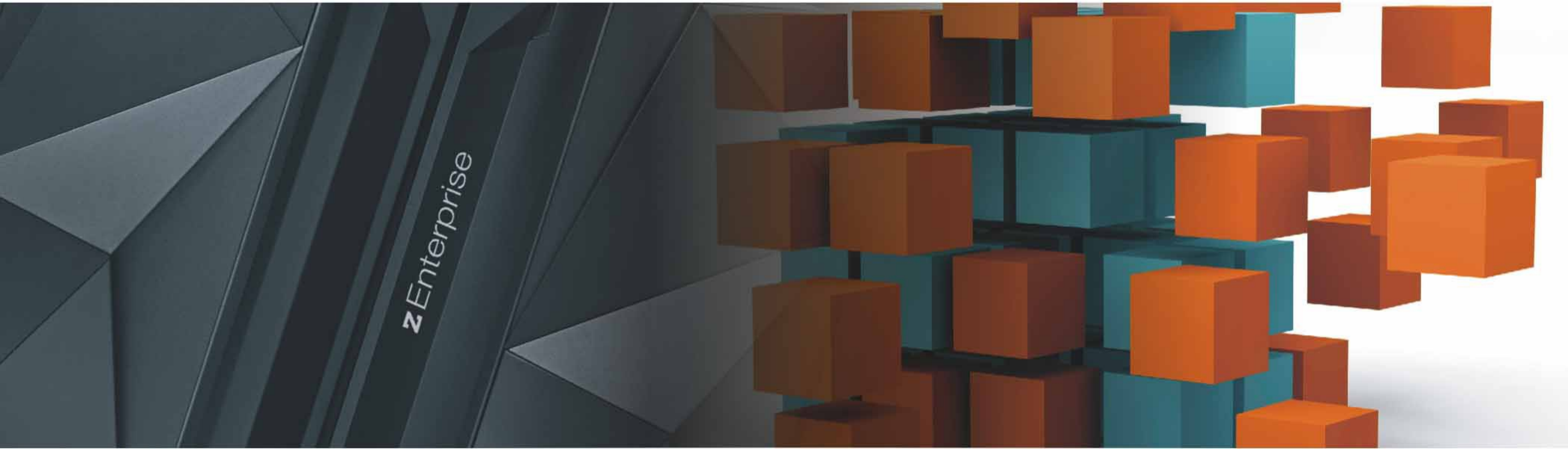
GSE Frühjahrstagung 2013

Dienstag, 23. April 2013

Natalie Salm und Jens Remus

z/VSE Supervisor Highlights



Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Notice Regarding Specialty Engines (e.g., zIIPs, zAAPs and IFLs)

- Any information contained in this document regarding Specialty Engines ("SEs") and SE eligible workloads provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g., zIIPs, zAAPs, and IFLs). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the “Authorized Use Table for IBM Machines” (“AUT”) provided at: http://www.ibm.com/systems/support/machine_warranties/machine_code/aut.html
- No other workload processing is authorized for execution on an SE.
- IBM offers SEs at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

Agenda

- 64-bit Virtual
  - 64-bit Virtual Support
  - 64-bit I/O Support
  - System Dump Enhancements
- Sub-Capacity Software Pricing
  - Capacity Measurement Tool (CMT)
  - Sub-Capacity Reporting Tool (SCRT)
  - Enhancements to CMT and SCRT
  - Hints & Tips
- Query Virtual Server (QVS) API

z/VSE 5.1: 64-bit virtual overview

- Available with z/VSE 5.1
- Support 64-bit virtual addressing
- 64-bit area can be used for data only
  - No instruction execution above the bar
- **z/OS affinity**: APIs (IARV64 services) – to manage memory objects – compatible with z/OS
  - Private Memory Objects for use in one address space
  - Shared Memory Objects to be shared among multiple address spaces
- Maximum VSIZE still limited to 90 GB
- Access to memory objects via IARV64 services and switch into AMODE 64 (SAM 64)
- Advantages:
  - Eases the access of large amounts of data
  - Reduces complexity of programs
  - Chosen design has no dependencies to existing APIs, minor impact on existing system code

64-bit virtual – Naming Conventions

- Area above 2GB – private or shared = **Extended Area**
- Area above 2GB – private area = **Extended Private Area**
- Area above 2GB – shared area = **Extended Shared Area**

- **The (2GB) bar**: a line that separates the address space into storage below 2GB (below the bar) and above 2GB (above the bar)

- 64-bit general purpose registers = **8 byte registers**
  - High order half = 0-31 bits of register
  - Low order half = 32-63 bits of register

Memory Objects

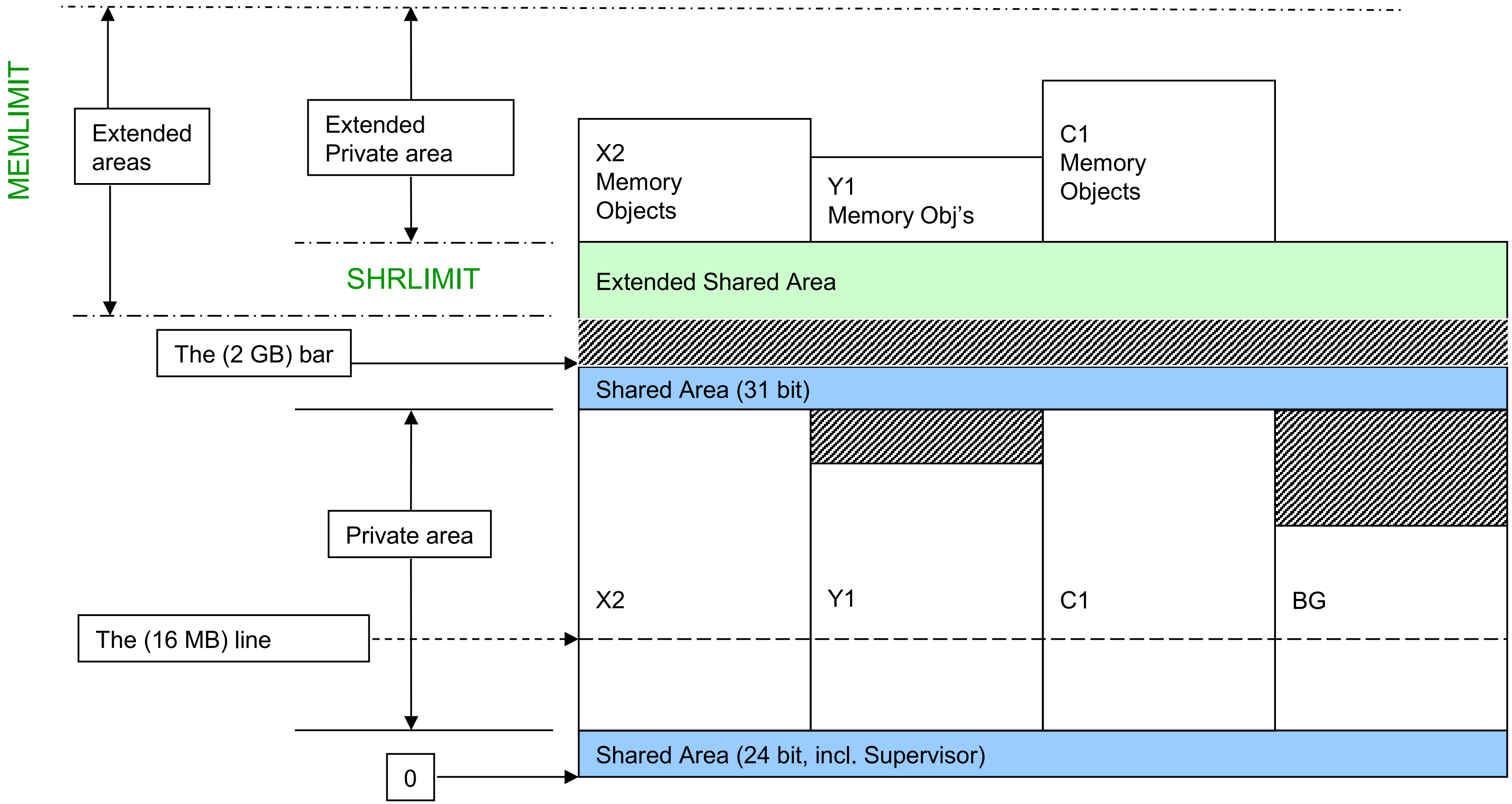
- “Chunks” of virtual storage obtained by a program
- Allocated above the 2 GB bar
- Contiguous range of virtual addresses
- Begins on a 1 MB boundary and is multiple of 1 MB in size
- Two types of memory objects:
  - Private Memory Objects are created within an address space
  - Shared Memory Objects are created within the Extended Shared Area
- Support 64-bit virtual addressing

64-bit virtual – Define System Limits

- SYSDEF statement to define the limits for memory objects
  - Before IARV64 macro can be used
  - SYSDEF MEMOBJ,MEMLIMIT=,SHRLIMIT=,LFAREA=,LF64ONLY=
    - MEMLIMIT – maximum virtual storage available for memory objects
    - SHRLIMIT – maximum virtual storage available for Shared Memory Objects
    - LFAREA – maximum real storage to fix Private Memory Objects
    - LF64ONLY – NO|YES – memory objects are fixed in 64-bit frames only
  - Example:

```
SYSDEF MEMOBJ,MEMLIMIT=1G,SHRLIMIT=500M,LFAREA=10M,LF64ONLY=YES
AR 0015 1I40I  READY
```

64-bit virtual – Address Space Layout



IARV64 Macro

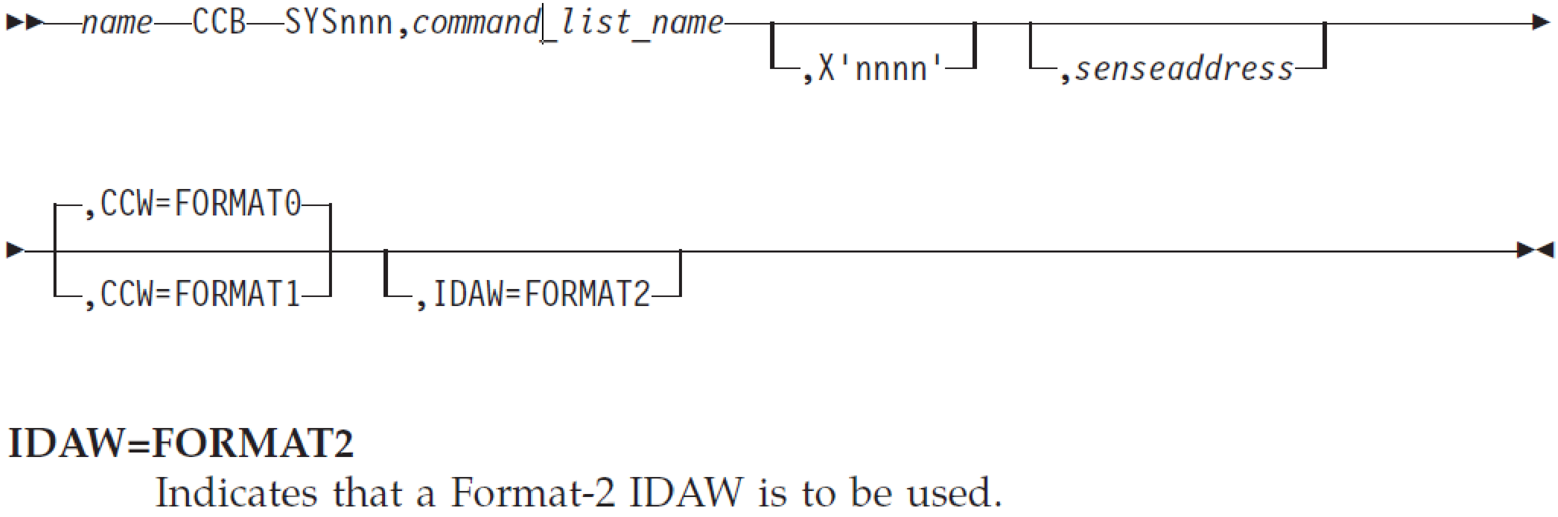
- IARV64 macro – ported from z/OS – provides services to
  - Create and free storage areas above the bar
  - Manage the physical frames behind the storage
- Programs use the IARV64 macro to obtain memory objects
- Service (IARV64 REQUEST=):
  - GETSTOR – create a Private Memory Object
  - DETACH – free one or more memory objects
  - GETSHARED – create a memory object that can be shared across multiple address spaces
  - SHAREMEMOBJ – request that the specified address space be given access to be a Shared Memory Object
  - PAGEFIX – fix pages within one or more Private Memory Objects
  - PAGEUNFIX – unfix pages within one or more memory objects

z/VSE 5.1.1 – 64-bit I/O Support

- GA: 12/19/2012 via PTF UD53915/UD53917/UD53916 (APAR DY47619)
- I/O requests for Private Memory Objects only
  - No I/O support for Shared Memory Objects
- EXCP only
  - LIOCS not supported
- DASD (ECKD) only
  - No 64-bit support for Tape, FBA/SCSI
- Advantages:
  - Eases allocation of large I/O buffers
    - e.g. instead of using and managing data spaces
  - Performance improvement
    - Large I/O buffers
    - Less complexity of programs
  - Chosen design has no dependencies to existing APIs
    - Minor impact on existing system code

64-bit I/O Request: User CCB for EXCP

- CCB must remain below the bar
- CCB macro is extended by a new parameter:



```
►►—name—CCB—SYSnnn,command_list_name—┬─────────┬──┬───────────────┬──────────►
                                     └─,X'nnnn'─┘  └─,senseaddress─┘

   ┌─,CCW=FORMAT0─┐
►──┼──────────────┼──┬───────────────┬───────────────────────────────────────►◄
   └─,CCW=FORMAT1─┘  └─,IDAW=FORMAT2─┘
```

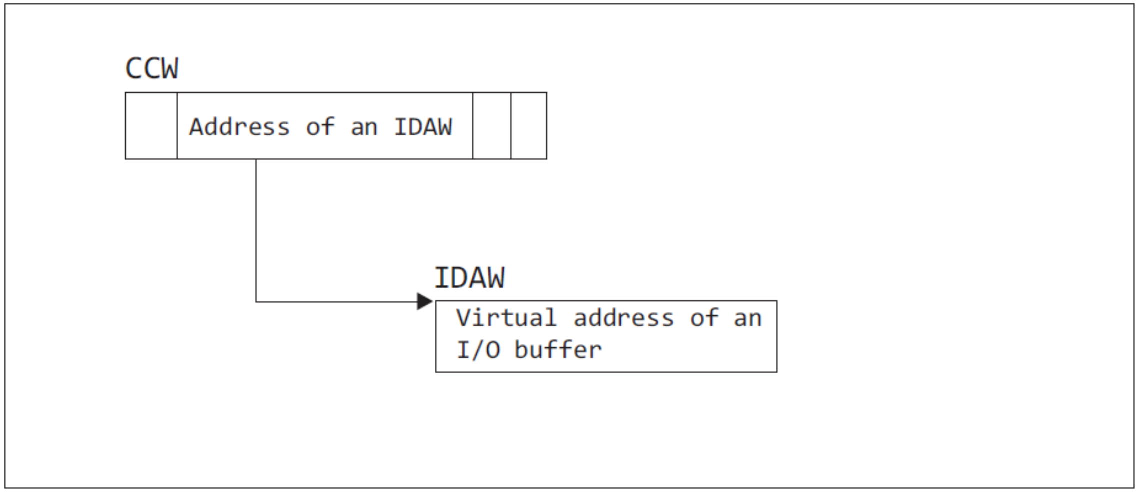
IDAW=FORMAT2

Indicates that a Format-2 IDAW is to be used.

- Format-1 IDAW not supported
- 2k-IDAW-control not supported

64-bit I/O Request: User CCW

- CCWs must remain below the bar
- CCW can either be Format-0 or Format-1
- Indirect-Data-Addressing (IDA)-bit set
- CCW points to a single Format-2 IDAW containing a virtual 64-bit address



How to Perform a 64-bit I/O Request

1) Create a Private Memory Object using an IARV64 REQUEST=GETSTOR

→ The storage key of the PMO must be equal to the partition key

2) Prepare CCWs providing a 64-bit virtual address by specifying (a) the IDA-bit and (b) a data address that points to a single virtual Format-2 IDAW

→ The CCW can either be Format-0 or Format-1

3) Create a CCB indicating Format-2 IDAW by using the CCB macro with the parameter IDAW=Format-2

4) Issue an EXCP request using the EXCP macro in AMODE 24 or AMODE 31

→ All parameter lists, control blocks, IDAWs, CCWs, and EXCP appendage routines must remain 31-bit addressable

When the I/O operation is complete, your program can then free the memory object using the IARV64 DETACH request.

z/VSE 5.1 Enhancements – System Dump for Memory Objects

- Announcement: 04/02/2013

- Program initiated dump of memory objects
  - Introduced with z/VSE 5.1.0
  - Using SDUMPX macro

- System dump for memory objects
  - New JCL options introduced with z/VSE 5.1 enhancement

- JCL will allow to specify:
  - OPTION NOMODUMP | MODUMP
  - STDOPT MODUMP=NO | YES

OPTION MODUMP and STDOPT MODUMP=YES

- Enables the system to dump memory objects in case of an abnormal program end

- The ABEND routine checks the following requirements:
  - The program runs in 64-bit addressing mode
  - At least one general register contains a 64-bit address within the range of a memory object

- Memory object dump
  - The data to be dumped is -/+ 4K of the memory object address
  - Output is written to SYSLST

QUERY STDOPT and QUERY OPTION

- New options supported by QUERY STDOPT and QUERY OPTION commands:

```
QUERY OPTION,BG
AR 0015 NOACANCEL  ACL        ALIGN      NODECK     NODSPDUMP  PARTDUMP
AR 0015 ERRS       NOIGNLOCK  NOJCANCEL  NOLINK     LIST       NOLISTX
AR 0015 LOG        NOLOGSRC   NOMODUMP   NORLD      SADUMP=000 NOSCANCEL
AR 0015 NOSLISKIP  SUBLIB=AE  NOSYM      SYSDUMP    NOSYSDUMPC NOTERM
AR 0015 SXREF      60C
AR 0015 1I40I  READY
```

```
QUERY STDOPT
AR 0015      ACL=YES     DUMP=PART   LINES= 56  ACANCEL=NO  SADMPSMO=NO
AR 0015      LOG=YES     ERRS=YES    LISTX=NO   CHARSET=60C SYSDUMPC=NO
AR 0015      RLD=NO      LIST=YES    SXREF=YES  DSPDUMP=NO
AR 0015      SYM=NO      TERM=NO    HCTRAN=YES  JCANCEL=NO
AR 0015     DATE=MDY     XREF=NO    MODUMP=NO   SCANCEL=NO
AR 0015     DECK=NO     ALIGN=YES   SADUMP=000  SYSDUMP=YES
AR 0015 1I40I  READY
```

STDOPT SADMPSMO

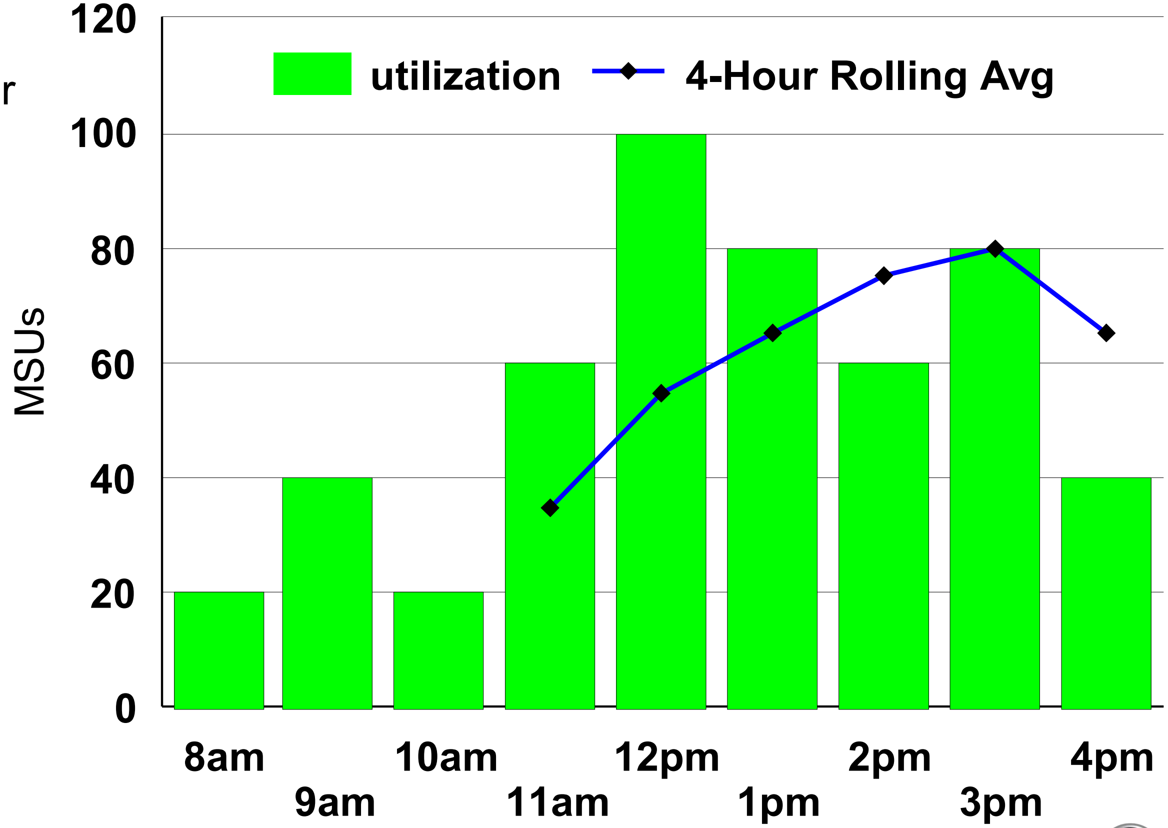
- STDOPT/OPTION SADUMP=(n,m,o) introduced with z/VSE 5.1.0
  - Controls whether private memory objects to be included in a stand-alone dump
  - Dump of shared memory objects were included by default

- New STDOPT to include or exclude shared memory objects in a stand-alone dump
  - STDOPT SADMPSMO=NO | YES
  - Cannot be overwritten by a corresponding OPTION statement

Sub-Capacity Software Pricing
Four Hour Rolling Average (4HRA)

- Work is measured in Service Units (SU)
- Capacity and utilization is measured in Millions of Service Units (MSU) per hour

Capture the 4-hour rolling average of utilization for each interval in the month

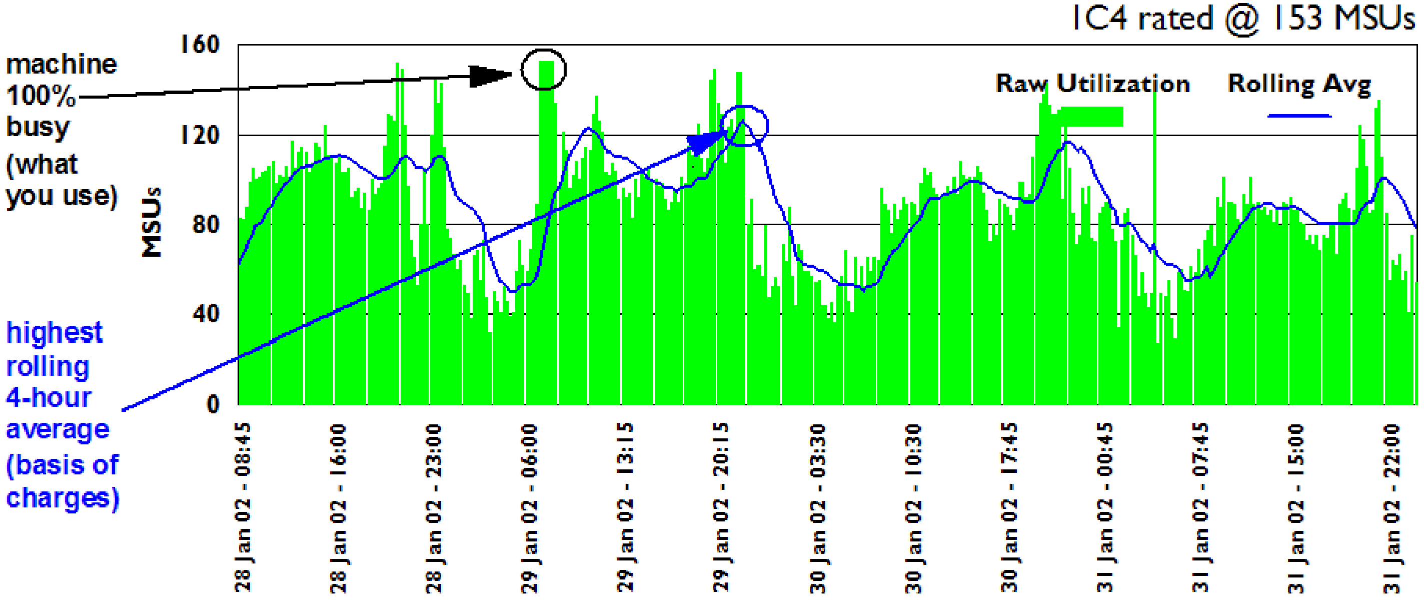


4-Hour Rolling Average

11 am	(8,9,10,11):	35 MSUs
12 pm	(9,10,11,12):	55 MSUs
1 pm	(10,11,12,1):	65 MSUs
2 pm	(11,12,1,2):	75 MSUs
3 pm	(12, 1, 2, 3):	80 MSUs
4 pm	(1, 2, 3, 4):	65 MSUs

Sub-Capacity Software Pricing
Four Hour Rolling Average (4HRA) (cont.)

Sub-Capacity Software Pricing



Rolling 4-Hour Average utilization smoothes out peaks in raw utilization.
Allows for varied peaks & bases Software charges on more moderate measure.

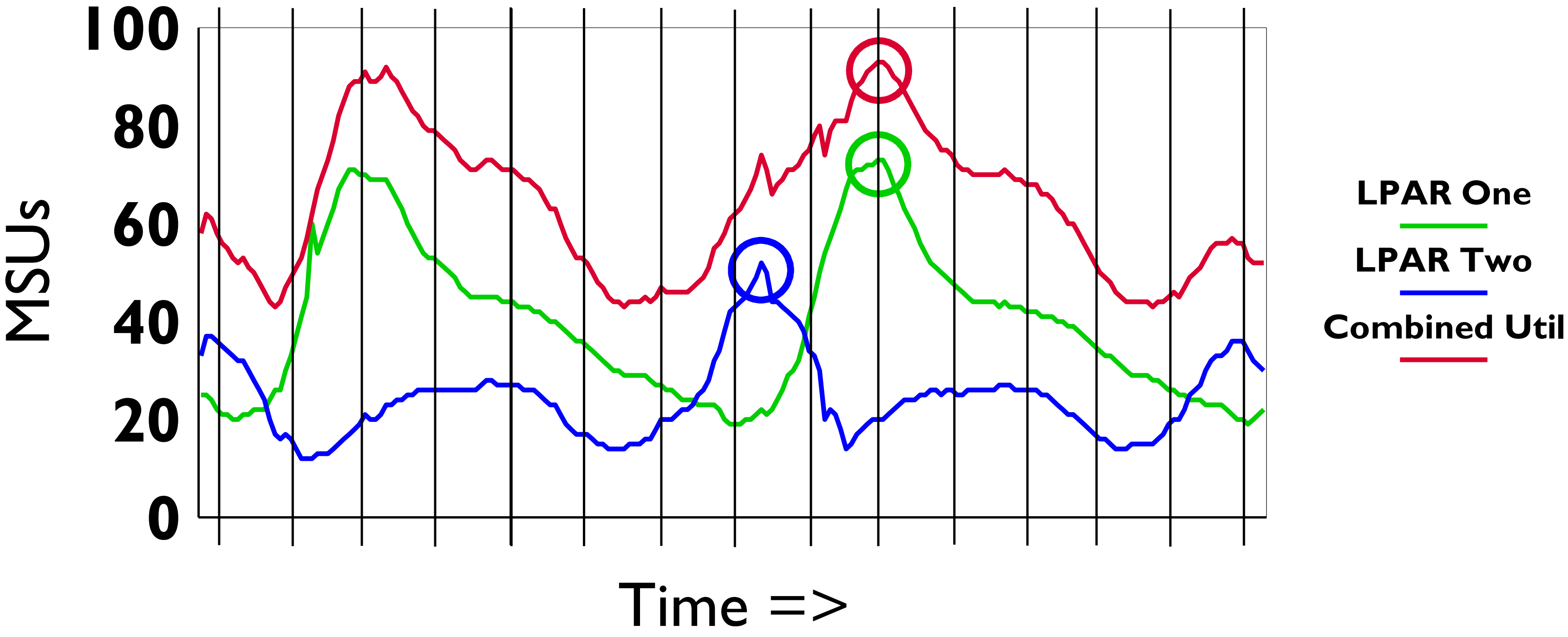
Sub-Capacity Software Pricing
Four Hour Rolling Average (4HRA) (cont.)

Peak LPAR1 = 73

Peak LPAR2 = 52

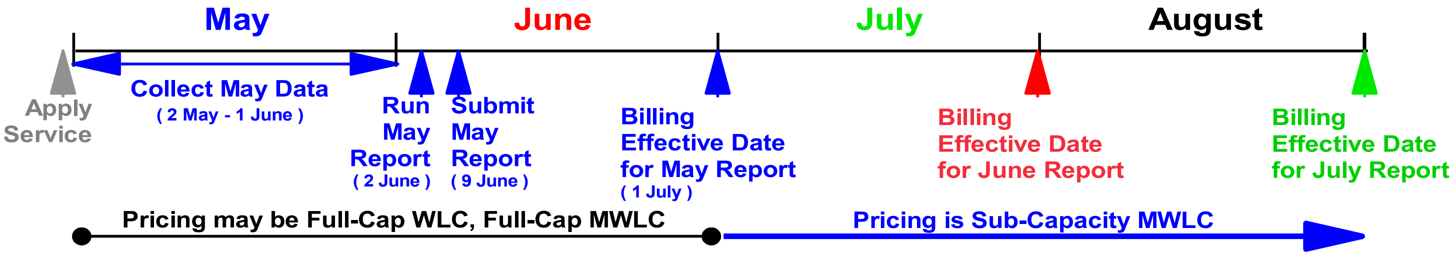
Combined Peak = 93

(<u>NOT</u> PeakLPAR1 + PeakLPAR2 = 125)



Sub-Capacity Software Pricing for z/VSE

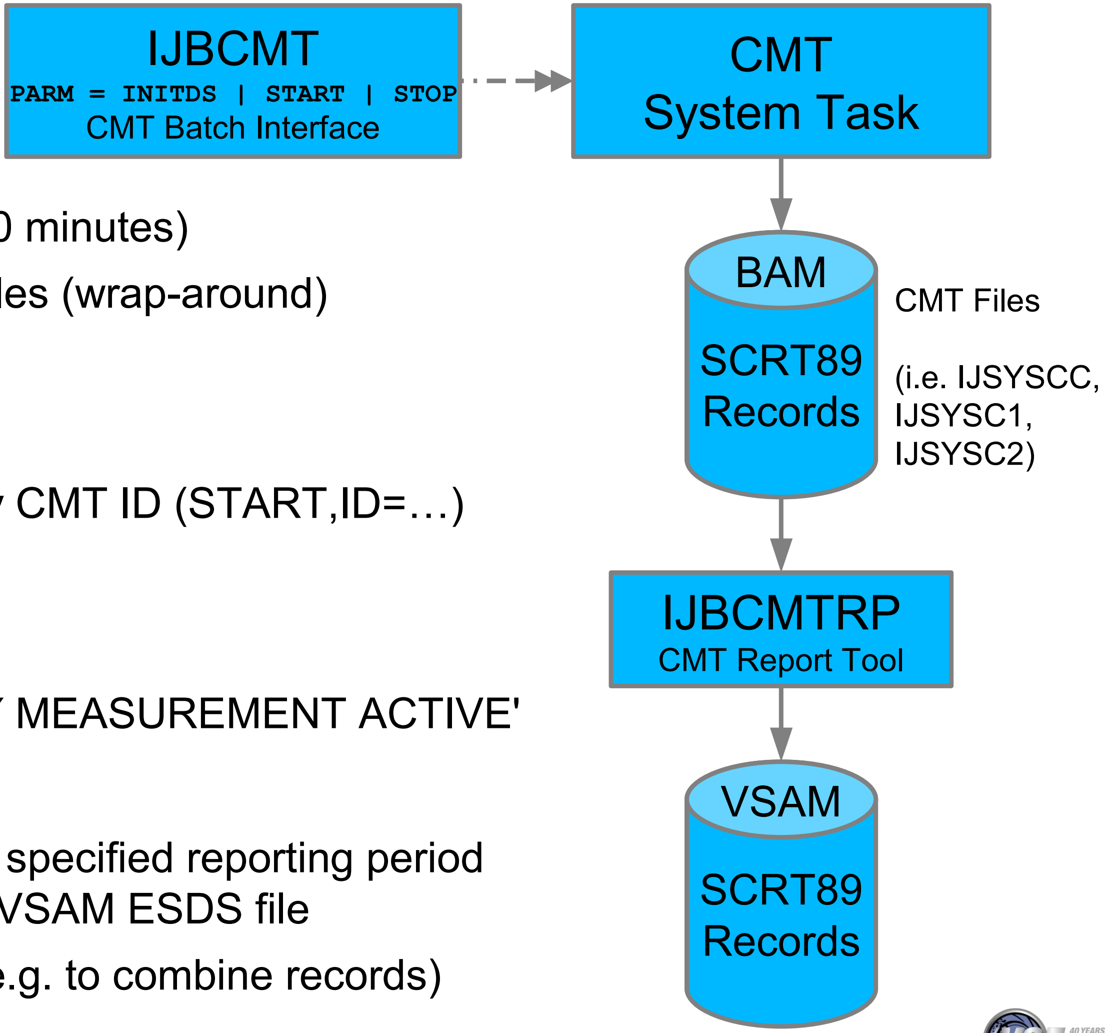
- Minimum HW & SW requirements:
  - System z9 (z9 and z10: not model A01)
  - z/VSE 4.1
  - z/VM 5.2 if running under z/VM
- Process
  - Data **collection period**: 2nd of a month to 1st of the following month
  - Data **submission period**: 2nd to 9th of a month following data collection


May
June
July
August
Apply Service
Collect May Data
(2 May - 1 June)
Run May Report
(2 June)
Submit May Report
(9 June)
Billing Effective Date for May Report
(1 July)
Billing Effective Date for June Report
Billing Effective Date for July Report
Pricing may be Full-Cap WLC, Full-Cap MWLC
Pricing is Sub-Capacity MWLC

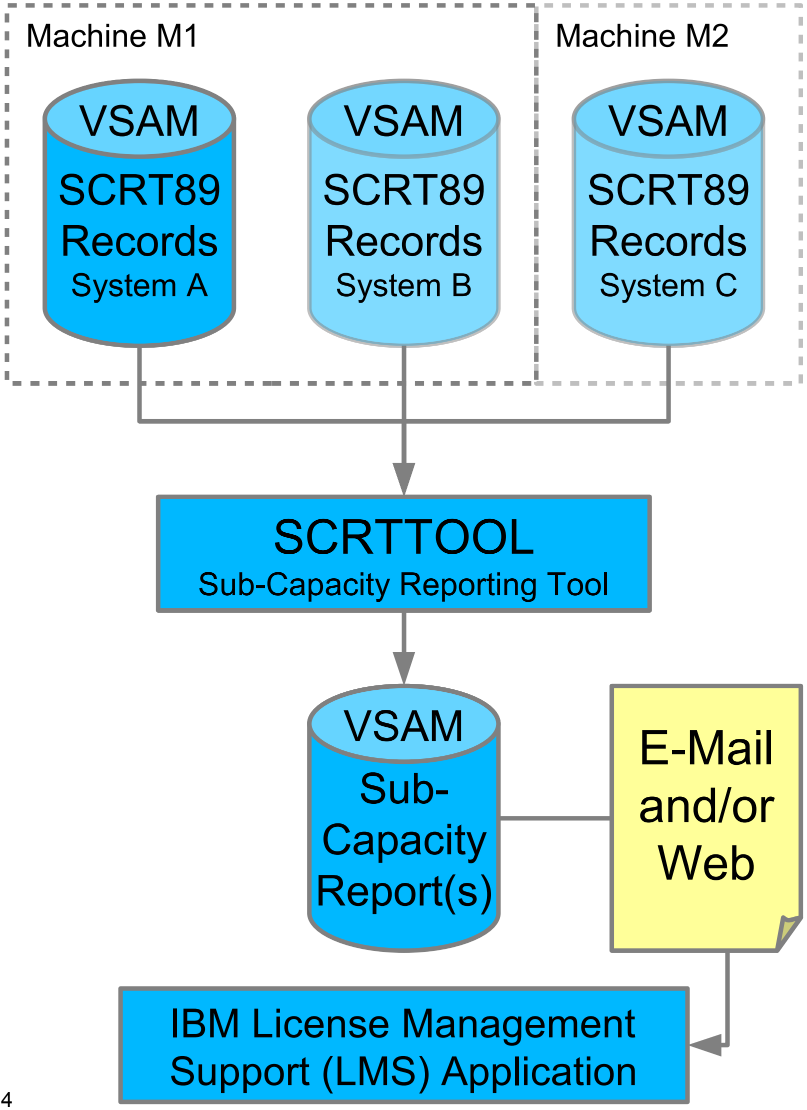
- Reporting Requirements
  - Must report on **all** LPARs and z/VM guests (e.g. production, test, development, …)
  - 95% overall data collection
  - Default (i.e. worst case) are full-capacity prices

z/VSE Capacity Measurement Tool (CMT) Overview

- CMT System Task
  - Measures used CPU capacity
  - Generates a SCRT89 record every measurement interval (30 minutes)
  - Writes the record to the CMT files (wrap-around)
- CMT Batch Interface (IJBCMT)
  - Initialize CMT files (INITDS)
  - Start measurement and specify CMT ID (START,ID=…)
  - Stop measurement (STOP)
- SIR AR Command
  - Query CMT status: 'CAPACITY MEASUREMENT ACTIVE'
- CMT Report Tool (IJBCMTRP)
  - Extracts SCRT89 records for a specified reporting period from the CMT BAM files into a VSAM ESDS file
  - Either overwrites or appends (e.g. to combine records)



Sub-Capacity Reporting Tool (SCRT) Overview



- Sub-Capacity Reporting Tool (SCRT)
  - Processes SCRT89 records from multiple systems (and optional from multiple machines) together
  - Generates a Sub-Capacity Report ("SCRT report") for each machine
- The SCRT report is in Comma Separated Values (CSV) format; can be viewed in any spreadsheet application (e.g. MS Excel, OO.org Calc)
- Submission process to IBM License Management Support (LMS) application:
  - Download report to workstation for review and editing, then use either e-mail or web for submission to LMS
  - Send by e-mail to LMS, then manage, edit, and confirm via web

Enhancements to z/VSE Capacity Measurement Tool (CMT) and Sub-Capacity Reporting Tool (SCRT)

- z/VSE 4.2: Support for reporting of z/VSE system image capacities of less than one MSU
  - APARs for z/VSE 4.1 and z/VSE 4.2 (DY47027 and DY47029)
- z/VSE 4.3: Support for z196, z114, zEC12, and Static Power Save Mode
  - APARs for z/VSE 4.1 and z/VSE 4.2 (DY47110 and DY47111)
  - Static Power Save Mode only available on z196 and zEC12
  - Includes support for CMT data files that exceed 32767 tracks on ECKD disks in CMT Report Tool (IJBCMTRP)
  - CMT message 0S48 is highlighted and retained on the console (z/VSE 4.3 onwards)
- z/VSE 5.1: Support for date and time offsets in CMT Report Tool (IJBCMTRP)
  - APARs for z/VSE 4.1, z/VSE 4.2, and z/VSE 4.3 (DY47196, DY47194, and DY47197)
  - Useful in case of e.g. end-of-year processing tests
- Support for z/VSE V4 and V5 on one machine with SCRT V20.1.0
- SCRT V21.2.0 released on 04/10/2013

z/VSE Capacity Measurement Tool (CMT)
Hints & Tips

- CMT files (IJSYSCC, IJSYSC1, IJSYSC2):
  - Must not be shared among multiple systems (for write); each system requires its own set of files
  - Can reside on a shared disk as long as disk sharing is set up correctly:
    - IPL DLF statement (one single shared lock file)
    - IPL ADD statement with option SHR (for each shared disk)
  - Must not be relocated (neither moved on the disk nor to another disk)
  - It is recommended to stick to the default configuration values (e.g. number of tracks or blocks specified for ECKD and FBA disks)
- CMT ID (=SYSID in z/OS) must be unique among all systems and across all operating systems:
  - If every machine (=CPC) is processed on its own SCRT run, then the IDs must only be unique for each machine (this is the preferred solution)
  - If multiple machines (=CPCs) are to be processed in one SCRT run, the IDs must be unique for all machines being processed together
- SCRT User's Guide: http://www.ibm.com/systems/z/resources/swprice/subcap/scrt/

z/VSE Capacity Measurement Tool (CMT) Hints & Tips (cont.)

- Guidelines for selected NO89 products:
  [Source: Sub-Capacity Reporting Tool V21.2.0, chapter 10, page 194-195]
  - DB2 SERVER FOR VSE & VM (5697-F42)
    - When the Data Restore feature was the only DB2 feature being used on a z/VSE system, then do not assign 5697-F42 to that z/VSE native or guest system because you are already being billed separately for this feature
    - If you ran DB2 under z/VM in an LPAR without any z/VSE guests, do not assign 5697-F42 to that LPAR because you are already being billed separately for DB2 under z/VM
  - HIGH LEVEL ASSEMBLER VSE & VM (5696-234)
    - If you ran HLASM under both z/VM and z/VSE, do not assign 5696-234 to any LPAR because you are already being billed separately for HLASM under z/VM

Query Virtual Server (QVS) API

- Provides information on the system
  - CEC, LPAR, and z/VM maximum capacity in MSU/h
  - LPAR Name and ID
  - z/VM Name
- Available under z/OS, z/VM, and z/VSE
  - HLASM macros IRAQVS and SYSEVENT QVS
  - C header file IWMQVSH
- Documented in z/VSE Supervisor Calls and Internal Macros
- New with z/VSE 5.1:
  - Support for PR/SM hard-capping based on weights when running in LPAR
  - Transparent pass-trough of z/VM QVS data when running under z/VM (requires at least z/VM 5.4)

Questions?

?

Support for Standby CPUs and Logical Processor Add (z/VSE 4.3)

- Standby CPUs are not directly available for processing; they must be configured prior to use
- Standby CPUs are detected and added to the z/VSE CPU configuration at IPL-time
- Standby CPUs that are added to the LPAR at run-time using the HMC/SE task 'Logical Processor Add' are detected and added to the z/VSE CPU configuration automatically
- **`QUERY TD,<…>`** (AR command) displays the current CPU configuration including standby CPUs (z/VSE CPU states: active, inactive, quiesced, and standby)
- **`SYSDEF TD,<…>`** (AR, JCL command)
  - **`SYSDEF TD,STARTSBY=<cpuaddr>`** starts a standby CPU
  - **`SYSDEF TD,STOPSBY=<cpuaddr>`** sets an active, inactive, or quiesced CPU into standby state
  - **`SYSDEF TD,START=ALL|<cpuaddr>`** does not allow to start standby CPUs
- **`STARTSBY|STOPSBY`** due to the following considerations:
  - Behavior of existing LPAR configurations should not change
  - Standby CPUs therefore do not participate in z/VSE TD CPU balancing
  - In contrast to online (stopped or operating) CPUs, standby CPUs do not consume any LPAR CPU share, as it is divided equally among all online CPUs of an LPAR

Support for Standby CPUs and Logical Processor Add (z/VSE 4.3)
Example: QUERY TD

```
query td
AR 0015  CPU   STATUS    SPIN_TIME    NP_TIME TOTAL_TIME NP/TOT
AR 0015   00   ACTIVE            0       2525       3392  0.744
AR 0015   01   INACTIVE
AR 0015   02   STANDBY
AR 0015                 ----------------------------------------
AR 0015 TOTAL                    0       2525       3392  0.744
AR 0015
AR 0015               NP/TOT: 0.744      SPIN/(SPIN+TOT): 0.000
AR 0015  OVERALL UTILIZATION:   0%        NP UTILIZATION:   0%
AR 0015
AR 0015  CPU BALANCING:       NOT ACTIVATED
AR 0015
AR 0015  ELAPSED TIME SINCE LAST RESET:      3071099
AR 0015 1I40I  READY
```

Support for Standby CPUs and Logical Processor Add (z/VSE 4.3) Example: HMC/SE Task 'Logical Processor Add'

```
query td
AR 0015  CPU   STATUS    SPIN_TIME    NP_TIME TOTAL_TIME NP/TOT
AR 0015   00   ACTIVE            0         43      23070  0.001
[…]

AR 0030 0W03I  00002 STANDBY CPUS HAVE BEEN ADDED TO THE
CONFIGURATION

query td
AR 0015  CPU   STATUS    SPIN_TIME    NP_TIME TOTAL_TIME NP/TOT
AR 0015   00   ACTIVE            0         65      23095  0.002
AR 0015   01   STANDBY
AR 0015   02   STANDBY
[…]
```

Support for Standby CPUs and Logical Processor Add (z/VSE 4.3) Example: SYSDEF TD,STARTSBY=<cpuaddr>

```
query td
AR 0015  CPU   STATUS    SPIN_TIME    NP_TIME TOTAL_TIME NP/TOT
AR 0015   00   ACTIVE            0       2525       3392  0.744
AR 0015   01   STANDBY
[…]

sysdef td,startsby=01
AR 0015 1YH7I  NUMBER OF CPU(S)  - ACTIVE: 1  - QUIESCED: 0  -
INACTIVE: 0  -
          STANDBY: 1
AR 0015 1I40I  READY

query td
AR 0015  CPU   STATUS    SPIN_TIME    NP_TIME TOTAL_TIME NP/TOT
AR 0015   00   ACTIVE            0          1          1  1.000
AR 0015   01   ACTIVE            0          1          2  0.500
[…]
```

Support for Standby CPUs and Logical Processor Add (z/VSE 4.3) Example: SYSDEF TD,STOPSBY=<cpuaddr>

```
query td
AR 0015  CPU   STATUS    SPIN_TIME    NP_TIME TOTAL_TIME NP/TOT
AR 0015   00   ACTIVE            0         58      43697  0.001
AR 0015   01   ACTIVE            0         73      48143  0.001
[…]

sysdef td,stopsby=01
AR 0015 1YH7I  NUMBER OF CPU(S)  - ACTIVE: 2  - QUIESCED: 0  -
INACTIVE: 0  -
          STANDBY: 0
AR 0015 1I40I  READY

query td
AR 0015  CPU   STATUS    SPIN_TIME    NP_TIME TOTAL_TIME NP/TOT
AR 0015   00   ACTIVE            0          2          3  0.666
AR 0015   01   STANDBY
[…]
```

Signal Quiesce/Shutdown

- A *signal quiesce* (also referred to as *signal shutdown*) event is presented to a system
  - in LPAR, if a disruptive operation is performed by a Hardware Management Console (HMC) or Service Element (SE) for the LPAR
  - under z/VM, if a `SIGNAL SHUTDOWN` command is issued for the guest
- If the operating system is enabled for this kind of event, it is granted a grace period to perform a controlled system shutdown before the disruptive operation is executed
- `IPL SYS QUIESCE=YES` enables the support for signal quiesce in z/VSE
- z/VSE issues decision message 0W01D when a signal quiesce event occurs and it is enabled for this kind of event

```
0W01D DO YOU WANT TO CONTINUE SYSTEM SHUTDOWN (WILL BE FORCED
AFTER TIMEOUT)? REPLY 'YES' TO ENTER HARD WAIT STATE OR 'NO'
```

- Console automation programs, for example, can initiate a controlled system shutdown as a response to message 0W01D

z/VSE Supervisor Diagnosis Reference (DRM) and z/VSE Supervisor Calls and Internal Macros

- Updated manuals for z/VSE 5.1 available on the z/VSE homepage:
  - z/VSE Supervisor Diagnosis Reference
  - z/VSE Supervisor Calls and Internal Macros

http://www.ibm.com/systems/z/os/zvse/documentation/