Processor Configuration and Analysis

- Barton@VelocitySoftware.com
- HTTP://VelocitySoftware.com

"If you can't Measure it,
I am Just Not Interested ™"

VELOCITY SOFTWARE

PROVEN PERFORMANCE

Topics

Performance Problem Overview

CPU Performance Analysis

- Basic concepts
- LPAR
- z/VM
- Linux

- Where to start

z/VM CPU Analysis and Configuration

- Master Processor
- PLDV, Dispatching
- Linux Guidelines

Common Reported CPU Performance Problems

Problems from a “Linux perspective”:

- Workload is timing out
- Applications are running slowly
- Workload/Server is in “CPU wait” (steal time is high)

Analysis must be from the top down

- LPAR Weights vs IFL utilization (entitlement)
- LPAR vCPU vs Share (entitlement spread over more vCPUs)
- z/VM Share settings poor (share spread over more vCPUs)
- Operation on GP, not on IFL engines (it happens)
- Processor utilization is high

Miscellaneous causes – Workload related:

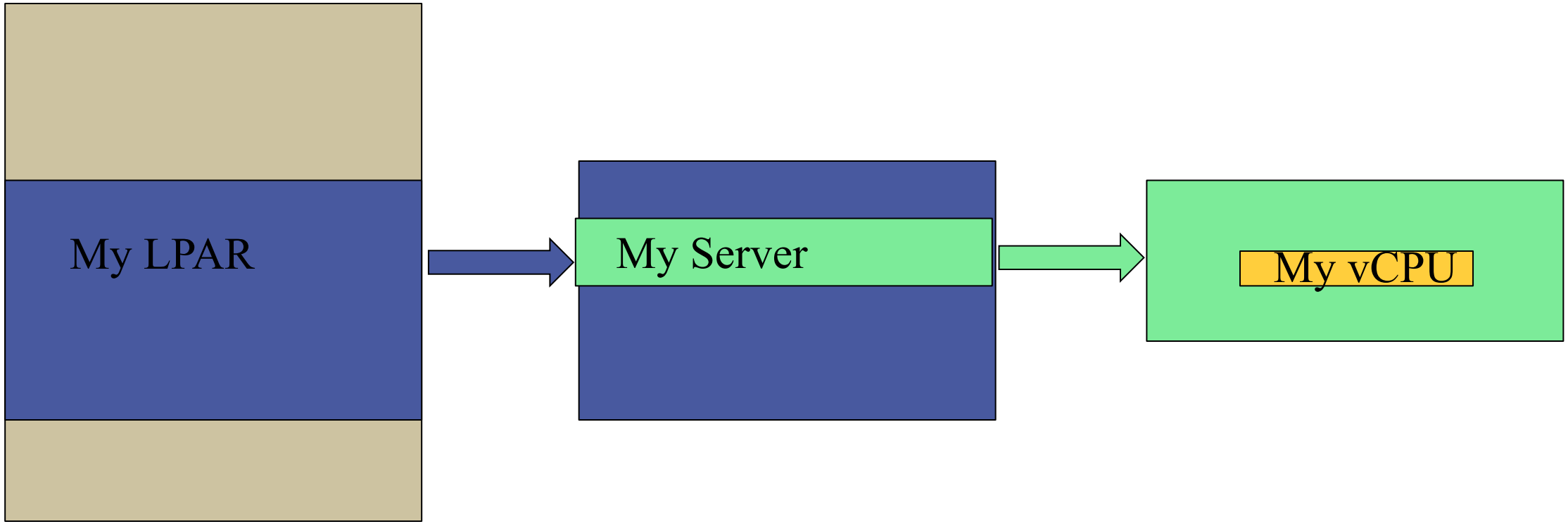
- z/VM Master processor is over utilized
- Cron jobs synchronized (100 processes across 100 servers)
- Spin locks – DIAG 44/9C (too many virtual machine vCPUs

Analysis starts at the top

Processor Utilization Hierarchy:

- CEC: TOTAL IFL (GP) Utilization (What is paid for)
  - Is there extra capacity?
- LPAR Utilization (What is allocated/used by the LPAR)
  - Is what is allocated actually being used?
  - "steal time" Virtual machine is dispatched but vCPU is used by another LPAR
- Virtual Machine/Linux server (What is allocated to the server)
  - "steal time" Linux is ready to run, but not being dispatched
- "My" Share (is there enough provided for the workload?)

Process Share – CPU Hierarchy



CEC, n engine
Weight -> share of CEC

Virtual machine
Share of LPAR

VCPU
Share of Virtual Machine

Process Share – CPU Hierarchy

Linux only measures Linux and “steal time”

- Bottom up analysis vs top down
- “top” gives a limited view
- Linux admins complain about “steal time”

Linux process top down analysis

- Are the engines on the CEC highly utilized?
- Is the LPAR sufficiently entitled?
- Is the Virtual Machine share sufficient?
- Is the process niced?

Processor Distribution

At the CEC Level:

- IFLs shared or dedicated at LPAR level

Shared Processor distribution “managed”:

1. The LPAR is assigned a “weight”
   - An “entitlement” of the IFLs (ESALPAR)
2. The Virtual Machine is assigned a “share”
   - A “share” of the LPAR (ESAUSRC/ESAUSP2)
3. The Linux Processes have a “priority”
   - Processes are “prioritized” by “nice” settings (ESALNXC/P)

Processor Performance Concepts - Utilization

What is “Percent” CPU Utilization?

- Percent of the box? (Capacity planning question)
- Percent of assigned?

Measured Utilization vs Reported Utilization

- Virtual Linux measures what?
  - Percent of wall clock time originally, now is “steal timer”
- z/VM measures what? CPU seconds (hardware timer)
- SMT: Core vs thread – this is confusing
- Hardware measurement is the only valid method of measuring CPU

Percent of Percent is misleading

- Can not be used directly for capacity planning
- Can not be used directly for accounting/chargeback
- Is often misleading for performance analysis

Processor Performance Concepts - Utilization

Utilization/Capacity is important

- IFLs are what you pay for
- Higher utilization requires less hardware and software
- Higher utilization requires correct configuration

CPU Utilization is used for:

- Performance Analysis
- Capacity Planning
- Accounting/Chargeback
- Operational Alerts

Processor Performance Reporting - Utilization

All zVPS numbers are measured in CPU seconds

- Percent is always based on CPU seconds divided by wall clock time
- 200% means using 2 engines worth of CPU seconds
- Measured by the hardware in microseconds

This impacts the measurements of:

- Total IFLs/GPs
- LPARs
- z/VM Virtual Machines
- Linux processes
- zVSE Jobs/Partitions
- z/OS Jobs/Partitions

Processor Utilization Components

LPAR Level:

- LPAR Physical Overhead
- LPAR Assigned time – Overhead
- LPAR Assigned time – Virtual

z/VM Level: (LPAR Assigned time – Virtual)

- System Time (z/VM Control Program)
- User Overhead (allocated system time)
- Emulation (z/VM guest time)

Linux Level: (Emulation – z/VM guest time)

- System time (kernel time)
- IRQ time
- User time (“real application work”)

IDLE

LPAR Options

Weights: Sets entitlement between Logical Partitions

- Set weights based on business requirements

Capping

- Limits Assigned Time to LPAR – use it carefully
- Useful for outsourcing or fixed contracts

Wait Completion

- “No” – gives up the processor if idle (default)
- “Yes” – the Partition keeps the processor, even if idle (rarely/never used)

Start with a z16

- Some number of engines (GP/CP, IFL, zIIP, ICF)

"Total" Utilization is the used capacity of the z16 (ESALPARS)

- LPAR Physical Overhead (Mgmt)
- LPAR Assigned time – Overhead (Ovhd)
- LPAR Assigned time – Virtual (Logical)

```
Totals by Processor type:
<---------CPU-------> <-Shared Processor bus>
Type Count Ded shared  Total  Logical Ovhd Mgmt
---- ----- --- ------ ------ -------- ---- ----
CP       7   0      7  511.9    501.5  4.5  5.9
IFL     10   0     10  915.6    894.5  8.6 12.5
ZIIP     3   0      3   23.9     22.3  0.4  1.2
```

Evaluate CEC perspective first:

- There are 10 IFLs that are “shared”
- IFLS are 92% assigned (915.6/1000)
- Note the system overhead – “Ovhd” and “Mgmt” (8.6 and 12.5)
- CPU delays can be expected! Shares/Weights need to be managed

```
Report: ESALPARS      Logical Partition Summary
-----------------------------------------------------------------

Totals by Processor type:
<----------CPU--------> <-Shared Processor busy->
Type Count Ded shared  Total  Logical Ovhd Mgmt
---- ----- --- ------ ------ -------- ---- ----
CP       7   0      7  511.9    501.5  4.5  5.9
IFL     10   0     10  915.6    894.5  8.6 12.5
ZIIP     3   0      3   23.9     22.3  0.4  1.2
```

LPAR Architecture

Z16 divided into “LPARs”

- Each LPAR configured with virtual CPUs
- Each LPAR gets a weight/entitlement

LPAR Entitlement:

- (LPAR Weight) / SUM(LPAR Weights)
- z/VM entitlement of IFLs (ZVMQA – 15% of 10 IFLs)

Report: ESALPARS Logical Partition Summary

Time	<--Complex--> Phys CPUs	Dispatch Slice	<--------Logical Partition-------> Name	Nbr	Virt CPUs	CPU Type	<%Assigned> Total	Ovhd	<-Assigned <---LPAR--> Weight	Pct	
00:15:00	23	Dynamic	Totals:	0	22	CP	506.0	4.5	999	100	
			Totals:	0	23	IFL	903.1	8.6	1000	100	
			ZVMQA	11	6	IFL	374.8	0.9	150	15.0	<---
			MVSPRD	7	10	CP	320.1	3.2	860	86.1	
			MVSQA	1	6	CP	181.8	1.1	71	7.1	
			ZVMDEQ	9	4	IFL	131.6	2.0	100	10.0	
			ZVMPRD	8	10	IFL	333.7	4.9	650	65.0	
			ZVMSHR	12	3	IFL	63.0	0.8	80	8.0	
			MVSTST	17	3	CP	5.1	0.1	8	0.8	

z/VM entitlement of IFLs (ZVMQA – 15% of 10 IFLs)

Report: ESALPARS Logical Partition Summary

Time	<--Complex--> Phys CPUs	Dispatch Slice	<-------Logical Partition-------> Name	Nbr	Virt CPUs	CPU Type	<%Assigned> Total	Ovhd	<Assigned <---LPAR--> Weight	Pct
00:15:00	23	Dynamic	Totals:	0	22	CP	506.0	4.5	999	100
			Totals:	0	23	IFL	903.1	8.6	1000	100
			ZVMQA	11	6	IFL	374.8	0.9	150	15.0
			MVSPRD	7	10	CP	320.1	3.2	860	86.1
			MVSQA	1	6	CP	181.8	1.1	71	7.1
			ZVMDEQ	9	4	IFL	131.6	2.0	100	10.0
			ZVMPRD	8	10	IFL	333.7	4.9	650	65.0
			ZVMSHR	12	3	IFL	63.0	0.8	80	8.0
			MVSTST	17	3	CP	5.1	0.1	8	0.8

Totals by Processor type:

Type	Count	Ded	shared	<---------CPU-------> Total	Logical	<-Shared Processor busy-> Ovhd	Mgmt
CP	7	0	7	511.9	501.5	4.5	5.9
IFL	10	0	10	915.6	894.5	8.6	12.5
ZIIP	3	0	3	23.9	22.3	0.4	1.2

ZVMQA LPAR entitlement:

- 15% of 10 IFLs
- 1.5 IFLs
- (Using 3.75)

ZVMPRD has priority

Virtual CPU Entitlement non-hiperdispatch (horizontal):

- Each vCPU in the LPAR gets equal part of the weight
- LPAR entitlement divided by the enabled vCPUs
  - (LPAR entitlement) / (Number of CPUs in the LPAR)
- The more vCPUs, the smaller the vCPU entitlement
- The more vCPUs the slower the work will go

- The same concept applies to Linux servers!

LPAR Configuration Summary

z/VM share of IFLs (always start here):

```
Report: ESALPARS     Logical Partition Summary
------------------------------------------------------------------------
          <--Complex--> <--------Logical Partition-------->  <-Assigned
          Phys Dispatch                Virt CPU  <%Assigned>  <---LPAR-->
Time      CPUs    Slice Name      Nbr CPUs Type Total   Ovhd  Weight  Pct
--------  ---- -------- -------- --- ---- ---- -----  ----  ------ ----
00:15:00    23  Dynamic Totals:    0   22  CP  506.0   4.5     999  100
                        Totals:    0   23 IFL  903.1   8.6    1000  100
                        ZVMQA     11    6 IFL  374.8   0.9     150 15.0
                        ZVMDEQ     9    4 IFL  131.6   2.0     100 10.0
                        ZVMPRD     8   10 IFL  333.7   4.9     650 65.0
                        ZVMSHR    12    3 IFL   63.0   0.8      80  8.0
Totals by Processor type:
<---------CPU-------> <-Shared Processor busy->
Type Count Ded shared  Total  Logical Ovhd Mgmt
---- ----- --- ------ ------ -------- ---- ----
IFL     10   0     10 915.6    894.5  8.6 12.5
```

- ZVMQA entitlement: 150/1000 (15%) of 10 shared IFLs
- So ZVMQA entitlement is 1.5 IFLs
- Virtual CPU entitlement: 1.5 IFLs / 6 vCPUs
- (.25 IFL core per vCPU)
- ZVMQA is using 375% shared IFLs (more than entitlement)
- IFLs running 915/1000% (92%) busy
- ZVMPRD entitlement: 6.5 IFLs but using 3.3

Entitlement field added! Validate assigned vs entitled

ESALPARS Logical Partition Summary

<-------Logical Partition------->				<Assigned Shares---->				Entitled
		Virt CPU		<%Assigned>		<---LPAR-->	<VCPU Pct>		CPU Cnt
Name	Nbr	CPUs	Type	Total	Ovhd	Weight	Pct	/SYS	/CPU	
Totals:	00	387	IFL	4451	156	3860	100			
L1A1	21	4	IFL	2.6	0.4	50	1.3	0.32	44.3	1.77
L1D1	01	50	IFL	167.0	10.1	500	13.0	0.26	35.5	17.75
L1D2	02	40	IFL	490.8	38.7	900	23.3	0.58	79.9	31.94
L1D3	03	30	IFL	1.2	0.4	50	1.3	0.04	5.91	1.77
L1D4	04	14	IFL	1.3	0.5	10	0.3	0.02	2.52	0.35
L1E1	05	20	IFL	64.8	3.6	500	13.0	0.65	88.7	17.75
L1C1	11	40	IFL	3228	80.5	200	5.2	0.13	17.7	7.10
L1C2	12	31	IFL	11.1	0.7	10	0.3	0.01	1.14	0.35
L1C3	13	14	IFL	1.4	0.5	10	0.3	0.02	2.52	0.35
L1C4	14	14	IFL	1.0	0.4	10	0.3	0.02	2.52	0.35
L1A2	22	2	IFL	1.0	0.4	10	0.3	0.13	17.7	0.35
L1B1	25	20	IFL	310.7	7.1	300	7.8	0.39	53.2	10.65
L1B2	26	31	IFL	99.0	5.5	700	18.1	0.58	80.1	24.84
L1B3	27	30	IFL	1.2	0.4	50	1.3	0.04	5.91	1.77
L1B4	28	14	IFL	1.0	0.4	10	0.3	0.02	2.52	0.35
LN12	31	4	IFL	0	0	Ded	2.8	0	0	0
LOI3	32	14	IFL	64.0	4.7	500	13.0	0.93	127	17.75

LPAR vCPU Case Study

```
Report: ESALPARS     Logical Partition Summary                          TEST MAP
Monitor initialized: 08/04/03 at 18:52:10 on 2084 serial 4B54A          First recor
-------------------------------------------------------------------------------
          <--Complex--> <-------Logical Partition--->  <-Assigned Shares---->
          Phys Dispatch                 Virt <%Assigned>  <---LPAR--> <VCPU Pct>
Time      CPUs     Slice Name      Nbr CPUs Total  Ovhd  Weight  Pct /SYS /CPU
--------  ---- -------- -------- --- ---- -----  ----  ------ ---- ---- ----
Average:     8 Dynamic  Totals:    0   22 188.7   2.1      60  100
                        ZVM        6    8  82.8   1.4      10 16.0 2.00 16.0
                        CF01       1    1  99.9   0.0      10 16.0 16.0  128
                        LINUXSW    2    2     0     0      10 16.0 8.00 64.0
                        S01        3    4   4.6   0.4      10 16.0 4.00 32.0
                        S02        4    0
                        VMTPC      5    5   1.2   0.2      10 16.0 3.00 24.0
                        ZVMCSS1   16    2   0.2   0.0      10 16.0 8.00 64.0
```

- ZVM allocated (10/60) or 16% of 8 CPUs (~1.2 Entitlement)
  - Each virtual CPU allocated is 2% of the system (8 CPUs)
  - Each processor is entitled to 16% of real processor

(HiperDispatch modifies vCPU entitlement dynamically)

LPAR Weights Example (Horizontal)

ESALPAR (Partial report with horizontal scheduling)

Note that each vCPU is running at 10%

z/VM can dispatch 8 concurrent virtual machines

- Less queueing and slower service
- Each single vCPU runs “VERY slow”

This is why we now have HiperDispatch/Vertical scheduling

<--Logical->		<--------Logical Processor---------							
	Phys	Dispatch	<-Partition>		VCPU	<%Assigned>			Cap-	Wait
Time	CPUs	Slice	Name	No.	Addr	Total	Ovhd	Weight	ped	Comp
Average:	8	Dynamic	ZVM	6	0	8.3	0.2	10	No	No
					1	10.2	0.2	10	No	No
					2	11.0	0.2	10	No	No
					3	11.1	0.2	10	No	No
					4	10.5	0.2	10	No	No
					5	10.5	0.2	10	No	No
					6	10.5	0.2	10	No	No
					7	10.6	0.2	10	No	No
						82.8	1.4			
					LPAR					

LPAR Share Example

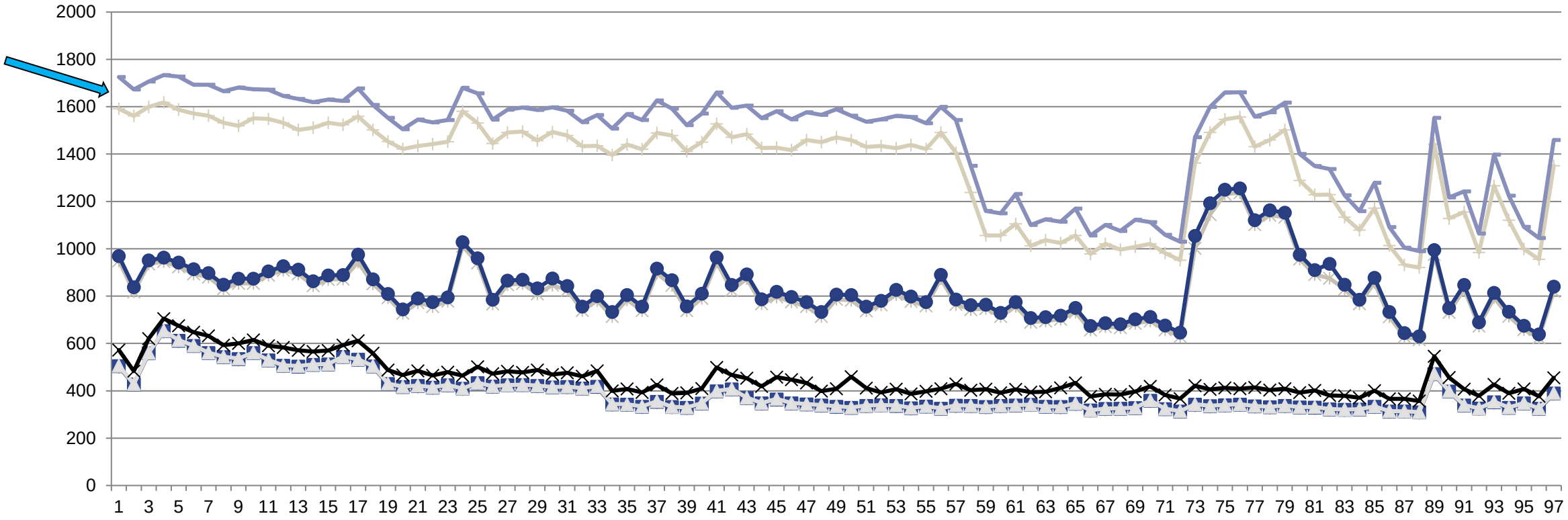
Guaranteed processor share (speed) by dropping vcpu from 8 to 4

- ((10 / 60) / 4) * 8 = .32 (up from .16)
- This is a real problem in many installations!
- This is why HiperDispatch is required – vertical scheduling

Too many logical processors will slow you down!

- Specifically the master processor…
- The same concept applies to Linux virtual processors
- This is why hiperdispatch is good!

LPAR Configuration Overhead



17 IFLs, 7 LPARs, 17 vCPUs each – 7:1 overcommit

Physical Overhead (6-7%) significant from real processor overcommit

The Problem:

- If too many LPAR vCPUs are defined, performance declines
- Cache competition and overhead
- Errors on the weight settings by installations

The Solution: HiperDispatch – implemented in both z/OS and z/VM

- Parking of low entitlement vCPUs and weight redistribution
- Parking level is determined every 2 seconds… TOO MUCH PARKING?
- Recommendation – if using 4 engines or less, horizontal is better (no SMT)
- SET SRM POLARIZATION HORIZontal | VERTical

(ESAOPER)

```
00:00:03 CPU Park from 20 to  18 CPUUtil= "8.75", Projected= "9.26"
00:00:05 CPU Unpark from  18 to  22 CPUUtil= "8.09", Projected= "8.97"
00:00:09 CPU Park from 22 to  18 CPUUtil= "7.39", Projected= "8.98"
00:00:11 CPU Unpark from  18 to  20 CPUUtil= "7.32", Projected= "8.80"
00:00:13 CPU Park from 20 to  18 CPUUtil= "8.15", Projected= "8.98"
00:00:17 CPU Unpark from  18 to  20 CPUUtil= "8.40", Projected= "8.97"
00:00:29 CPU Park from 20 to  18 CPUUtil= "8.62", Projected= "10.2"
00:00:37 CPU Unpark from  18 to  20 CPUUtil= "8.40", Projected= "8.96"
00:00:39 CPU Park from 20 to  18 CPUUtil= "8.48", Projected= "8.96"
00:00:41 CPU Unpark from  18 to  20 CPUUtil= "8.31", Projected= "8.93"
00:00:43 CPU Park from 20 to  18 CPUUtil= "8.27", Projected= "8.93"
00:00:53 CPU Unpark from  18 to  20 CPUUtil= "8.57", Projected= "8.76"
00:00:57 CPU Park from 20 to  18 CPUUtil= "7.82", Projected= "8.91"
```

LPAR with HiperDispatch

```
Report: ESALPAR      Logical Partition Analysis
Monitor initialized: 05/11/21 at 03:36:13 on 8561 serial XXXXXX
-------------------------------------------------------------------
         CEC  <-Logical Partition-> <----------Logical Processor--
         Phys                Pool     VCPU <%Assigned> VCPU Weight/
Time     CPUs Name      No   Name     Addr Total  Ovhd TYPE Polar
-------- ---- -------- --- -------- ---- -----  ---- ---- --- ---
03:38:00   79 VSILNX1   31         .    0   6.7   0.3 IFL  300 VHi
                                         1   5.1   0.2 IFL  300 VMe
                                         2   7.4   0.2 IFL  300 VMe
                                         3   0.0   0.0 IFL  300 VLo
                                             -----  ----
                                    LPAR  19.1   0.7

              VSIMGTA   01         .    0   3.3   0.1 CP    38 VMe
                                         1   2.8   0.1 CP    38 VMe
                                         2   0.0   0.0 CP    38 VLo
                                         3   0.0   0.0 CP    38 VLo
                                         4   0.0   0.0 CP    38 VLo
                                         5   0.0   0.0 CP    38 VLo
                                         6   0.0   0.0 CP    38 VLo
                                         7   0.0   0.0 CP    38 VLo
                                         8   0.0   0.0 CP    38 VLo
                                         9   0.0   0.0 CP    38 VLo
                                             -----  ----
                                    LPAR   6.1   0.2
```

HiperDispatch requires Vertical scheduling

- Exposed on ESALPAR
- To get more “Vertical Highs” requires higher weights

See: CP SET SRM UNPARKING LARGE | MEDIUM | LOW

HiperDispatch Questions

What happens to work on a CPU when it gets parked?

- z/OS gets a 50ms warning
- z/VM? Work gets “stolen” at some point
- New Linux support for hiperdispatch – process might get parked
- Recent announcement suggests z/VM will take advantage of the 50ms warning

Does HiperDispatch improve performance?

- Yes – there are fewer vCPUs with higher dispatching weights – which is better

LPAR Parking Options

CP SET SRM UNPARKING LARGE | MEDIUM | SMALL

- Large was the default until z/VM 7.2, then it became medium
- Large unparks almost all vCPUs, even vertical-low
- Use “LARGE” on small LPARs, “MEDIUM” on larger LPARs

CP SET SRM EXCESSUSE TYPE IFL HIGH | MEDIUM | LOW

- Medium is the default
- HIGH aggressively uses vertical-low even though not entitled

CP SET SRM CPUPAD TYPE IFL 200%

- Pads the SRM CPU estimates of how much excess capacity to keep online
- Only valid when GPD is not available (other LPAR utilization data)

The z/VM Master Processor

Master Processor / Locking Overview

Every operating system has multiple “locking” methods

Much system code is NOT re-entrant

- Must be single threaded
- Can not update one control block by multiple processors simultaneously

Implementation

- Hardware locks: TS, CS, CDS instructions
- Software locks: “Ownership” of resources (such as in a database)
- Running on the Master Processor

SPIN Locks

- Test for lock, if it fails, test for lock
- Linux uses “spin lock”, replaced with DIAG44, then DIAG9C
- Linux spin locks are an issue, costs CPU

Resource Serialization – Master Processor

Many CP processes run “master only” to ensure integrity of the system

- Spooling
- Some IUCV services (*MSG, *RPI, *ACCOUNT from CP)
- Page migration
- Execution of ALL CP commands
- Line mode console I/O

Master processor utilization shows up as:

- Higher system overhead
- Higher user overhead

Higher Master CPU busy on a system with more processors

- Master calls are measured
- Simulation wait is measured
- Processor imbalance can be a problem

CPU Example

- User overhead high on master
- System overhead high on master
- Master processor can be a limiter

```
Report: ESACPUU      CPU Utilization
-------------------------------------------------------------------------------------------
         <----Load---->     <-----CPU (percentages)-----> <------External (per second)------->
         <-Users-> Tran     Total  Emul  User   Sys  Idle <--Page---> <--Spool--> RSCH+
Time     Actv In Q /sec CPU  util  time ovrhd ovrhd  time  Read Write  Read Write  SSCH  ExInt
-------- ---- ---- ---- -  ----- ----- ----- ----- ----- ----- ----- ----- ----- -----  -----
09:19:12    7  5.0  0.1  1   99.4  20.9  58.8  19.8     0     0     0     0     0     3    140
                          2   84.7  43.6  30.7  10.3  15.0     0     0     0     0     0    154
                          3   84.2  43.2  30.9  10.1  15.5     0     0     0     0     0    153
                          4   84.5  43.6  31.1   9.7  15.2     0     0     0     0     0    155
                             ----- ----- ----- ----- ----- ----- ----- ----- ----- -----  -----
System:                      352.7 151.3 151.6  49.9  45.7     0     0     0     0     3    602
```

Would adding another processor help this system?

Customer reports very bad performance

- Look at wait states first

Master Processor Case Study

Report: ESAXACT Transaction Analysis Velocity Software, Inc.

			<-Percent non-dormant->														
UserID	<-Samples->							E-	D-	T-		Tst				Lim	Pct
/Class	Total	In Q	Run	Sim	CPU	SIO	Pg	SVM	SVM	SVM	CF	Idl	I/O	Ldg	Oth	Lst	Elig
System:	5936	149	5.4	34	8.7	0	3	0	0	6.0	2	36	4.7	.	0	.	0
Hi-Freq:	176K	7057	2.0	17	2.8	0	1	0	3.8	4.2	49	17	3.1	0	0	.	0
***Resource use by User Class																	
*Servers	3720	568	3.0	29	4.2	0	0	0	21	6.9	1	28	7.6	0	0	.	0
*Keys	1080	490	1.6	0.6	6.7	0	0	0	16	19	1	43	13	0	0	.	0
*TheUsrs	172K	6108	1.9	16	2.6	0	1	0	1.2	3.0	57	14	2.5	0	0	.	0

User state sampling shows wait compared to “running”

- Significant amount of CPU wait
- Simulation wait is even greater

Master Processor Case Study

```
Report: ESASSUM Subsystem Activity Velocity Software, Inc.
-------------------------------------------------------------------------------------------------------
         <---Users----> Transactions <Processor> Storage (MB) <-Paging--> <-----I/O-----> <MiniDisk> Spool
         <-avg number->    Per  Avg. Utilization Fixed Active <pages/sec> <-DASD--> Other <-Cache--> Page
Time       On Actv In Q Minute  Resp Total Virt.  User Resid. XStore DASD Rate Resp  Rate  Rate %Hit Rate
-------- ---- ---- ---- ------ ----- ----- ----- ----- ------ ------ ---- ---- ---- ----- ----- ---- -----
08:00:08 1479  244 34.3 1310.1 0.603   124    87  36.9  192.0    888  451  641 15.4    40 687.9 49.3    36
08:01:08 1500  248 46.0 1260.9 0.543   147   110  37.3  192.7    904  494  732 20.1    37 881.6 53.9    32
*********************************************Summary*********************************************
Average: 1483  245 37.3 1297.8 0.589   130    93  37.0  192.1    892  461  664 16.7    39 736.4 50.7    35
```

A high-level view of processor utilization shows system with capacity to spare

- Using 147% out of 300%

Next step – “zoom” to processor configuration

Master Processor Case Study - Effects of Logical Partitioning

Report: ESACPUU CPU Interval Analysis Velocity Software, Inc.

	<----Load---->				<-----CPU (percentages)----->					<---Internal (per second)---->				
	<-Users->	Tran			Total	Emul	User	Sys	Idle	Diag-	Inst.	SIE	Fast	Page
Time	Actv	In Q	/sec	CPU	util	time	ovrhd	ovrhd	time	nose	sim.	intrcp	path	fault
08:00:08	244	34.3	24.6	0	48.5	27.3	16.7	4.5	9.9	1449	1478	1753	0	18
				1	35.9	28.8	5.2	1.9	11.8	818	599	716	0	9
				2	39.5	31.4	5.9	2.2	13.5	902	682	815	0	11
System:					124.0	87.4	27.9	8.7	35.3	3170	2758	3284	0	37
08:01:08	248	46.0	24.0	0	53.6	32.5	16.7	4.4	7.1	1557	1588	1806	0	24
				1	44.6	37.2	5.4	1.9	6.5	843	594	685	0	11
				2	48.8	40.2	6.4	2.2	7.4	903	704	817	0	12
System:					147.0	109.9	28.5	8.6	21.0	3303	2886	3308	0	48

A more detailed view of processor utilization seems to confirm this hypothesis

- CPU to spare??

Master Processor – Case Study

```
Report: ESALPAR Logical Partition Analysis Velocity Software, Inc.
-------------------------------------------------------------------------------
         <----Load----> <--Complex--> <--Logical-> <--------Logical Processor----->
         <-Users-> Tran Phys Dispatch <-Partition> VCPU <%Assigned>       Cap- Wait
                        Slice    Name      No. Addr Total Ovhd Weight  ped Comp
-------- ---- ---- ---- ---- -------- -------- --- ---- ----- ---- ------ ---- ----
08:02:08  244 34.3 24.6    3 Dynamic CMS2        1    0  58.7  0.2    155   No  Yes
                                                    1  47.8  0.1    155   No  Yes
                                                    2  53.2  0.1    155   No  Yes
                                                       ----- ----
                                                 LPAR 159.7  0.4

                                SWCF        2    0  36.6  0.1    130   No  Yes
                                                    1  43.0  0.1    130   No  Yes
                                                    2  46.7  0.1    130   No  Yes
                                                       ----- ----
                                                 LPAR 126.3  0.3

                                CMS8        3    0   9.1  0.1     15   No  Yes
                                                    1   4.6  0.2     15   No  Yes
                                                       ----- ----
                                                 LPAR  13.7  0.3

Total Logical Partition busy:                             299.6
Total Physical Management time: 0.366
```

z/VM system does not have access to 100% of each processor

- 51% entitlement, 1.5 processors (155/300)
- Each vCPU is entitled to 50% of one real CPU, the master processor is constrained
- Reducing CMS2 LPAR to 2 processors will perform better

Master Processor Activity

```
Report: ESAPLDV      Processor Local Dispatch Vector Activity  Linux Test  ESAMAP 3.7.4
-------------------------------------------------------------------------------------
         <----Users----->   Tran       <VMDBK Moves/sec>  <--------PLDV Lengths------> Dispatcher
Time     Logged Actv In Q   /sec  CPU  Steals  To Master   Avg  Max Mstr MstrMa %Empty Long Paths
-------- ------ ---- ----  -----   -   ------  ---------  ----  --- ---- ------ ------ ----------
12:01:00    129  103  118    9.1   0       0        2.5   3.2  4.0  0.0      1.    8.3     4497.1
                                   1       0          0   2.1  4.0    .          38.3     3942.1
                                   2       0          0   2.0  4.0    .          41.7     3942.7
                                   3       0          0   1.8  3.0    .          38.3     3741.7
                                          ------  ---------  ----  --- ---- ------ ------ ----------
System:                                    0        2.5   9.2 15.0  0.0      1.  126.7    16123.5
```

Each processor has a PLDV - “Processor Local Dispatch Vector”

The Dispatcher selects users from the PLDV

The Master Processor has a special PLDV from which “master only” work for users is selected

Evaluate if High Simulation Wait

(Steals 0 – the system provides affinity to maintain cache)

Master Processor Activity

```
Report: ESACPUU      CPU Utilization Report                          Linux Test
Monitor initialized: 05/06/08 at 12:00:00 on 2094 serial AEA7D     First record analyzed:
-------------------------------------------------------------------------------------
         <----Load---->     <-----CPU (percentages)-----> <------External (per second)---
         <-Users-> Tran     Total  Emul  User    Sys  Idle <--Page---> <--Spool--->  RSCH+
Time     Actv In Q /sec CPU  util  time ovrhd ovrhd  time  Read Write  Read Write  SSCH
-------- ---- ---- ---- -  ----- ----- ----- ----- ----- ----- -----  ----- -----  -----
12:01:00  103  118  9.1  0   92.8  88.6   2.3   1.9   7.2    11    52      0     0   220
                          1   93.8  90.5   2.2   1.0   6.2    14     0      0     0   182
                          2   94.4  90.9   2.2   1.2   5.6    17     0      0     0   196
                          3   94.5  90.9   2.1   1.5   5.5    13     0      0     0   179
                             ----- ----- ----- ----- ----- ----- -----  ----- -----  -----
System:                     375.4 361.0   8.9   5.5  24.4    55    52      0     0   778
```

Processor utilization has three components:

- Emulation time – running users in Interpretive Execution
- User overhead – CP time performing services for a user
- System overhead – CP “housekeeping”

Note the master processor – Only a problem if architecturally constrained

Master Processor Linux Case Study

Report: ESAXACT Transaction Delay Analysis

UserID /Class	<-Samples-> Total	In Q	<-----Percent non-dormant (Wait Run	Sim	CPU	SIO	Pag	E- SVM	D- SVM	T- SVM
01/12/21										
14:01:00	140	126	0	56	0	0	0	0	0	0
Hi-Freq:	11400	7780	1.4	40	20	10	0.9	0	1.6	0.1
Hi-Freq:	12000	8038	0.7	18	1.1	0.0	0.0	0	1.4	27
Hi-Freq:	12000	7932	1.9	1.2	2.0	0.0	0.5	0	0.7	41
Hi-Freq:	12000	8044	1.0	1.0	1.3	0.0	0.0	0	1.1	42
Hi-Freq:	12000	8006	0.9	0.9	2.8	0.0	0	0	1.3	41
14:23:00	140	132	2.3	52	37	0	0	0	0	0
Hi-Freq:	11400	7601	1.7	19	31	3.9	0.3	0	1.4	17
Hi-Freq:	11400	7694	1.0	45	16	6.3	0.3	0	1.5	0.1
Hi-Freq:	12000	8068	0.8	18	0.5	0	0.0	0	1.5	27
Hi-Freq:	12000	8065	1.1	0.9	1.2	0.0	0.0	0	1.5	42
Hi-Freq:	12000	8028	1.2	0.9	1.6	0.0	0	0	1.6	42
Hi-Freq:	12000	8001	1.0	0.8	2.2	0.0	0.0	0	1.7	42
Hi-Freq:	12000	7975	1.1	0.9	1.3	0.0	0.0	0	2.0	42
Hi-Freq:	11600	7725	1.2	8.3	19	0.1	0.0	0	2.3	30
Hi-Freq:	11286	7689	1.7	48	27	3.7	0.4	0	2.0	0.7
Hi-Freq:	11880	7832	0.9	3.7	0.3	0.0	0.0	0	1.9	39
Hi-Freq:	11880	7809	0.9	0.8	0.5	0.0	0	0	1.9	42

Simulation wait is “sometimes very high”

CPU wait is “sometimes very high”

Master Processor Case Study - “z” Processor Overview (ESAHDR)

```
Machine Model/Type                    Z13:2964/725
Multithreading Status:Enabled
System Sequence Code                   00000000000B9177
Processor  0 model/serial            2964-725  /0D9177
Processor  1 model/serial            2964-725  /0D9177
Processor  2 model/serial            2964-725  /0D9177
Processor  3 model/serial            2964-725  /0D9177
Processor  4 model/serial            2964-725  /0D9177 Master
Processor  5 model/serial            2964-725  /0D9177
..............
Processor 18 model/serial            2964-725  /0D9177
Processor 19 model/serial            2964-725  /0D9177

 Power of processor in terms of service Units: 56939
 CPU Capability Factor:                 492
 CPU(GP) Capability Factor:             492
 CPU Cycles/ns:                        5000
 CPU Cycles/ns (GP):                   5000
 Operating on IFL Processor(s)
 Channel Path Measurement Facility(CPMF) Extended is installed
```

Service Units from table

Understand the CEC (two books)

- z/VM (20 threads)

Master Processor Case Study

Report: ESACPUU CPU Utilization Report

Time	Total util	Emul time	User ovrhd	Sys ovrhd	Idle time	Steal time
	<--------CPU (percentages)-------->					
01/12/21						
System:	361.9	6.2	322.9	32.8	137.1	1.0
System:	157.5	106.4	41.0	10.0	335.3	7.2
System:	311.9	212.3	79.8	19.8	174.0	14.1
System:	189.1	138.5	39.7	10.8	301.6	9.3
System:	172.4	125.2	36.7	10.4	318.8	8.8
System:	175.4	132.6	33.9	8.9	316.2	8.4
System:	168.9	127.1	33.2	8.6	322.8	8.3
System:	187.8	146.7	32.7	8.4	304.4	7.9
System:	168.0	125.8	33.4	8.8	323.6	8.4
System:	176.7	131.2	36.9	8.6	315.1	8.1
System:	379.6	68.2	299.4	12.1	111.4	8.9
System:	301.9	8.6	268.6	24.7	200.6	0
System:	143.5	98.4	36.3	8.9	349.5	7.0
System:	167.9	125.9	33.2	8.8	323.8	8.3
System:	176.5	131.2	36.6	8.7	315.4	8.1
System:	168.1	123.2	36.3	8.7	323.6	8.2
System:	168.0	126.0	33.7	8.4	323.9	8.1
System:	276.8	101.6	165.8	9.5	214.3	8.9
System:	476.7	16.7	411.8	48.2	21.9	1.5
System:	186.2	142.7	34.9	8.6	305.4	8.4
System:	165.8	124.3	32.9	8.6	325.9	8.3

ESACPUU:

- Is CPU at 100%?
- What is the user overhead?
- Which users?

Master Processor Case Study

```
Report: ESAUSP2       User
--------------------------
         <---CPU time--->
UserID   <(Percent)> T:V -
/Class   Total  Virt  Rat
-------- ----- ----- ----
01/12/21
14:01:00 321.7  6.18 52.1
14:02:00 147.5 106.4 1.39
14:03:00 292.1 212.3 1.38
14:04:00 178.2 138.5 1.29
14:05:00 162.0 125.3 1.29
14:06:00 234.2 169.0 1.39
14:07:00 186.7 138.7 1.35

14:15:00 154.3 122.5 1.26
14:16:00 165.7 128.0 1.29
14:17:00 161.1 126.7 1.27
14:18:00 166.5 132.6 1.26
14:19:00 160.2 127.1 1.26
14:20:00 179.4 146.7 1.22
14:21:00 159.1 125.8 1.27
14:22:00 168.1 131.2 1.28
14:23:00 378.5 68.23 5.55
14:24:00 265.5  8.49 31.3
14:25:00 134.7 98.45 1.37
```

ESAUSP2:

- Check user data
- Which users?
- Non-specific

Master Processor Case Study

```
Report: ESAMFC       MainFrame Cache Magnitudes
-----------------------------------------------
             <CPU Busy><-------Processor------>
             <percent> Speed/<-Rate/Sec->
Time      CPU Totl User Hertz Cycles Instr Ratio
-------- --- ---- ---- ----- ------ ----- -----
01/12/21
System:       362  6.2 5208M  18.8G 19.0G 0.990
System:       157  106 5208M  8222M 1856M 4.430
System:       312  212 5208M  16.3G 3083M 5.287
System:       189  139 5208M  9872M 1830M 5.394
System:       172  125 5208M  9001M 1692M 5.319
System:       248  169 5208M  12.9G 2341M 5.531

System:       188  147 5208M  9797M 2365M 4.143
System:       168  126 5208M  8769M 1691M 5.187
System:       177  131 5208M  9225M 2040M 4.521
System:       380 68.2 5208M  20.1G 17.0G 1.184
System:       302  8.6 5208M  15.3G 15.1G 1.015
System:       144 98.4 5208M  7495M 1623M 4.619
System:       168  126 5208M  8766M 1646M 5.326
System:       177  131 5208M  9212M 1959M 4.702
System:       168  123 5208M  8777M 1794M 4.891
System:       168  126 5208M  8769M 1748M 5.016
System:       277  102 5208M  14.5G 8991M 1.616
System:       477 16.7 5208M  24.8G 24.8G 1.000
System:       186  143 5208M  9716M 2370M 4.100
System:       166  124 5208M  8655M 1634M 5.297
```

ESAMFC:

- Workload changes
- Cycles / instruction
- VERY TIGHT LOOP?
- System function?

Master Processor Case Study

```
Report: ESADIAG      Diagnose Rate Repoate   ZMAP 5
-------------------------------------------------
Date     CPU <--Total-->   <----------------------
/Time        <Diags/Sec>   DIAG: Rate DIAG: Rate
             User  IBM
-------- --- ----- -----   ----- ----  ----- ----
01/12/21
System:          0 517.7   0000:  0.4  0024:  8.4
System:          0  1225   0000:  1.6  0044: 85.4
System:          0  6349   0000:  0.1  0044: 5952 ←

System:          0 326.0   0000:  0.2  0044: 17.4
System:          0 295.9   0000:  0.2  0044: 18.0
System:          0  1076   0000:  0.8  0040:  0.0
System:          0 907.8   0000:  0.3  0044:  3.9
System:          0 311.3   0000:  0.1  0044:  7.1
System:          0 267.7   0000:  0.1  0044:  6.9
System:          0 311.3   0000:  0.2  0044:  7.6
System:          0 426.4   0000:  0.7  0044:  6.4
System:          0 923.0   0000:  0.3  0044:  9.1
System:          0 196.7   0000:  0.1  0044:  3.0
System:          0  1838   0000:  0.1  0044: 1792
System:          0 366.2   0000:  0.1  0044:  139
System:          0 470.8   0000:  0.8  0044: 10.1
System:          0 923.0   0000:  0.3  0044:  9.5
System:          0 333.0   0000:  0.1  0044:  5.8
System:          0 356.4   0000:  0.2  0044:  4.7
System:          0 252.1   0000:  0.1  0044:  5.2
System:          0 227.1   0000:  0.7  0044: 12.0
System:          0  1029   0000:  0.7  0044: 43.3
System:          0 286.5   0000:  0.1  0044:  5.8
```

Processor Analysis/Simulation can be tedious

Look for patterns in CPU functions

- Diagnose?
- “Symptom”?

Look at ESAUSRD for specific user, if that is the issue

Master Processor Case Study

Report: ESADIA2		Dt			Vel
Date /Time	CPU	QUICKDSP Adds	<--DIAG9C counts---> HCPDSP	HCPSYN	HCPHVR
01/12/21					
System:		5.1	0	1136.5	0
System:		8.2	0	0.1	0
System:		7.8	0	0.0	0
System:		7.7	0	0.0	0
System:		8.7	0	0	0
System:		8.7	0	0.0	0
System:		8.0	0	0.1	0
System:		8.3	0	0.0	0
System:		8.1	0	0.1	0
System:		5.4	0	1586.5	0
System:		5.5	0	1392.3	0
System:		7.6	0	0.1	0
System:		8.0	0	0	0
System:		7.8	0	0.1	0
System:		8.2	0	0.2	0
System:		7.7	0	0.1	0
System:		7.5	0	203.3	0
System:		6.1	0	1278.7	0
System:		7.8	0	0	0
System:		7.6	0	0.0	0

Processor Analysis/Simulation can be tedious

Look for patterns in other CPU functions

- ESADIA2 (obscure)
- HCPSYN
- SIMWAIT direct correlation
- Master processor related

When IBM sees a problem, monitor fields are created

z/VM implemented Diagnose in the microcode for fast path function and communication for guests

Common diagnose instructions:

- Page release (214) – tells z/VM page no longer in use
- Spin lock (44, 9C) – tells z/VM not to dispatch, wait for lock

Diagnose Analysis

```
Report: ESAUSRD
Monitor initia6 seri: 06/
-------------------------
UserID  Total
/ClassID rate    044    09C
-------- ----  ----   ----
21:25:00  15K  9149   5392
 ***User Class
TheUsers  15K  9149   5392

LDBAMAP1  114   106    7.8
LDBPMPC1  169   127   42.6
LDMDMPC1  292   167    125
LEACMAP1  307   306    0.6
LEBAMAP1  264   255    9.0
LLBAMAP1  229   228    0.7
LPBAMAP1 53.4  42.5   10.9
LPBAMAP2  373   372    1.9
LQACMAP1  513   299    214
LQB1SDB4  126   122    4.1
LQCDEWN3  203   199    3.8
LQECOSM1 50.0  49.2    0.8
LQECOWN1  232   174   57.2
LQEPBDM1  186   184    1.3
LQEPBHT1  162   161    0.8
LQEPBWN1 34.8  27.6    7.2
LQFRXDB1  4.5   0.0    3.5
LQFXEDM1  155   154    1.5
LQFXEWN1  664  21.1    643
```

Linux diagnose for locking:

- DIAG44 is high overhead, DIAG9C is not

Two data sources:

- “System” (CPU by CPU)
- “User”

```
Report: ESADIAG
Date      CPU <--Total-->  ts per Second-------------------
/Time        <Diags/Sec>  DIAG: Rate DIAG: Rate DIAG: Rate
             User  IBM
-------- --- ----- -----  ----- ---- ----- ---- ----- ----
21:25:02   0     0 963.3  0024:    0 0044:  431 0058:  0.1
                          007C:  0.1 008C:    0 009C:  430
                          0270:  9.6 0288:  0.6 02FC:  0.1
              ----- -----  ---- ---- ----- ---- ----- ----

System:           0 14883  0024:  0.1 0044: 9149 0058:  0.1
                          007C:  0.2 008C:  0.0 009C: 5392
                          0270: 76.0 0288:  8.7 02FC:  0.5
```

z/VM Shares – Absolute and Relative

z/VM Shares – Absolute and Relative

- Absolute (ABS) Share is a percent of an LPAR
- Relative (REL) Share is comparable to LPAR “weight”

When to use Absolute vs Relative?

- If share should go up as workload increases (TCPIP, RACF), then use ABS
- If the resource requirement is to be guaranteed, also use ABS
- If share should go down as more users logon, then use REL

IBM Defaults are not optimum…

Processor Utilization Hierarchy

Given a weight and entitlement – now z/VM:

Processor Utilization Hierarchy:

- CEC: Total IFL Utilization
  - What is paid for
- LPAR Utilization
  - What is allocated / used by an LPAR
  - An “entitlement” of IFLs (ESALPARS)
- Virtual Machine/Linux server
  - What is allocated to a server / virtual machine
  - A “share” of the LPAR (ESAUSRC/ESAUSP2)
- “My” share
  - Is there enough provided for the workload?

Normalizing z/VM Shares

z/VM virtual machines have a SHARE of the LPAR

- Each Virtual Machine is assigned a **relative** or **absolute** share
- SRMRELDL is the value of “total relative”
- Share is then “normalized” to “normalized share”
  - Normalized = absolute
  - Normalized = (relative / (SRMRELDL)) * (100 – absolute)

“Normalized” share is a percent of the LPAR

- Normalized share is the “guarantee”

Each vCPU has an equal part of the VM normalized share

- Linux process running on a virtual machine vCPU
  - Gets virtual machine vCPU share

Scheduling/Dispatching vCPUs is based on the normalized share

Processor Measurements – User View

```
Report: ESAUSP2       User Resource Rate Report
Monitor initialized: 05/06/08 at 12:00:00 on 2094 serial
-------------------------------------------------------
         <---CPU time--> <----Main Storage (pages)----->
UserID   <(Percent)> T:V <Resident> Lock <-----WSS----->
/Class    Total  Virt Rat Totl Activ  -ed Totl Activ  Avg
-------- ----- ----- --- ---- ----- ---- ---- ----- ----
12:01:00 369.9 361.0 1.0  17M   17M  417  17M   17M 129K
 ***User Class Analysis***
*Servers  1.95  1.72 1.1 7566  7555   49 8674  7444  207
*Linux   184.0 180.6 1.0  15M   15M  305  15M   15M 185K
*Misc    183.7 178.5 1.0   2M 1642K   11   2M 1642K 328K
 ***Top User Analysis***
LXPWK001 183.5 178.4 1.0   2M 1641K    3   2M 1641K   2M
LXWKB215 37.63 37.01 1.0 782K  782K    1 782K  782K 782K
LXWKB211 33.97 33.88 1.0 514K  514K    0 514K  514K 514K
LXWKB210 17.64 17.55 1.0 298K  298K    2 298K  298K 298K
LXWKB214 16.86 16.68 1.0   1M 1188K    0   1M 1254K   1M
LXWKB228  6.01  5.98 1.0 731K  731K    3 731K  731K 731K
LXWKB222  5.06  4.94 1.0 621K  621K    5 621K  621K 621K
LXWKB183  4.70  4.57 1.0 231K  231K    0 230K  230K 230K
LXWKB220  3.69  3.66 1.0 125K  125K    8 124K  124K 124K
LXWKB225  3.65  3.52 1.0 780K  780K    0 780K  780K 780K
ESATCP    0.45  0.35 1.3 1038  1038    1 1037  1037 1037
TCPIP2    0.02  0.01 2.0 1142  1142   48  198   198  198
```

Measure CPU consumption:

- ESAUSP2 (Traditional)
- CPU Consumption (Percent)
  - Total of all users
  - By user
  - By class

Note:

- One server dominates the CPU

T:V Ration is Total to Virtual

- 1.0 is best

Limiting Users by Limiting Shares

```
Q  share vmservu
USER VMSERVU :RELATIVE SHARE= 100 MAXIMUM SHARE= NOLIMIT
Ready; T=0.01/0.01 16:58:54
```

Limits:

- LIMITHARD caps resource consumption regardless of other user demands
- LIMITSOFT caps resource consumption unless all users have received their target minimum and there are no unlimited users who can consume resources

Limits should only be used when truly understood…

```
set share vmservu relative 200 500 limitsoft
USER VMSERVU : RELATIVE SHARE= 200 MAXIMUM SHARE=LIMITSOFT RELATIVE 500
Ready; T=0.01/0.01 17:01:12

set share mvsys1 abs 5% abs 20% limithard
USER MVSYS1 : ABSOLUTE SHARE = 5%
MAXIMUM SHARE = LIMITHARD ABSOLUTE 20%
Ready; T=0.01/0.01 14:40:49
```

User complaints: “In Q” goes up
Check processor, CPU is constant, I/O is constant

Report: ESASSUM Subsystem Activity Velocity Software

	<---Users---->			Transactions	<Processor>		Storage (MB)		<-Paging-->	<-----I/O----->			
	<-avg number->			Per	Avg.	Utilization	Fixed	Active	<pages/sec>	<-DASD-->		Other	
Time	On	Actv	In Q	Minute	Resp	Total	Virt.	User	Resid.	XStore	DASD	Rate	Resp	Rate
14:01:00	1061	156	20.0	763.0	0.733	41	35	18.5	999.5	5	5	536	1.0	27.5
14:02:00	1063	157	25.0	803.0	0.594	41	35	18.5	1022.0	7	4	634	1.0	27.8
14:03:00	1064	188	52.0	981.0	1.112	41	35	18.5	1162.0	7	5	318	1.0	33.4
14:18:00	1064	154	31.0	729.0	1.055	41	36	18.5	986.5	0	3	277	1.0	26.3
14:19:00	1065	161	36.0	727.0	0.704	41	34	18.5	1061.1	226	3	303	1.3	35.3
14:20:00	1065	186	47.0	773.0	1.954	41	35	18.5	1315.9	432	2	377	1.1	30.8
14:21:00	1066	190	72.0	843.0	2.160	41	34	18.7	1308.9	1	2	769	0.8	38.9
14:22:00	1065	213	73.0	833.0	2.367	41	35	18.7	1394.9	1	3	548	0.9	31.1
14:23:00	1067	243	88.0	830.0	2.824	41	35	18.9	1537.0	1	3	858	0.8	29.8
14:24:00	1067	259	81.0	775.0	2.389	41	34	18.7	1660.4	13	3	683	0.8	18.2
14:25:00	1067	215	46.0	509.0	1.095	41	34	18.7	1452.4	8	2	583	0.8	28.5
14:30:00	1069	266	108	838.0	1.623	41	35	19.2	1618.2	5	3	511	0.8	28.8
14:31:00	1069	274	116	787.0	0.655	41	35	19.2	1630.7	8	3	569	0.8	29.0
14:32:00	1067	266	126	650.0	1.191	41	34	19.2	1580.9	4	3	774	0.8	30.7

Limiting Processor Case Study

Always understand at the high level first
Check LPAR configuration:

- Check weights
- VM shares with MVS and TEST – Share is 179 / (179 + 260 + 5) = 40%
- Only one CP defined
- VM LPAR is capped at 40% of one CPU!! VM is running 100%

```
Report: ESALPARS     Logical Partition Summary   Velocity Software
-------------------------------------------------------------------------------------
          <--Complex--> <-------Logical Partition--->  <-Assigned Shares---->            Proce
          Phys Dispatch              Virt <%Assigned>  <---LPAR--> <VCPU Pct> Cap-  Wait Type
Time      CPUs    Slice Name     Nbr CPUs Total  Ovhd  Weight  Pct /SYS /CPU  ped   Comp
--------  ---- -------- -------- --- ---- -----  ----  ------ ---- ---- ----  ----  ---- ----
14:01:00     1  Dynamic Totals:    0    3  80.4   0.5     444  100                          
                        VM         1    1  41.2   0.1     179 40.0 40.0 40.0   Yes    No CP
                        MVS        2    1  39.2   0.4     260 59.1 59.1 59.1    No    No CP
                        TEST       3    1     0     0       5  1.0 0.96 0.96    No    No CP
                        TESTTEST   5    0
```

Check User Wait States (ESAXACT)

- Running went down as a percent of non-dormant, inqueue time
- CPU wait stayed the same
- Asynchronous I/O wait is the bottleneck – but DASD I/O was constant?
- Clue – something was on the Limit List – this is a result of SHARE CAP
- Wait state sampling tests I/O wait before testing Limit. If I/O wait, then it stops

```
Report: ESAXACT      Transaction Delay Analysis                    Velocity Software
-------------------------------------------------------------------------------------
                     <---------------Percent non-dormant---------------------->        Times
UserID   <-Samples->                        E-  D-  T-     Tst <Asynch>        Lim  Pct    I/O
/Class    Total  In Q Run Sim CPU SIO Pag SVM SVM SVM  CF Idl I/O Pag Ldg Oth Lst Elig Throttl
-------- ----- ----- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---- -------
14:01:00  1061    20 5.0 5.0  40   0   0   0   0  10   0  35   0       .   0   0    0       .
Hi-Freq: 62599  1880 3.1 1.5  39 2.8   0   0  23 4.3 3.3  22 0.8   0   0   0 3.0    0       0
-------------------------------------------------------------------------------------
14:31:00  1069   116 0.9 0.9  34   0   0   0   0 1.7   0 3.4  59       .   0   0    0       .
Hi-Freq: 64140  7755 0.7 1.2  39 1.0   0   0 9.1 2.1 0.3 4.0  42   0   0 0.5   0    0       0
-------------------------------------------------------------------------------------
14:32:00  1067   125   0 4.0  46   0   0   0   0 2.4   0 5.6  42       .   0   0    0       .
Hi-Freq: 64020  7508 0.8 1.2  42 1.0   0   0 8.7 2.1 0.3 3.7  40   0   0 0.5   0    0       0
```

Check User Share settings (ESAUSRC)

- Cap on the database servers (soft cap applies if LPAR is at 100%)
- CPU consumption reaches a point where the database servers are limited
- Fall over the cliff
- Solution: Remove all caps – z/VM does a better job

```
Report: ESAUSRC      User Configuration
-------------------------------------------------------------
                                      <---------SHARE--------->
                  Account  ACI Grp    <Normal> <-Maximum>
UserID   ClassID  Code     Name        Rel Abs Type Share Limit
-------- -------- -------- -------- ---- --- ---- ----- -----
TIFSHRE  *BMAdmn  SYSTEMS          .  200   .  Abs  10.0  Soft
TIFSHRE2 *BMAdmn  SYSTEM           .  200   .  Abs  10.0  Soft
TIFSHRE3 *BMAdmn  SYSTEMS          .  200   .  Abs  10.0  Soft
TIFSHRE4 *BMAdmn  SYSTEM           .  200   .  Abs  10.0  Soft
```

Managing Virtual Processor Distribution

Managing Distribution – LPAR share of IFLs

- Based on weight of the LPAR
- Weight is divided by vCPU in the LPAR
- The more vCPUs, the less entitlement to each vCPU
- Horizontal vs Vertical using HiperDispatch

Managing Distribution – virtual machine share of LPAR

- Share is defined in relative or absolute
- Share is divided over the number of vCPUs
- The more vCPUs, the less entitlement to each vCPU

Affinity Processing

What is affinity processing?

- Virtual CPU will be dispatched on the same thread/CPU
- Theory: Reuses existing data in the hardware cache
- Theory: Reduces overhead and delays in re-loading cache
- Understanding this will pay your way to this free conference
- Reality: Probably good for z/OS
- Reality: Linux and TPF seriously break this model

How it works:

- Virtual CPUs are assigned to a PLDV - “Processor Local Dispatch Vector”
- Virtual CPUs stay assigned to the PLDV unless stolen
- Stealing is delayed 50 milliseconds to encourage affinity

Affinity Case Study

```
ESALPARS
VML1       06     2 IFL   200.1    0.0      Ded
```

```
Report: ESAXACT       Transaction Delay Analysis
-------------------------------------------------
                      <-----Percent non-dormant (Wait
UserID    <-Samples->                          E-  D-  T-
/Class    Total  In Q Run Sim CPU SIO Pag SVM SVM SVM
-------- ----- ----- --- --- --- --- --- --- --- ---
03/13/23
Hi-Freq:   3180  1576 8.3 0.1  12 0.1   0   0 8.0 0.6
Hi-Freq:   3180  1580 6.2 0.1 5.1   0   0   0  10 0.8
Hi-Freq:   3180  1572 4.1 0.1 2.8   0   0   0  12 0.6
Hi-Freq:   3180  1575 4.3 0.2 3.0   0   0   0  14 0.6
Hi-Freq:   3180  1579 5.2 0.1 5.1   0   0   0 8.7 0.6
16:06
Hi-Freq:   3180  1569 7.3 0.1  10   0   0   0  10 0.7
Hi-Freq:   3180  1568 6.8 0.1 8.4   0   0   0  10 0.4
Hi-Freq:   3180  1568 5.4 0.3 8.9   0   0   0  10 0.4
```

Why is CPU Wait high?

- High CPU Utilization?

Why is CPU Wait high?

- High CPU utilization? NO, 55% per thread
- Queuing theory – 1 server, 1 queue, 50% time in queue
- Queuing theory – 4 servers, 1 queue, 6% time in queue
- Affinity forces vCPU to stay on the PLDV

```
Report: ESACPUU      CPU Utilization Report
Monitor initialized: 03/13/23 at 16:00:00 on 3906 serial 06FCC8
-------------------------------------------------------------
         <----Load---->           <--------CPU (percentages)----
         <-Users-> Tran     CPU   Total  Emul  User   Sys  Idle
Time     Actv In Q /sec CPU Type   util  time ovrhd ovrhd  time
-------- ---- ---- ---- -  ----  ----- ----- ----- ----- -----
03/13/23
-------------------------------------------------------------
16:06:00   28 26.0  0.5  0  IFL    54.9  50.2   1.9   2.8  45.0
                         1  IFL    55.1  51.5   1.9   1.6  44.9
                         2  IFL    54.0  50.7   1.5   1.8  45.9
                         3  IFL    53.8  50.5   1.6   1.6  46.2
                                   ----- ----- ----- ----- -----
System:                           217.8 203.0   6.9   7.9 182.0
```

Modlevels – “Secret” Command…

```
q syscontrol
DISPATCH THDAFFINITY ON
DISPATCH PREEMPTLOCAL OFF
DISPATCH TSEARLY  50
DISPATCH INCHIPBUSY     50000
DISPATCH INCHIPDELAY    50000
DISPATCH INNODEBUSY    100000
DISPATCH INNODEDELAY   100000
DISPATCH INSYSBUSY     200000
DISPATCH INSYSDELAY    200000
Ready; T=0.01/0.01 11:24:20

CP SET SYSCONTROL DISPATCH MODLEVEL 0
Ready; T=0.01/0.01 11:24:24
q syscontrol
DISPATCH THDAFFINITY OFF
DISPATCH PREEMPTLOCAL ON
DISPATCH TSEARLY   0
DISPATCH INCHIPBUSY         0
DISPATCH INCHIPDELAY        0
DISPATCH INNODEBUSY     50000
DISPATCH INNODEDELAY    50000
DISPATCH INSYSBUSY     200000
DISPATCH INSYSDELAY    200000
Ready; T=0.01/0.01 11:24:27

CP SET SYSCONTROL DISPATCH MODLEVEL 1…
```

If in CPU Wait, but have excess capacity, less “affinity” is enforced when set to MODLEVEL 0