENCY)
  INACTIVE
  DATE(20200427)
  REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT424)
UPDATE CHECK(IBMVSAMRLS,VSAMRLS_SHCDS_MINIMUM_SIZE)
  INACTIVE
  DATE(20200427)
  REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT425)
UPDATE CHECK(IBMVSAMRLS,VSAMRLS_SINGLE_POINT_FAILURE)
```

```

    INACTIVE
    DATE(20200427)
    REASON('Force global to run on VT1')
ADDREPLACE POLICY,STATEMENTNAME(MODT426)
    UPDATE CHECK(CPWR_THRUPUT_MGR,MS_MHS_TM_HOLDS)
    INACTIVE
    DATE(20200427)
    REASON('HCL Config')

```

B.4 JCL jobs

Dit zijn de JCL jobs die gebruikt worden voor het opstellen van de log die daarna doorgestuurd word via mail.

B.4.1 FHZSVT11

```

//FHZSVT11 JOB (610VV110000,4352,, ,1800),'RTN=FHZSPRVT',CLASS=D,
//          MSGCLASS=A,MSGLEVEL=(1,1)
/*ROUTE XEQ NJEVT
/*ROUTE PRINT NJOVT
//**JBS BIND VT01
//*
//HZSPRINT EXEC PGM=HZSPRNT,
//  PARM='CHECK(*,*),EXCEPTIONS'
//SYSOUT DD DISP=(NEW,PASS),DSN=%%OUT,LRECL=256,RECFM=FB,
//          DATACLAS=PSEN
//*
//SNDMEM EXEC PGM=SMTPAPIX,PARM='MSGLVL=4,FROM=NOREPLYÖVOLVO.COM',
//          COND=(0400,EQ,HZSPRINT)
//APIFILE DD *
)SEND
  ETITLE      VT01: Health Checker exceptions
  OPTION      FORCE
  DEST        jonas.braemÖsupplier.volvo.com
  DEST        david.westbrandÖhcl.com
  DEST        kevin.somersÖhcl.com
  DEST        bengt.gellingskogÖhcl.com
  MEMO FILEA
)END
//FILEA DD DISP=(OLD,DELETE),DSN=%%OUT

```

B.4.2 FHZSVT21

```
//FHZSVT21 JOB (610VV110000,4352,, ,1800), 'RTN=FHZSPRVT', CLASS=0,
//          MSGCLASS=A, MSGLEVEL=(1,1)
/*ROUTE XEQ NJEVT2
/*ROUTE PRINT NJOVT
//**JBS BIND VT02
//*
//HZSPRINT EXEC PGM=HZSPRNT,
//  PARM='CHECK(*,*) , EXCEPTIONS'
//SYSOUT DD DISP=(NEW,PASS), DSN=##OUT, LRECL=256, RECFM=FB,
//          DATACLAS=PSEN
//*
//SNDMEM EXEC PGM=SMTPAPIX, PARM='MSGLVL=4, FROM=NOREPLYÖVOLVO.COM',
//          COND=(0400,EQ,HZSPRINT)
//APIFILE DD *
)SEND
  ETITLE    VT02: Health Checker exceptions
  OPTION    FORCE
  DEST      jonas.braemÖsupplier.volvo.com
  DEST      david.westbrandÖhcl.com
  DEST      kevin.somersÖhcl.com
  DEST      bengt.gellingskogÖhcl.com
  MEMO FILEA
)END
//FILEA DD DISP=(OLD,DELETE), DSN=##OUT
```

B.4.3 FHZSVT31

```
//FHZSVT31 JOB (610VV110000,4352,, ,1800), 'RTN=FHZSPRVT', CLASS=D,
//          MSGCLASS=A, MSGLEVEL=(1,1)
/*ROUTE XEQ NJEVT
/*ROUTE PRINT NJOVT
//**JBS BIND VT03
//*
//HZSPRINT EXEC PGM=HZSPRNT,
//  PARM='CHECK(*,*) , EXCEPTIONS'
//SYSOUT DD DISP=(NEW,PASS), DSN=##OUT, LRECL=256, RECFM=FB,
//          DATACLAS=PSEN
//*
//SNDMEM EXEC PGM=SMTPAPIX, PARM='MSGLVL=4, FROM=NOREPLYÖVOLVO.COM',
//          COND=(0400,EQ,HZSPRINT)
//APIFILE DD *
)SEND
  ETITLE    VT03: Health Checker exceptions
```

```

OPTION    FORCE
DEST      jonas.braemösupplier.volvo.com
DEST      david.westbrandöhcl.com
DEST      kevin.somersöhcl.com
DEST      bengt.gellingskogöhcl.com
MEMO FILEA
)END
//FILEA DD DISP=(OLD,DELETE),DSN=&&OUT

```

B.4.4 FHZSVT41

```

//FHZSVT41 JOB (610VV110000,4352,, ,1800),'RTN=FHZSPRVT',CLASS=0,
//          MSGCLASS=A,MSGLEVEL=(1,1)
/*ROUTE XEQ NJEVT2
/*ROUTE PRINT NJOVT
/**+JBS BIND VT04
/**
//HZSPRINT EXEC PGM=HZSPRNT,
//  PARM='CHECK(*,*),EXCEPTIONS'
//SYSOUT DD DISP=(NEW,PASS),DSN=&&OUT,LRECL=256,RECFM=FB,
//          DATACLAS=PSSEN
/**
//SNDMEM EXEC PGM=SMTPAPIX,PARM='MSGLVL=4,FROM=NOREPLYÖVOLVO.COM',
//          COND=(0400,EQ,HZSPRINT)
//APIFILE DD *
)SEND
ETITLE    VT04: Health Checker exceptions
OPTION    FORCE
DEST      jonas.braemösupplier.volvo.com
DEST      david.westbrandöhcl.com
DEST      kevin.somersöhcl.com
DEST      bengt.gellingskogöhcl.com
MEMO FILEA
)END
//FILEA DD DISP=(OLD,DELETE),DSN=&&OUT

```

Bibliografie

- Bezzi, M., O'Conner, N., Philips, S. & Thiemann, U. (2010, december). *Exploiting the IBM Health Checker for z/OS Infrastructure*.
- Cosimo, G. & Kuehner, L. (2018, april). *ABCs of z/OS System Programming Volume 2*.
- Ebbers, M., Kettner, J., O'Brien, W. & Ogden, B. (2011, maart). *Introduction to the New Mainframe z/OS Basics*.
- IBM. (2010). Serving dynamic Web pages on z/OS. Verkregen van https://www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.zos.zmidwebs/zmiddle_86.htm?view=embed#zmiddle_86__usingplug
- IBM. (2019). *MVS System Messages Volume 6 (GOS -IEA)*.
- IBMCorporation. (2019, juni 24). *IBM Health Checker for z/OS User's Guide*. version 2 edition 3.
- Parziale, L., Fadel, L. & Jon, S. (2017, november). *ABCs of z/OS System Programming Volume 1*.
- Rayns, C. (2011, december). *CICS Transaction Server from Start to Finish*.
- Walle, M. (2013, februari 5). Introduction and Getting Started with the IBM Health Checker for z/OS. Verkregen van <https://share.confex.com/share/120/webprogram/Session13118.html>