Processor Analysis and Tuning (2023)

Velocity Software Inc.
196-D Castro Street
Mountain View CA 94041
650-964-8867

Velocity Software GmbH
Max-Joseph-Str. 5
D-68167 Mannheim
Germany
+49 (0)621 373844

Barton Robinson,
barton@velocitysoftware.com

If you can't measure it, I'm just not interested….

Processor Topics

- Processor utilization
- CPU Management Hierarchy
- Managing processor allocations
- LPAR, HYPERDispatch, Horizontal
- Processor measurements
- Master Processor
  - PLDV, Dispatch rates

Processor Performance Concepts - Utilization

Utilization/ Capacity is important

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

Processor Utilization Hierarchy:

- TOTAL IFL Utilization (What is paid for, remaining capacity)
- LPAR Utilization (What is allocated / used by LPAR)
- Virtual Machine / Linux server (What is allocated to server)
- "My" share (is there enough provided for workload?)

CPU Utilization used for:

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

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 originally, now "steal timer"
- z/VM measures what? CPU Seconds (hardware timer)
- Hardware measurement only valid method of measuring CPU

Percent of Percent misleading

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

Processor Performance Reporting - Utilization

All zVPS numbers are measured in CPU Seconds

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

Impacts measurements of

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

Processor Distribution – CPU Hierarchy

At CEC Level:

- IFLs Shared or Dedicated at LPAR level

Shared Processor distribution "managed"

1) LPAR is assigned a weight
   - an "entitlement" of IFLs (ESALPARS)
2) Virtual Machine's assigned a "share"
   - A "share" of the LPAR (ESAUSRC/ESAUSP2)
3) Linux Processes have "priority",
   - Processes "prioritized", "nice" settings (ESALNXC/P)

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 (Emulation (z/VM Guest time))

- System time (kernel time)
- IRQ Time
- User time ("real application work")

IDLE

VELOCITY SOFTWARE

Process Share – CPU Hierarchy

Linux only measures Linux and "steal time"

- Bottom up analysis vs top down
- "top" gives limited view

Linux process top down analysis

1) Are engines on CEC highly utilized?
2) Is LPAR sufficiently entitled?
3) Is Virtual Machine Share sufficient?
4) Is process niced?

Managing Distribution - What is my share?

Entitlement / Share analysis:

- **ESALPARS for LPAR / IFL utilization**
- **ESACPUU / ESACPUA for z/VM perspective**
- **ESAUSRC for share settings**
- **ESAUSP2 for how much am I getting out of used**

LPAR VCPU Entitlements

Each LPAR gets a weight,

- each vcpu in lpar gets part of weight

LPAR's entitlement:

- (LPAR Weight) / SUM(LPAR Weights)

VCPU entitlement (horizontal):

- (LPAR entitlement) / (Number CPUs in LPAR)
- The more VCPU, the smaller the VCPU entitlement
- The more VCPU, the slower work will go

LPAR Analysis (ESALPARS)

- Total IFL utilization (of CEC, are there cycles to spare?)
- LPAR weight (entitlement)
- LPAR Utilization

VCPU Entitlement

- LPAR entitlement divided by number “cores” in LPAR
- More VCPU results in lower entitlement / performance
- Hiperdispatch corrects by “parking” VCPUs
- SMT threads share “core” entitlement

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	900	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	<---------CPU---------> Count	Ded	shared	<-Shared Processor busy-> 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

ZVMQA LPAR entitlement:

- 15% of 10 IFLs
- 1.5 IFLs

LPAR Configuration

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
--------  ---- -------- -------- --- ---- ---- -----  ----  ------ ----
00:15:06    23  Dynamic Totals:    0   22  CP  506.0   4.5      99  100
                        Totals:    0   73 IFL  903.1   8.6     100  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
  - ZVMQA Entitlement: **1.5** IFLs
  - VCPU entitlement: 1.5 IFLs / 6 vcpu (.25 IFLs)
- ZVMQA is **using** 375% SHARED IFLs (more than1.5)
- IFLs running 915/1000% (91.6%) busy
- ZVMPRD entitlement: 6.5 IFLs, using 3.3

LPAR Summary Report

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 16% of 8 CPUs (~1.2 CPU Entitlement)
Each virtual cpu allocated 2% of system (8 CPUs)
Each processor rated at 16% speed of real processor
(HiperDispatch modifies vcpu entitlement dynamically)

VELOCITY SOFTWARE

LPAR Weights Example

(Why HI/MED/LOW LPAR Designation)

ESALPAR (Partial report, horizontal scheduling)

Note each vcpu running at 10%?

z/VM can dispatch 8 concurrent virtual machines

- Less queueing, slower service
- But, each single vcpu runs "VERY slow"

			<--Logical->		<--------Logical Processor--------				
Time	Phys CPUs	Dispatch Slice	<-Partition> Name	No.	VCPU Addr	<%Assigned> Total	Ovhd	Weight	Cap-ped	Wait 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
					LPAR	**82.8**	1.4			

LPAR Share Example

Processor Details: If change to 4 logical processors:

- 4 LPARs active
- Total of all shares: 60
- z/VM Weight: 10 (out of 60)
- z/VM Logical Processors: 4
- Physical processors online: 8

Guaranteed processor share (speed)

- **((10 / 60) / 4) * 8 = .32**
- Real problem in many installations
- Why HYPERdispatch required, vertical scheduling

Too many logical processors will slow you down!

- Specifically the master processor….
- The same concept applies to Linux virtual processors

LPAR with HYPERDispatch

Stated Purpose of HYPERDispatch and vertical scheduler:

- Localize work to L1/L2 cache
- Reduces impact of installation configuration errors
- Increase weight for unparked engines in proportion

Impact

- Virtual CPUs disabled, share redistributed
- Unparked Engines have no preloaded hardware cache

```
ESAOPER:
07:00:41 CPU Park from 15 to  13 CPUUtil= "12.9",
07:00:43 CPU Unpark from  13 to  15 CPUUtil= "12.5
07:05:35 CPU Park from 15 to  13 CPUUtil= "12.2",
07:05:37 CPU Unpark from  13 to  15 CPUUtil= "12.0
07:05:53 CPU Park from 15 to  12 CPUUtil= "12.0",
07:05:55 CPU Unpark from  12 to  15 CPUUtil= "10.4
07:07:13 CPU Park from 15 to  13 CPUUtil= "12.5",
07:07:15 CPU Unpark from  13 to  15 CPUUtil= "11.9
07:07:19 CPU Park from 15 to  13 CPUUtil= "12.1",
07:07:21 CPU Unpark from  13 to  15 CPUUtil= "11.8
07:07:29 CPU Park from 15 to  13 CPUUtil= "12.1",
```

LPAR with HYPERDispatch

HYPERDispatch requires Vertical scheduling

High/Medium/Low obvious – Based on entitlement

Report: ESALPAR Logical Partition Analysis
Monitor initialized: 05/31/16 at 00:00:00 on 2827 serial 2F5A7

	<--Complex-->		<--Logical->		<----------Logical Processor-				
	Phys	Dispatch	<-Partition>		VCPU	<%Assigned>		VCPU	
Time	CPUs	Slice	Name	No.	Addr	Total	Ovhd	TYPE	Weight
07:15:00	19	Dynamic	VSSYSG	2	0	87.9	0.8	IFL	400
					1	89.3	0.7	IFL	400
.....									
					11	81.9	0.8	IFL	400
					12	77.5	1.0	IFL	400
					13	75.5	0.9	IFL	400
					14	60.9	0.7	IFL	400
						-----	----		
					LPAR	1245	11.8		
			VSSYS1	3	0	48.6	2.1	IFL	500
					1	35.5	1.6	IFL	500
					2	40.4	1.7	IFL	500
					3	38.9	1.5	IFL	500
					4	36.8	1.7	IFL	500
					5	38.8	1.7	IFL	500
					6	40.1	1.3	IFL	500
					7	32.5	1.3	IFL	500
					8	30.0	1.2	IFL	500
					9	18.6	0.9	IFL	500
					10	17.8	1.4	IFL	500
					11	0.0	0.0	IFL	500
					12	0.0	0.0	IFL	500
					13	0.0	0.0	IFL	500
					14	0.0	0.0	IFL	500
						-----	----		
					LPAR	378.1	16.4		

VELOCITY SOFTWARE

LPAR with HYPERDispatch

HYPERDispatch requires Vertical scheduling

- Now Exposed on ESALPAR
- To get more “Vertical High” requires higher weights
- See “CP SET SRM UNPARKING LARGE | Medium | Low”

```
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
```

VELOCITY SOFTWARE

LPAR Options

Time Slice: Dynamic, used exclusively

Weights: Sets entitlement between Logical Partitions

Virtual processors:

Capping

- Limits Assigned time to LPAR
- Useful for outsourcing, fixed contracts

Wait Completion

- "no" gives up processor if idle (default)
- "yes", Partition keeps processor even if idle (rarely/never used)

LPAR Overhead: Over-configure worst Case Study

Metrics to understand:

- Virtual machine VCPU / IFL
- LPAR VCPU / IFL

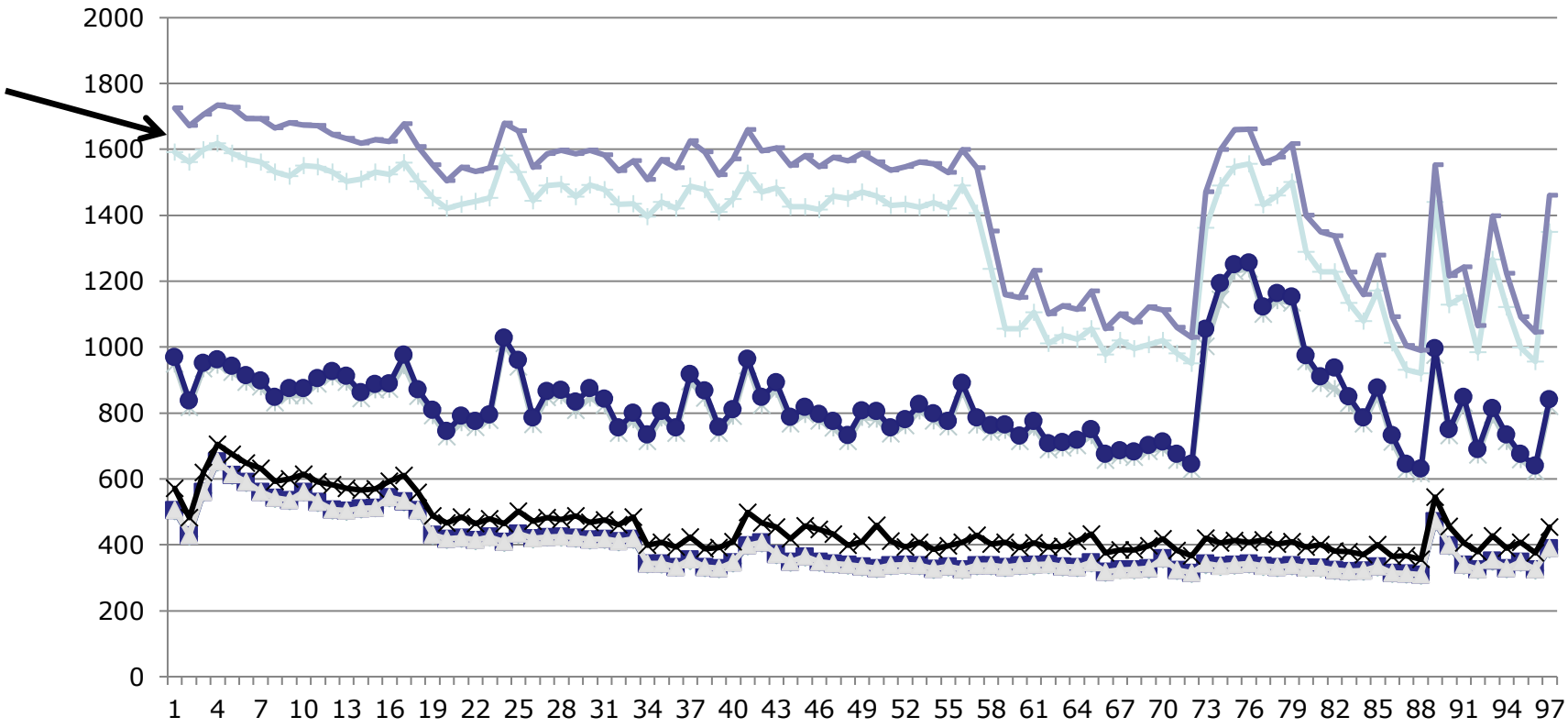
Linux Servers

- 120 servers total (Big, ORACLE)
  - 4gb-40gb
  - (1 / 2 size from original SUN servers)

VCPU Counts

- 17 IFLs
  - 7 servers per IFL, each server multiple vcpu, normal
  - 395 vcpus (**23:1** overcommit (23 Linux vcpu / real cpu)
- 7 LPARs, each with 17 VCPU
  - Worst case possible for physical LPAR overhead

LPAR Configuration Overhead



17 IFLs, 7 lpars, 17 vcpus each, 7:1 overcommit

Physical Overhead significant from real processor overcommit

VELOCITY SOFTWARE

z/VM Options: z/VM Shares

z/VM LPAR has entitlement of IFLs

z/VM virtual machines have SHARE of LPAR

- Each Virtual Machine assigned a relative or absolute share
- Each Virtual Machine has "normalized share"
  - Normalized = absolute
  - Normalized = (relative / (total relative)) * (100 – absolute)

Each vcpu has equal part of VM share

- Linux process running on virtual machine vcpu
  - Does NOT get virtual machine entitlement
  - Does get virtual machine vcpu share

"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)

Processor Measurement

```
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 master processor – only problem if architecturally constrained

Master Processor Overview

Every operating system has multiple methods

Much system code 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 database)
- running on the Master Processor

SPIN Locks

- Test for lock, if fail, test for lock
- Linux uses “spin lock”, replaced with Diag44 -> DIAG9C
- Linux spin locks an issue, cost in CPU

Diagnose Analysis

Linux diagnose for locking:

- 44 is high overhead, 9c is not

Two data sources, “system”, “user”

```
Report: ESADIAG
Date      CPU <--Total-->  ts per Second------------------
/Time         <Diags/Sec>  DIAG: Rate DIAG: RateDIAG: 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
```

```
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
```

Resource Serialization Master Processor

Many CP processes run "master only" to ensure integrity of 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

- higher System Overhead and
- Higher User Overhead.

Higher Master CPU busy higher on a system with more processors.

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

Master Processor 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?

Processor Dispatch Vector 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 “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: system provides affinity to maintain cache

VELOCITY SOFTWARE

Effects of Logical Partitioning – 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
```

The fallacy of not going top down

- You will have to explain this to Linux admins…

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

- Using 147% out of 300%

Next step, "zoom" to processor configuration

VELOCITY SOFTWARE

Effects of Logical Partitioning – Case study

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?

VELOCITY SOFTWARE

Effects of Logical Partitioning – Case Study

Report: ESAXACT Transaction Analysis Velocity Software, Inc.

			<-Percent non-dormant->					E-	D-	T-		Tst				Lim	Pct
UserID /Class	<-Samples-> 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 even greater.

VELOCITY SOFTWARE

Effects of Logical Partitioning – 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      e
-------- ---- ---- ---- ---- -------- -------- --- ---- ----- ---- ------ ---- ----
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 entitled to 50% of one real CPU, master processor is constrained
- Reducing CMS2 LPAR to 2 processors will perform better.

Hierarchy: z/VM Shares, Controls

Hierarchy:

Shared Processor distribution based "managed"

1) LPAR is assigned a weight
   - an "entitlement" of IFLs (ESALPARS)
2) Virtual Machine's assigned a "share"
   - A "share" of the LPAR (ESAUSRC/ESAUSP2)
3) Linux Processes have "priority",
   - Processes "prioritized", "nice" settings (ESALNXC/P)

Processor Measurements User View

Measure CPU consumption:

ESAUSP2:

CPU Consumption (percent)

- **Total all user**
- **By user**
- **By Class**

Note

- **one server dominates CPU**

T:V Ratio is Total to Virtual,

- **1.0 is best**

```
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
```

VELOCITY SOFTWARE

z/VM CPU Allocation

z/VM Allocates CPU based on SHARE

- ABSOLUTE SHARE is percent of LPAR
- RELATIVE SHARE is comparable to LPAR "weight"

SHARE is "normalized" to percent of system

- Normalized share is the "guarantee"

When to use Absolute vs Relative?

- If share should go up as workload increases (TCPIP,RACF) then use ABS
- If share should go down as more users logon, use REL

IBM Defaults are not optimum….

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 Only to 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
```

VELOCITY SOFTWARE

Limiting Processor Case Study

User complaints: InQ goes up

Check processor, cpu is a 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

VELOCITY SOFTWARE

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 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

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

- Cap on the database servers
- CPU consumption reaches point where 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
```

How Much Available Processor Capacity?

Processor utilization at CEC level shows box capacity

SMT provides additional capacity within LPAR

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
			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-------> Type	Count	Ded	shared	<-Shared Processor busy-> 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

VELOCITY SOFTWARE

LPAR Weights, parking

The Problem:

- If too many lpar virtual cpu defined, performance declines
- Cache competition, Overhead
- Errors on weight settings by installations

The solution: HYPERdispatch, implemented in z/OS, z/VM

- parking of low entitlement virtual cpus
- Parking level determined every 2 seconds…. – TOO MUCH PARKING
- Recommendation, 4 engines or less, horizontal is better (no SMT)
- SET SRM POLARIZATION HORIZontal | VERTical

```
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 Entitlement

Many installations have questions about parking

- Parking based on entitlement and srm setting
- LPAR weights define both
- Point of HYPERdispatch, parking is to reduce configuration errors

ESALPARS Logical Partition Summary **(141 IFLs on box, big z13)**

<-------Logical Partition------->				<%Assigned>		<-Assigned Shares----			
		Virt	CPU			<---LPAR-->		<VCPU	Pct
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
L1D1	01	50	IFL	167.0	10.1	500	13.0	0.26	35.5
L1D2	02	40	IFL	490.8	38.7	900	23.3	0.58	79.9
L1D3	03	30	IFL	1.2	0.4	50	1.3	0.04	5.91
L1D4	04	14	IFL	1.3	0.5	10	0.3	0.02	2.52
L1E1	05	20	IFL	64.8	3.6	500	13.0	0.65	88.7
L1C1	**11**	**40**	**IFL**	**3228**	**80.5**	**200**	**5.2**	**0.13**	**17.7**
L1C2	12	31	IFL	11.1	0.7	10	0.3	0.01	1.14
L1C3	13	14	IFL	1.4	0.5	10	0.3	0.02	2.52
L1C4	14	14	IFL	1.0	0.4	10	0.3	0.02	2.52
L1A2	22	2	IFL	1.0	0.4	10	0.3	0.13	17.7
L1B1	25	20	IFL	310.7	7.1	300	7.8	0.39	53.2
L1B2	26	31	IFL	99.0	5.5	700	18.1	0.58	80.1
L1B3	27	30	IFL	1.2	0.4	50	1.3	0.04	5.91
L1B4	28	14	IFL	1.0	0.4	10	0.3	0.02	2.52

(5.2% of IFLs = 7.1)

VELOCITY SOFTWARE

LPAR Parking Options

CP SET SRM UNPARKING LARGE | MEDIUM | SMALL

- LARGE (default): Unparks almost all, even vertical-low

CP SET SRM EXCESSUSE TYPE IFL HIGH | MEDIUM | LOW

- HIGH aggressively uses vertical-low even though not entitled
- Default is “MEDIUM”

CP SET SRM CPUPAD TYPE IFL 200%

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

LPAR Entitlement

Entitlement field added…

ESALPARS Logical Partition Summary

<-------Logical Partition------->				<%Assigned>		<-Assigned Shares---->				**Entitle**
		Virt CPU				<---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

VELOCITY SOFTWARE

Processor Summary

Processor efficiency has a price tag

- 100% is all you get

Performance options:

- LPAR Weights
- Virtual machine shares
- Capping
- Number and type of engines

z/OS engines about $1M fully loaded

vs IFLs $25K fully loaded….

- Big savings (97%?) to move workload

Managing Virtual Processor Distribution

Managing Distribution – LPAR Share of IFLs

- Based on weight of LPAR
- Weight divided by vcpu in LPAR
- More VCPUs, the less entitlement to each vcpu
- Horizontal vs Vertical using HYPERdispatch

Managing Distribution – virtual machine SHARE of LPAR

- Share defined in relative or absolute
- Share divided over number of vcpu
- More VCPUs, the less entitlement to each vcpu

Modlevels Secret Command..

If CPU Wait, but excess capacity, less “affinity” enforced

```
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
```

VELOCITY SOFTWARE

Common Reported CPU Performance Problems

Problems from "Linux perspective":

- Workload timing out
- Applications running slow
- Workload/Server in "CPU" wait (steal time high)

Analysis must be top down

- LPAR Weights vs IFL utilization (entitlement)
- LPAR VCPU vs SHARE (entitlement spread over more vcpu)
- z/VM Share settings poor (share spread over more vcpu)
- Operation on GP, not IFL (happens)
- Processor utilization high

Miscellaneous Causes – Workload

- z/VM Master processor
- Cron jobs synchronized (100 processes across 100 servers)
- Spin locks - Diag 44,9C (too many virtual machine vcpu)