

CICS TS aktuell

Heinz Peter Maassen

Lattwein GmbH

hp.maassen@lattwein.de

Hans Joachim Ebert

IBM Technical Sales zSeries

eberthj@de.ibm.com

GSE-Tagung VM/VSE

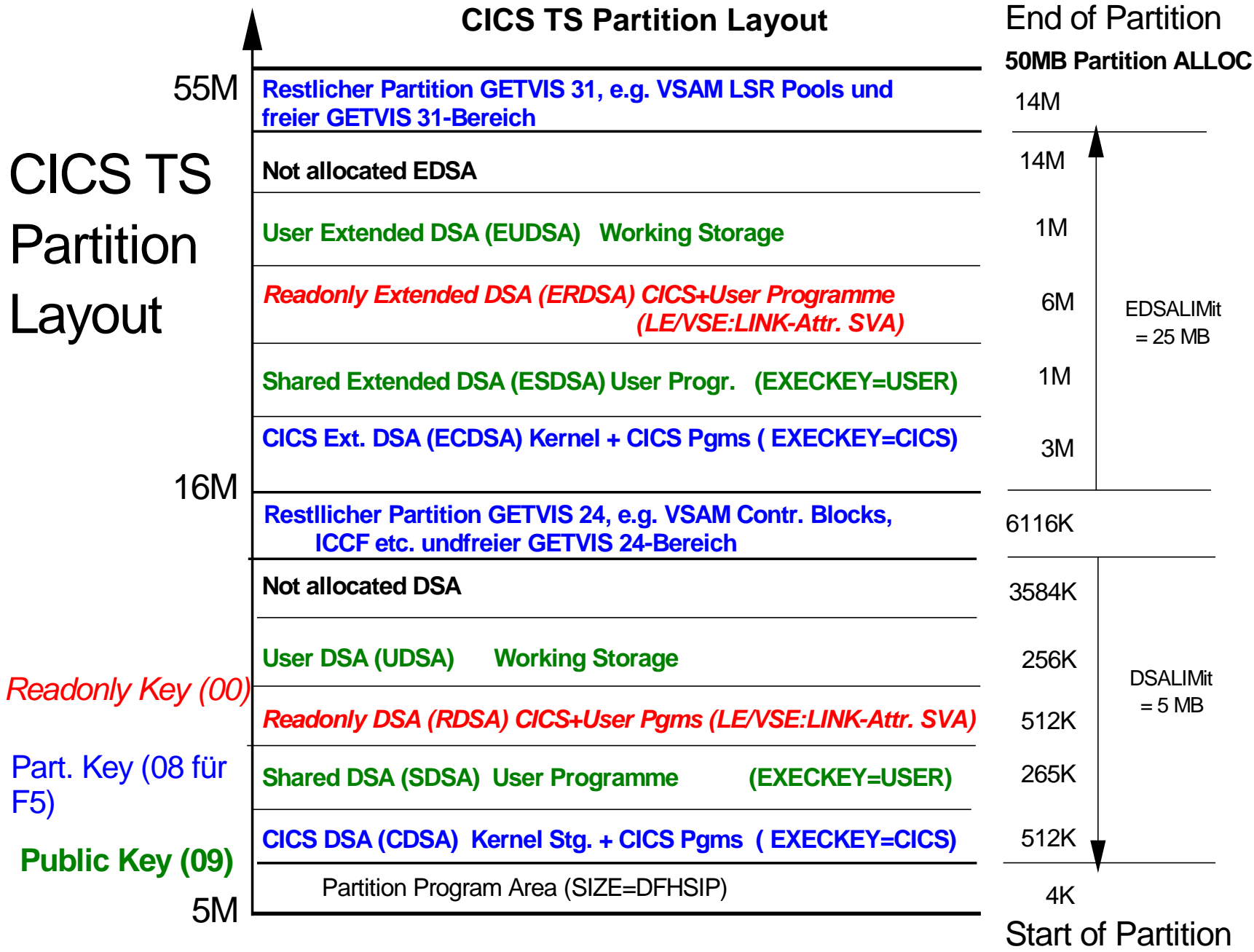
Leipzig, März 2004

Agenda Teil 1 - H. P. Maassen

- **Storage Management mit CICS TS**
- **TCTUALocation**
- **Datalocation (Program Definition)**
- **TASK Datalocation (Task Definition)**
- **EXEC CICS GETMAIN**

Storage Management mit CICS TS

- Nicht jeder Speicherbereich wird automatisch oberhalb der “magic Line“ angelegt.
 - ∅ Je VSAM Cluster werden “Below the Line“ 4 KB und je aktivem User Catalog ca. 14 KB belegt.
 - ∅ Alle Bereiche, die CICS Taskbezogen holt sind per Default below
 - ∅ Pro MXT Eintrag in der DFHSIT werden ca. 2.5 KB und ca. 13 KB im 31 Bit Bereich verwendet.
- CICS User Programme können mit dem Parameter ‘AMODE=31,RMODE=ANY‘ oberhalb geladen und ausgeführt werden, wenn sie mit LE Compilern erstellt wurden bzw. HLASM.
- Gilt auch für BMS- Maps !



TCTUALocation (BELOW/ANY)

- Wird in der DFHSIT definiert. Speicher wird in UDSA / CDSA angelegt.
- Nur ICCF und II brauchen laut Beschreibung TCTUALocation "below".
 - Ab VSE/ESA 2.7 kann TCTUAL auch ANY sein.
- 1000 Terminals und TCTUALength von 255 sind auch 255 KB.
- Erst prüfen, ob alle User Programme mit 31 Bit arbeiten.
 - Die Programme müssen mit AMODE31 gelinkt sein !
 - Das kann man über die CICS Statistik überprüfen
- Vendor Produkte können ebenfalls in der TCTUA Daten temporär ablegen.
Anfragen, ob deren Programme 31 Bit TCTUA verarbeiten können.

Datalocation (RDO –Programm Definition)

- Bei RDO für Programme kann DATAlocation auf Below oder Any eingestellt werden.
- Das gilt nur für CICS Commands, die mit der SET Option arbeiten. Nicht für INTO/FROM!
- Beispiel: EXEC CICS RECEIVE SET(R10) ... Holt einen Speicher von der EUDSA, wenn Datalocation für dieses Programm auf ANY eingestellt wurde.
- Achtung: Das hat keine Auswirkung auf den EXEC CICS GETMAIN Command.

Datalocation ...

- Falls DATALOCATION = BELOW definiert wurde, wird der Speicher aus der DSA geholt.
- Wenn ein Speicherbereich schon oberhalb der “magic Line“ geholt wurde (Bei Programm aufrufen mit XCTL/ LINK und Übergabe von Inputmsg() und unterschiedlicher Datalocation) wird der Bereich in einen 24 Bit Speicher Bereich kopiert. Das gilt nicht für CALL.
- Wenn Datalocation = ANY definiert wurde, versucht CICS den Bereich in der EDSA anzulegen.
- DATALOCATION(BELOW) für alle AMODE(24) Programme
- DATALOCATION(ANY) für AMODE(31) Programme- außer wenn diese Adressen an Amode24 Programme übergeben!

TASK Datalocation (RDO Task Def)

- TaskDataLocation gibt an ob der „Life-Time“ Storage einer Task unter oder oberhalb der 16 MB Line angefordert wird. Das sind u. a. EIB Storage und TWA Storage.
- Für alle Transaktionen, deren Programme im AMODE24 gelinkt sind, muss TASKDATAlocation = Below definiert werden sonst: **AEZC** Abend!
- EXEC CICS Command mit AMODE24 und TaskDataLocation = ANY erzeugt einen **AEZA** Abend. Auch bei Task related USER EXIT im AMODE24.

TASK Datalocation (RDO Task Def.) ...

- AMODE31 Programm mit TaskDatalocation=ANY dass ein Task Related User Exit Programm im AMODE24 ENABLEN will wird mit **AEZB** beendet.
- Falls ein Task related User Exit Programm im AMODE24 enabled wurde, werden alle Transaktionen auf TaskDataLocation=Below gesetzt bis zum Shutdown dieses CICS Laufes.

EXEC CICS GETMAIN LENGTH

- EXEC CICS GETMAIN SET(R10) LENGTH(HWORT)
INITIMG(‘ ‘)
- Dieser Command sollte so **nicht mehr** verwendet werden. Den Parameter LENGTH durch FLENGTH ersetzen.
- Holt **immer** einen Storage aus der DSA (below 16M), auch wenn das Programm mit AMODE31 gelinkt wurde.
- Maximale Größe des Speichers: 65.535 Bytes / CICSTS 65.520

EXEC CICS GETMAIN **CICSESA (2.3)**

- EXEC CICS GETMAIN SET(R10) FLENGTH(FWORT) INITIMG(' ')
- Maximale Größe mit Length=65.535
- Maximale Größe mit FLENGTH=1.073.741.824 (1 GB)

FLENGTH value	AMODE(24)	AMODE(31)
1-4095	CICS DSA	CICS DSA
4096-65 504	CICS DSA	31-bit GETVIS
65 505-1024MB	LENGERR	31-bit GETVIS
> 1024MB	LENGERR	LENGERR

- **Achtung: Das gilt nur für CICS ESA = 2.3.0**

EXEC CICS GETMAIN CICS TS

- EXEC CICS GETMAIN SET(R10) FLENGTH(FWORT) INITIMG(' ') **BELOW**
- Mit diesem Command wird Storage aus der CDSA/SDSA/UDSA bzw. ECDSA/ESDSA/EUDSA angefordert.
- Es gibt **kein** GETMAIN für RDSA/ ERDSA!
- Max. Größe: DSA oder EDSA Limit. Fehler: **NOSTG** abfragen

	AMODE24	AMODE31
FLENGTH +BELOW	DSA	DSA
FLENGTH	DSA	EDSA
LENGTH	DSA	DSA

EXEC CICS GETMAIN ...

- USERDATAKEY oder CICS DATAKEY **überschreibt** die TASK DATAKEY Option aus RDO.

Ohne SHARED Option	Ohne SHARED Option	Ohne SHARED Option
No DataKey Option	USERDATAKEY	CICSDATAKEY
Je nach TASK-DATAKEY in RDO	User Storage	CICS Storage
UDSA/EUDSA oder CDSA/ECDSA	UDSA/EUDSA	CDSA/ECDSA

EXEC CICS GETMAIN mit Shared Option

- **SHARED**

Mit SHARED Option	Mit SHARED Option	Mit SHARED Option
No DataKey Option	USERDATAKEY	CICSDATAKEY
Je nach TASK-DATAKEY in RDO	User Storage aus:	CICS Storage:
SDSA/ESDSA CDSA/ECDSA	SDSA/ESDSA	CDSA/ECDSA

- Achtung: Shared Storage wird **nicht** automatisch bei Task Ende freigegeben, auch **nicht** bei Abend der Task.

Agenda Teil 2 – H. J. Ebert

- CLER: RPTSTG dynamisch an- und abschalten
- ‚Abbreviated‘ CEMT I xxx auf der Konsole (CEMT-)
- DMF-Funktionalität für das Sammeln von Job Accounting Daten
- Tipps zum CPU Monitoring
- Wahl der CPU-Geschwindigkeit
- Wahl der Platten-Peripherie

CLER Transaktion mit LE/VSE 1.4.3

- Die CLER Transaktion, verfügbar mit LE/VSE 1.4.3 (VSE/ESA 2.7), hat eine sehr interessante Funktion: RPTSTG dynamisch an- und abschalten
 - Ø Im Test kann man mit CLER auf einfachste und schnellste Weise den Speicherbedarf speziell für HEAP und STACK ermitteln
 - Ø Nach dem Test PRTSTG wieder auf OFF setzen

```

CLER                                05/02/04                                11:37:07
ABPERC: ( NONE )                   STORAGE: ( 00 , NONE , NONE , 0 )
ABTERMENC:( ABEND )                TERMTHD: ( TRACE, MSGFL , 0 )
ALL31: ( ON )                      TRAP: ( ON, MAX )
CHECK: ( OFF )                     TERMTHDACT LSTQ  OPTIONS :
DEBUG: ( OFF )                     CLASS: ( L )      DISPOSITION: ( D )
HEAP: ( 4096 , 4080 , ANYWHERE , KEEP , 4096 , 4080 )
RPTOPTS: ( OFF )
RPTSTG: ( OFF ) <-----> ( ON )
STACK: ( 4096 , 4080 , ANYWHERE , KEEP )

```

•

„Abbreviated‘ CEMT I xxx auf der Konsole

- Mit dem APAR PQ84170 / PTF UQ85123 wird bei Abfrage von 1 Ressource verkürzter Output auf der Konsole angezeigt (PTF UQ65322 ist PREREQ!)
- Eingabeformat für ‚abbreviate‘: CEMT-INQ xxxx(yyyy)

msg f5,data=cemt i trans(vsam)

Transaction(VSAM)
 Priority(001)
 Program(VSAMTEST)
 Tclass(DFHTCL00)
 Status(Enabled)
 Purgeability(Notpurgeable)
 Prfile(DFHCICST)
 Taskdatakey(Udatakey)
 Taskdataloc(Below)
 Brexit(DFHWBLT)
 Facilitylike()

msg f5,data=cemt-i trans(vsam)

Tra(VSAM) Pri(001) Pro(VSAMTEST) Tcl(DFHTCL00) Ena
 Prf(DFHCICST) Uda Bel Bre(DFHWBLT)

Job Accounting im VSE/ESA mit DMF

- Job Accounting ist eine sinnvolle Messmethode für jede Installation
 - Ø Für Abrechnung des Ressourcen-Verbrauchs
 - Ø Für Performance-Untersuchungen
 - Ø Für Kapazitätsplanungen
- SYS JA=YES in IPLPROC zwingend erforderlich
- In ICCF-Lib 59 gibt es seit langem den Job SKJOBACC
 - Ø Druckt am Ende jeden Jobsteps eine Seite mit JA-Informationen
 - Ø Nachteil: keine konsolidierten Daten, schlecht maschinell bearbeitbar
- Neue Methode: JA-Routine über DMF-Recording (Data Management Facility)
 - Ø Sammelt die Daten über einen DataSpace, Sicherung auf VSAM
 - Ø Selektive Auswertung pro Job, Jobstep, Tag etc.
 - Ø Vergleich zwischen unterschiedlich schnellen CPUs möglich

Job Accounting im VSE/ESA mit DMF...

- Jobs für DMF-Recording sind vorhanden - einfach ausprobieren
 - DFHDMFSP abschicken, erstellt die DMF Kontrollinformationen
 - SKDMFST startet DMF-Partition (PRD1.BASE in LIBDEF wegen DFHDFx- Messages)
 - SKJADACC katalogisiert \$JOBACCT.PHASE
 - SKJOBDMF umwandeln/katalogisieren der Report-Job-Phase
 - SKJADOFF entlädt DMF-VSAM-Datei in neue ESDS-Datei, vorher SETDMF FLUSH und SETDMF SWITCH ausführen
 - Skeleton SKJADPRT anpassen und abschicken, erstellt die Reports (SORT required)
- Nachstehend ein Beispiel-Output dieser JA-DMF-Routine
(1 Zeile pro Job oder Jobstep)

JOBNAME	ST ID	DD.MM.YYYY	START	STOP	DURATION	CPU + OVRHD	=	TCPU	IO	IO/TCPU(sec)
DFH0STAT	2 BG	24.10.2002	10:02:10	10:02:19	000:00:09	1.270	0.173	1.443	22724	15781

...

...

CPU Monitoring

- **CPU Monitoring - das Monitoring Intervall nicht zu hoch ansetzen**
 - ∅ Monitoring-Intervall \leq 1 Minute
- **Was ist zu hoch: Intervalle größer 1 Minute können selbst für Batch-Anwendungen zu hoch sein**
 - ∅ Die Werte werden zu stark geglättet
- **Dieses Problem gilt noch stärker für Online-Benutzer**
 - ∅ Subsecond-Responsetime und 15 Minuten-Intervalle passen nicht zusammen
- **Setzen des z/VM-Intervalls für FCON/ESA**
 - ∅ MONITOR EVENT INTERVAL 60 SECONDS (ist gleich dem Defaultwert)
- **Fehler beim Intervall ist eine häufige Quelle für inkorrektes Capacity Planning**
 - ∅ 'Garbage in, garbage out'

Wahl der CPU-Geschwindigkeit

- **Schnellstmöglicher Einzelprozessor ist erste Wahl im VSE/ESA**
 - ⊘ Dies gilt ganz besonders für Online-Anwendungen
 - ⊘ Auch Batch-Anwendungen profitieren in der Regel davon
- **Horizontales Wachstum (noch ein Prozessor und noch einer...) ist aus technischer Sicht zu vermeiden**
 - ⊘ Eine Partition (Batch/Online/DB2) kann nur einen Prozessor nutzen
 - ⊘ Unvermeidlicher Overhead bei mehr Prozessoren
- **Können sich mehrere CPU-intensive Anwendungen einen Prozessor teilen?**
 - ⊘ Mit SET SHARE (z/VM) und PRTY SHARE (VSE/ESA) lassen sich Anwendungen sehr gut ausbalancieren
- **Vertikales Wachstum ist die beste Lösung in allen mir bekannten Installationen gewesen**

Ausbalancieren von Anwendungen

- **Automatisieren von SET SHARE .. HARDLIMIT im z/VM mit FCON/ESA**
 - ⊘ Definieren eines Schwellwerts (z. B. 95% CPU ohne I/O in 1 Minute) in \$PROFILE
 - ⊘ FCUSRLIMIT userid %CPU 95 01/01
 - ⊘ FC PROCESS ERRMSG * ,FCXUSL317A' EXEC procname PSSARG
 - ⊘ Procname = REXX-Prozedur, die folgenden Command absetzt:
CP SET SHARE userid ABSolute 60% LIMITHard
 - ⊘ Achtung: Das harte Limit ausschalten, wenn nicht mehr notwendig, denn das Limit ist wirklich hart und wirkt ohne Ansehen der Person

- **PRTY SHARE im VSE/ESA**
 - ⊘ Alle Partitions, die die Balance stören können, in eine ‚Balanced Group‘ stellen
 - ⊘ Keine dieser Partitions kann die anderen unangemessen behindern
 - ⊘ Den SHARE-Anteil nach Erfahrung optimieren
 - PRTY Y,P,C,V,BG,FB,FA,F9,F8,F7,F6,F4,U,X,W,Z,F2=F5,F3,R,F1
 - SHARE F2= 300, F5= 200

Wahl der Platten-Peripherie

- Alle CPUs warten gleich lang auf I/O, nur zu unterschiedlich hohen Kosten
- Parallel Channel - ESCON - FICON
 - ∅ Parallel Channel ca. 3 MBytes/Sec
 - ∅ ESCON ca. 12-16 MBytes/Sec
 - ∅ FICON ca. 50-60 MBytes/Sec
- Größe des Cache
 - ∅ READ-Cache, ergänzt durch z/VM Minidisk Cache
 - ∅ WRITE-Cache, davon kann es nicht genug geben
- Performance der Plattensysteme
 - ∅ ESS F20 gut, ESS 800 und ESS 800 Turbo (erst ab 7-8 TerraBytes) besser

Wahl der Platten-Peripherie...

- Vergleich ESS F20 versus 800

	F20	800
— Cache-Größe in GB	8, 16, 24, 32	8, 16, 24, 32, 64
— NVS-Größe (Write Cache) in MB	384	2048
— FICON Host-Adapter GBit	1	2
— Prozessorleistung Faktor	1	2
— Interne Bandbreite Faktor	1	> 1,5
— Peer-Peer Remote Copy-Faktor	1	>= 2
— FlashCopy Copy-Faktor	1	> 2



ESS 800 souverän schneller als die F20