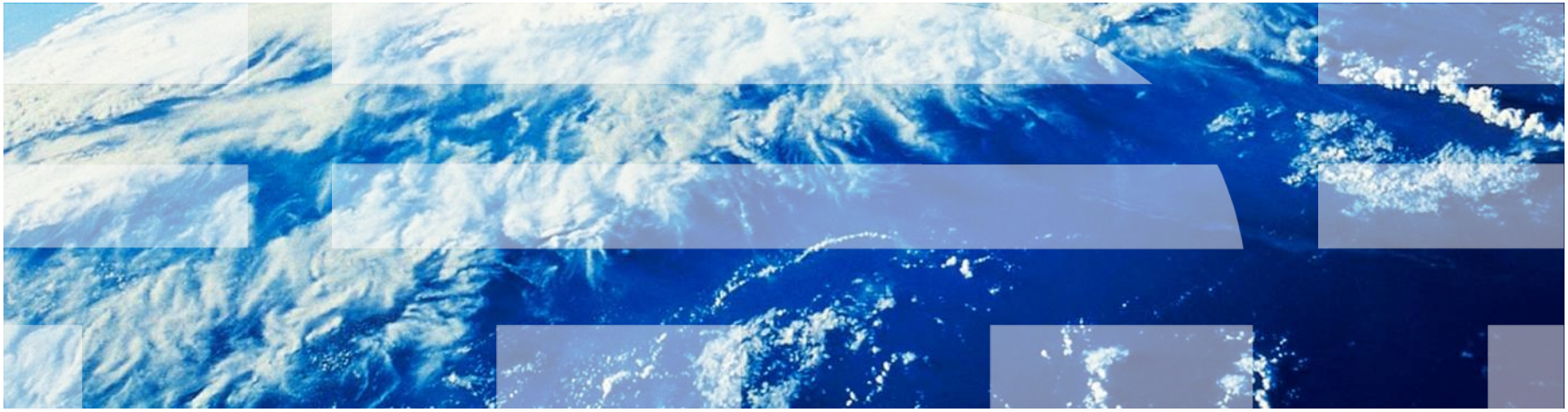
z/VM Scheduler Overview

Bill Bitner - z/VM Customer Care and Focus
bitnerb@us.ibm.com



Trademarks

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

IBM*	System z10
IBM Logo*	Tivoli*
DB2*	z10
Dynamic Infrastructure*	z10 BC
GDPS*	z9
HiperSockets	z/OS*
Parallel Sysplex*	z/VM*
RACF*	z/VSE
System z*	zEnterprise

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

OpenSolaris, 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.
INFINIBAND, InfiniBand Trade Association and the INFINIBAND design marks are trademarks and/or service marks of the INFINIBAND Trade Association.
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.

* 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.

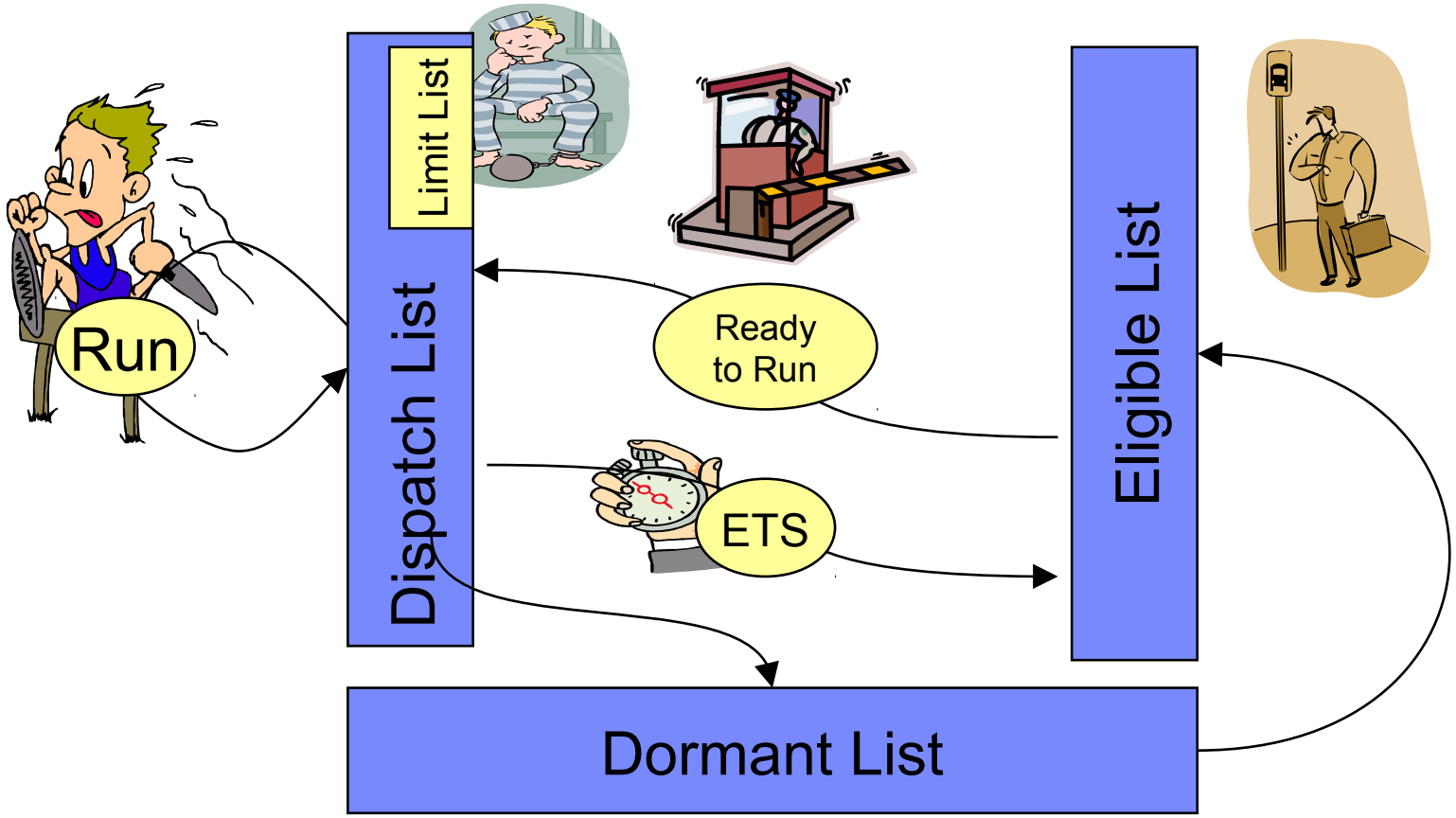
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Introduction

- Objectives
  - Provide useful information on how the z/VM Scheduler works
  - Explore some tuning methodologies
  - This presentation will not make you an expert
- Agenda
  - Background on the Scheduler
  - How things work
  - Where to find information
  - Tuning discussion

The Main Loops

- Each virtual processor is in one of the following lists:
  - Dispatch List – (D-List, in Q) users ready or near-ready to run
  - Eligible List – (E-list) Delayed here when cannot “fit” in D-List
  - Dormant List – users that are idle (from view of the scheduler)



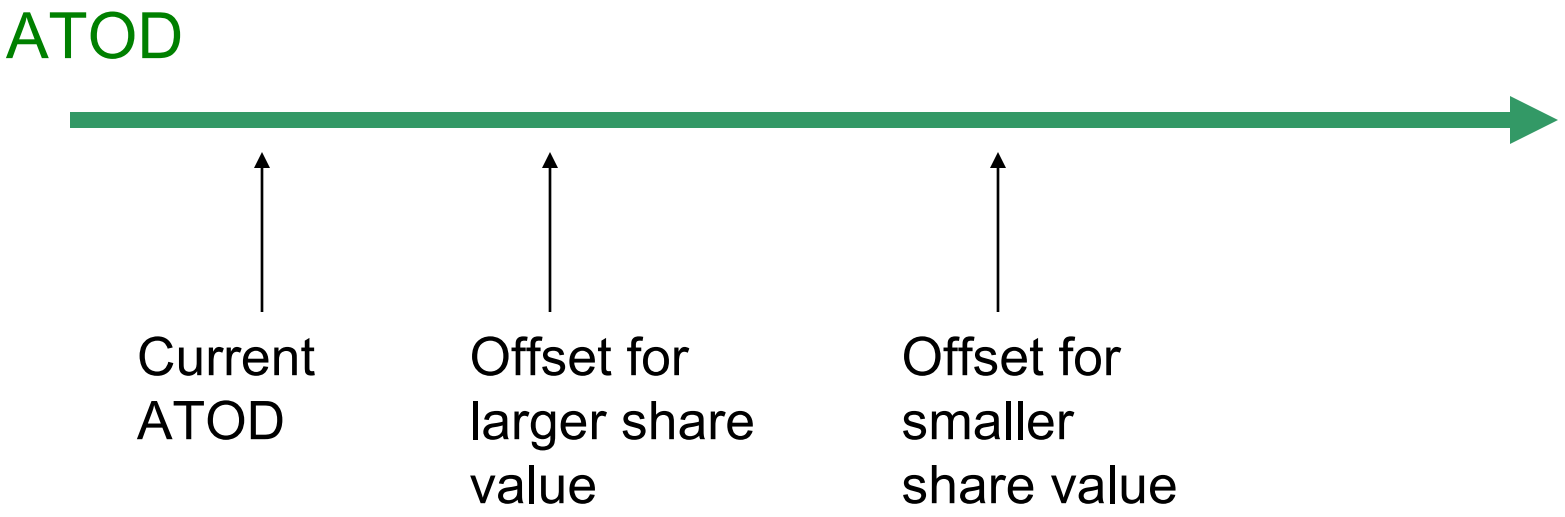
Class Structure

- Virtual Processor belongs to one Transaction Class
  - 1: vCPU with “short-running” transactions
  - 2: vCPU with “medium-running” transactions
  - 3: vCPU with “long-running” transactions
  - 0: vCPU of special virtual machines or with special considerations (QUICKDSP ON, hotshot, lockshot virtual machines)
- Elapsed Time Slice for each class
  - Class 1 ETS is dynamic with goal of keeping n% of virtual machines in class 1.
  - Class 2 ETS is 8 x C1ETS
  - Class 3 ETS is Max(48 x C1ETS, time to read in WSS)
  - Class 0 ETS is 6 x C1ETS
- When entering E-list from Dormant list, start as class 1.
- Bump up a class when ETS expires, but transaction is not complete.

Deadline Scheduling – Prioritizing Work

- Each virtual processor has a priority computed as a ‘deadline’ for when a unit of work should be completed.
- This ‘deadline’ is a time value on an artificial TOD often referred to as ATOD
- The ‘deadline’ is computed based on several factors, but the most significant is the normalized Share value
- Therefore the share setting is a big knob
- Virtual processors get ordered for dispatching based on their deadlines
- With VM64721 and SET SRM LIMITHARD CONSUMPTION, limit shares are controlled via a consumption scheduler instead of a deadline scheduler. More later.

ATOD and Deadline



Simplified offset formula used to set deadline 'offset' from current ATOD:

$$OFFSET = \frac{Minor\_TimeSlice + Previous\_TimeSlice\ Overrun}{Normalized\_Share \times Number\_PUs}$$

Routines of Interest

- HCPSCHEP: E-list priority at entry to E-list
- HCPSCHSE: Select E-list user based on priority and available resources
- HCPSCHTE: Miscellaneous transaction end processing
- HCPSCIAD: Initial D-list priority when user added to D-list
- HCPSCIDP: Re-prioritize after minor timeslice exceeded
- HCPSCJGL: Notification of growth in working set size
- HCPSCJXW: Exit from wait, arrange limit-list check.
- HCPSCJPG: Record new paging configuration values
- HCPSCJAL: Calculations for a newly added Limit-list user
- HCPSCJDL: Calculations for a newly dropped Limit-list user
- HCPSCJAA: Adjust ATOD and ATOD2
- HCPSCKDD: D-list processing when user is dropped
- HCPSCKPR: Choose a D-list user to preempt.
- HCPSCKXL: Exit a VMDBK from grouping below ATOD
- HCPSTK: Routines to manage status and actually move VMDBKs from list to list

Entry to Eligible List (HCPSCHEP)

- Determine new transaction or reason for coming to the E-list.
- Determine Elapsed time slice (VMDESLIC)
- Determine E-list priority (VMDEPRTY) based on:
  - User class
  - User resource consumption
  - System resource contention
  - Share setting
- Determine estimated core working set size (VMDCWSS)

Exit Eligible List (HCPSCHSE)

- Compute available memory (DPA) and largest user that can fit for each class.
- Selects vCPU to move to dispatch list
- Class 0 always selected
- Not selected if user class is blocked
- If user fits in memory then
  - Select if also meets LDUBUFF and DSPBUF limits
- If user does not fit in memory then
  - Select if use meets LDUBUFF and DSPBUF limits and is only user in its class
  - Select if E1 and behind by certain delay factor (pre-empt another)
  - Block class if not E1, but behind

Entry to Dispatch List (HCPSCHAD)

- Calculate D-list priority with the following factors
  - ATOD: Running timer in TOD form for user time
  - OFFSET: Used to initially offset form ATOD, based on other factors.
  - Paging Bias: Priority boost given when the transaction is being continued and pages have been stolen. One time boost.
  - IABIAS: Interactive Bias – priority improvement based on SRM IABIAS settings.
  - Hotshot: priority boost given to user with transaction in progress, but interacts with terminal.
- OFFSET

$$\frac{\text{Minor Time Slice + Previous Time Slice Overrun}}{\text{Dispatch List Share x Number of CPUs}}$$

- Minor Timeslice: Minor dispatch timeslice (SRM DSPSLICE) in TOD units
- Previous Timeslice Overrun: amount of time user exceeded previous minor timeslice
- Dispatch List Share: Factor of user's share and time spent in eligible list.

End of Minor Timeslice (HCPSCIDP)

- Calculate D-list priority much like the first time.
- Update some statistics
- If user got into the dispatch list by hotshort, then set flag to drop user.
- Do we need to limit the vCPU? Set flags as appropriate

Drop from Dispatch List (HCPSCKDD)

- Update statistics based on why dropping
- Keep track of status to help determine monitor transaction end
- Calculate resource consumption values for later
  - Estimated Core Working Set Size (WSS)
  - Paging Rate
- Record other resource stats (user and system)
- Optionally cut monitor record.

Preempt User (HCPSCKPR)

- Find user in D-list to preempt to make room for class 1 user to run.
- Preempt user if
  - Not class 0 or class 1
  - Not under the influence of a bias
  - Not last user in dispatch list of their class

QUERY Commands of Interest

```
CP QUERY SRM
IABIAS : INTENSITY=90%; DURATION=2
LDUBUF : Q1=200% Q2=200% Q3=200%
STORBUF: Q1=300% Q2=200% Q3=200%
DSPBUF : Q1=32767 Q2=32767 Q3=32767
DISPATCHING MINOR TIMESLICE = 5 MS
MAXWSS : LIMIT=9999%
...... : PAGES=999999
XSTORE : 0%
LIMITHARD METHOD: CONSUMPTION
```

QUERY Commands of Interest

```
CP QUERY QUICKDSP TCPIP
USER TCPIP    :  QUICKDSP = ON
Ready;
```

```
CP QUERY SHARE AVATAR

USER AVATAR  :  CP   RELATIVE SHARE = 90
                     MAXIMUM SHARE = LIMITSOFT RELATIVE 150
                ZAAP RELATIVE SHARE = 90
                     MAXIMUM SHARE = LIMITSOFT RELATIVE 150
                IFL  RELATIVE SHARE = 90
                     MAXIMUM SHARE = LIMITSOFT RELATIVE 150
                ICF  RELATIVE SHARE = 90
                     MAXIMUM SHARE = LIMITSOFT RELATIVE 150
                ZIIP RELATIVE SHARE = 90
                     MAXIMUM SHARE = LIMITSOFT RELATIVE 150
Ready;
```

INDICATE QUEUES EXPANDED command

```
CP INDICATE QUEUE EXP
EDLLIB14       Q3 IO  00002473/00002654 ..D. -.0217 A00
KAZDAKC        Q3 IO  00003964/00003572 .... -.0190 A02
BITNER         Q1 R00 00001073/00001054 .I.. -.0163 A01
LCRAMER        Q3 IO  00003122/00002850 ....  .0259 A00
DSSERV         L0 R   00007290/00007289 ....  .3229 A00
RSCS           Q0 PS  00001638/00001616 .I..  99999 A00
SICIGANO       Q3 PS  00000662/00000662 .I..  99999 A00
VMLINUX1       Q3 PS  00018063/00018063 ....  99999 A02
LNXREGR        Q3 PS  00073326/00073210 ....  99999 A02
VMLINUX        Q3 PS  00031672/00031672 ....  99999 A01
TCPIP          Q0 PS  00018863/00018397 .I..  99999 A02
EDLLNX2        Q3 PS  00032497/00032497 ....  99999 A01
EDLLNX1        Q3 PS  00015939/00015939 ....  99999 A02
```

Performance Toolkit – FCX145 SCHEDLOG (Option 3F)

Interval End Time	Total VMDBK in Q	<-- Users in Dispatch List ---> Q0	Q1	Q2	Q3	Lim <- Loading --> Q0	Q1	Q2	Q3	it Lst	In Eligible List --> <Loading-> E1	E2	E3	E1	E2	E3
>>Mean>>	268	2.8	2.4	4.4	258	.0	.0	.0	1.4	.0	.0	.0	.0	.0	.0	.0
10:42:00	269	2.0	4.0	19	244	.0	.0	.0	1.0	.0	.0	.0	.0	.0	.0	.0
10:43:00	271	3.0	5.0	1.0	262	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
10:44:00	268	3.0	3.0	1.0	261	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
10:45:00	261	2.0	1.0	1.0	257	.0	.0	.0	2.0	.0	.0	.0	.0	.0	.0	.0
10:46:00	271	3.0	3.0	2.0	263	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
10:47:00	274	3.0	5.0	1.0	265	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
10:48:00	267	2.0	2.0	2.0	261	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0

Class 1 Elapsed T-Slice	Sum of Abs. Shares	Sum of Rel. Shares	<----- Storage (Pages) ------> Total Consid	<----- Total WSS -----> Q0	Q1	Q2	Q3
1.720	0%	252395	66509k	1353	11512	415k	78M
1.836	0%	252433	66508k	910	9092	1819k	76M
1.798	0%	255733	66509k	1021	51097	7621	78M
1.758	0%	252267	66509k	1482	13098	59224	78M
1.847	0%	244367	66509k	1199	3565	5532	79M
1.836	0%	254934	66509k	1799	3773	62344	79M
1.921	0%	257600	66509k	1746	9956	56638	78M

Performance Toolkit – FCX154 SYSSET (Option F)

```
Initial Scheduler Settings: 2011/05/11 at 10:40:31
DSPSLICE (minor) 5.000 msec.           IABIAS Intensity    90 Percent
Hotshot T-slice  1.999 msec.           IABIAS Duration      2 Minor T-slices
DSPBUF Q1        32767 Openings        STORBUF Q1 Q2 Q3   400 % Main storage
DSPBUF Q1 Q2     32767 Openings        STORBUF Q2 Q3      400 % Main storage
DSPBUF Q1 Q2 Q3  32767 Openings        STORBUF Q3         400 % Main storage
LDUBUF Q1 Q2 Q3    100 % Paging exp.   Max. working set  9999 % Main storage
LDUBUF Q2 Q3       100 % Paging exp.   Loading user        5 Pgrd / T-slice
LDUBUF Q3          100 % Paging exp.   Loading capacity   70 Paging expos.
LIMITHARD algorithm    Deadline

Changed Scheduler Settings
Date  Time      Changed
..... ........  No changes processed
```

Performance Toolkit – FCX226 UCONF (Option 28)

Userid	SVM	Mach Mode	Flg ReO	Qck DSP	No MDC Fair	Atta-ched XSTOR	Stor Size	Reserved Pages
BILL	No	ESA	---	Off	No	0	32M	0
LINUX001	No	EME	---	Off	No	0	1024M	0
LINUX002	No	EME	---	Off	No	0	1024M	0

	<---- Virt. CPUs ----->			
Userid	Type	Aff	Def.	Ded.	Stop
BILL	IFL	On	1	0	0
LINUX001	IFL	On	1	0	0
LINUX002	IFL	On	1	0	0

	<- Share -->		<-- Max Share --->		
		%			%
Userid	SRel.	SAbs.	Limit	MRel.	MAbs.
BILL	1000	...	Hard	...	30
LINUX001	2000	...	...	...	...
LINUX002	2000	...	...	...	...

Performance Toolkit – FCX114 USTAT (Option 23)

Userid	%ACT	%RUN	%CPU	%LDG	%PGW	%IOW	%SIM	%TIW	%CFW	%IOA	%PGA	%LIM	%OTH
>System<	10	0	0	0	0	5	1	61	0	1	0	0	0
PDISRV	0	33	33	0	0	0	0	33	0	0	0	0	0
EDLWRK15	1	5	15	0	0	56	21	3	0	0	0	0	0
EDLWRK10	1	7	10	0	0	53	27	2	0	0	0	0	0
EDLWRK12	1	15	10	0	0	61	14	0	0	0	0	0	0

CPU Wait

Test Idle

Limit List Delay

Other Monitor Data

- Monitor Records:
  - Domain 2: Scheduler Domain
    - Add/Drop to D-List
    - Add to E-list
    - Add/Drop to L-List
    - System Timer Pop
    - Various changes
  - Domain 4: User Domain
    - Transaction End information

QUICKDSP Tuning

- Command: SET QUICKDSP *userid* ON
- Directory: OPTION QUICKDSP
- Userid becomes transaction class 0 and never waits in the eligible list.
- Impacts decision to move from E-list to D-list on virtual machine basis.
- Does not change share normalization
- Does influence Elapsed Time Slice, making it similar to Class 2
- Recommended for:
  - Mission Critical Servers
  - Virtual Machines that are extensions of Operating System
    - RACF, TCP/IP, SFS, etc.
  - Key systems management virtual machines (which z/VM userid do you use to set tuning values?

Fitting in Storage

- Total Available Memory =
  - Total DPA page frames
  - Minus non-pageable frames
  - Minus system owned resident shared frames
  - Minus system owned locked frames
- SCLADL_SRMTOTST monitor field
- QUERY FRAMES “Pageable” is rough approximation
- Performance Toolkit FCX145 SCHEDLOG “Total Consid.”
  - Also shows current totals for each class
- Add in bonus if applicable for XSTORE

SRM STORBUF Tuning

- Protects system from thrashing on memory
- Command: SET SRM STORBUF *p1 p2 p3*
- Impacts decision to move from E-list to D-list on a class basis.
  - *P1:* percentage of memory available for classes 1,2,3
  - *P2:* percentage of memory available for class 2, 3
  - *P3:* percentage of memory available for class 3
- Defaults z/VM 5.4 & 6.1: 125 105 95
- Recommendation for Linux and other Guest environments: 300 250 200

SET SRM XSTORE Tuning

- Command: SET SRM XSTORE *percentage*
- Impacts decision to move from E-list to D-list on system basis.
- Determines how much expanded storage to be viewed as real storage for purpose of fitting user in STORBUF limitation.
- Percentage of existing expanded storage to add to available storage.

STORBUF Tuning Example

- Guests showing up in E-list, get estimated WSS size and class and increase STORBUF appropriately
- Example
  - 96GB of available memory
  - Two guests with WSS of 32GB each constantly appear as E3 users.
  - Currently: STORBUF 100 85 75
  - 64 / 96 = 67%
  - Increase STORBUF by 70 (round up the 67)
    - SET SRM STORBUF 170 155 145

SRM DSPBUF Tuning

- Controls the absolute number of virtual CPUs allowed into the D-list for each class.
- Command: SET SRM DSPBUF *n1 n2 n3*
  - N1: number of class 1 vCPUs permitted in D-list
  - N2: number of class 2 vCPUs permitted in D-list
  - N3: number of class 3 vCPUs permitted in D-list
- Defaults: 32676 32767 32767 = basically off
- Leave this command alone unless instructed by a z/VM performance expert.

Share Tuning

- Two flavors
  - Absolute
  - Relative
- Impacts calculation of D-list priority on userid basis (directly) and system basis (indirectly).
- Shares normalized before being used with other users in D-list and E-list per processor type:
  - If sum of absolutes is > 99%, then normalized to 99%
  - Relatives normalized to absolute leftovers.
- There is a minimum (regular) share and a limit (maximum) share
- Performance Toolkit FCX145 SCHEDLOG
  - “Sum of Abs. Shares”
  - “Sum of Rel. Shares”

Absolute Share Tuning

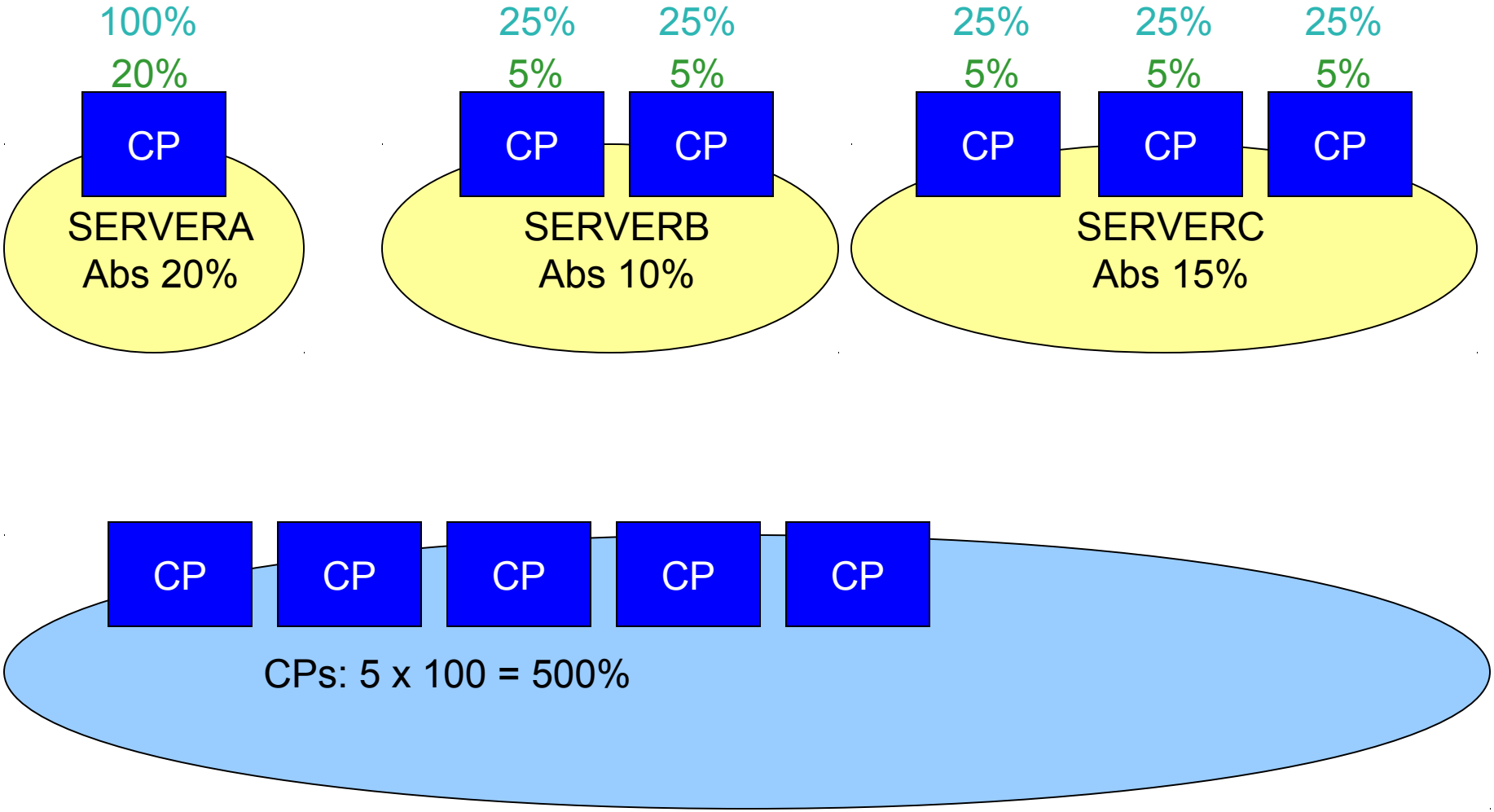
- Command: SET SHARE *userid* ABSOLUTE *ppp*%
- Directory: SHARE ABSOLUTE *ppp%*
- Percentage of system resources for user, in range 0.1 to 100%
- Value stays constant as long as sum < 99%

Relative Share Tuning

- Command: SET SHARE *userid* RELATIVE *nnnnnn*
- Directory: SHARE RELATIVE *nnnnnn*
- Value from 1 to 10000 (bigger being more important)
- As system becomes busier (more vCPUs in D-list), a relative share is normalized to a smaller value.

Share Tuning Example

NN% = (IPW) In Perfect World percentage of real processor
NN% = split of share per virtual processor



Maximum or Limit Share Settings

- Command and Directory have additional settings:
  - e.g. SET SHARE *userid* REL 2000 ABS 20% LIMITHARD
  - 2nd value is limit share
- Limit share can be relative or absolute
- Two types:
  - LIMITSOFT: means only allow virtual machine to use more than this amount of resources if no other virtual machine needs the resources.
  - LIMITHARD: means do not give the virtual machine more than this amount of resources even if they are available.
- ABSOLUTE LIMITHARD managed differently based on SET SRM LIMITHARD setting, added by VM64721.
  - SET SRM LIMITHARD DEADLINE current default (will change in future)
  - SET SRM LIMITHARD CONSUMPTION allows better accuracy

Virtual MP Virtual Machines & Specialty Engines

- Share value is distributed across the virtual processors
  - eg. Relative 100 on virtual 4 way looks like four Relative 25 virtual machines
- The share from a virtual CPUs put in an architecture stop state is redistributed to the non-stopped virtual CPUs
- The base VMDBK, vCPU, owns the memory of the virtual machine along with other key aspects. It stays in the D-list if any of the other vCPUs of the virtual machine.
- The share construct and ATOD is duplicated for each processor type available to the z/VM system.
- A virtual machine can have different Share settings for each processor type.

Other Considerations or Topics

- Priority (Offset calculations) based on minimal or normal share.
- Side effects of tuning – gated by I/O could limit how much CPU is used.
- Absolute does not mean “exact” or “precise”
- Dedicating processors requires changing how you look at things.
- TOD Tied concept
- Maxfall concept
- Growth Limit

Undesirable Features

- **Stuck in E-list:** An E-list deadline gets set much too far in the future (hours) in severe scenarios, even when no one else wants to run.
- **Non-dormant Dormant:** In highly constrained systems, users waiting on what should be short wait process (such as a page read), appear idle since task takes over 300 milliseconds. Virtual machine ends upin dormant list making analysis more difficult or misleading.
- **No Control on C1ETS:** There could be times when being able to bound class 1 ETS would be helpful.
- **Surplus or Excess Share Distribution:** If the entitled share is not used by virtual machines, the excess or surplus is distributed to other virtual machines that can use it. There are scenarios where this excess is not distributed proportionally to the normalized share, but that the virtual machine with the highest normalized share value gets it.

Summary

- Virtual Machines travel through various lists in VM scheduling, making various stops.
- You can see where the virtual machines are and have been through various monitor tools.
- A set of commands and tools exist to influence the system and individual virtual machines.