

VELOCITY
SOFTWARE

Processor Analysis and Tuning

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*](mailto:barton@velocitysoftware.com)
If you can't measure it, I'm just not interested...

Copyright © 2019 Velocity Software, Inc. All Rights Reserved.
Other products and company names mentioned herein may be
trademarks of their respective owners.

- **What is CPU utilization**
- **Common problems**
- **LPAR, HYPERDispatch, Horizontal**
- **Overview of Processors**
- **Processor measurements**
- **Steal time**
- **Master Processor**
- **PLDV, Dispatch rates**
- **MFC, SMT**

Processor Performance Concepts - Utilization

What is important?

- TOTAL IFL Utilization (at 100%, everybody complains)
- LPAR Utilization (at 100%, everybody in lpar complains)
- "My" share (is there enough left for me?)

CPU Utilization used for:

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

Processor Performance Concepts - Utilization

What is CPU Utilization?

- Percent of what?

Utilization measured in many ways

- Virtual Linux measures what?
 - Percent of wall clock originally, now “steal timer”
- z/VM measures what? CPU Seconds
- 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 Concepts - 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

- LPAR
- z/VM Virtual Machines
- Linux processes
- zVSE Jobs/Partitions

All Levels of Hierarchy have Allocation Settings

- 1) CEC has some number of engines
- 2) Each LPAR has weight, results in "entitlement of engines"
- 3) Each Virtual Machine's "share" results in "virtual entitlement"
- 4) Processes in Linux have "priority", "nice" settings
 - What is my share of the CEC's engines?

Linux only measures Linux and "steal time"

- Bottom up analysis vs top down

If Linux process does not get enough CPU, Top Down:

- 1) Are engines on CEC highly utilized?
- 2) Is LPAR sufficiently entitled?
- 3) Is z/VM Share sufficient?
- 4) Is process niced?

Each LPAR has weight, results in “entitlement”

- Entitlement divided equally by VCPU
- Each VCPU in LPAR has part of entitlement
- **Work running on VCPU**
 - Does NOT get LPAR entitlement
 - **Does get vcpu entitlement**

Each Virtual Machine’s “share” results in “entitlement”

- Each Virtual Machine has “share”
- Each vcpu of virtual machine has equal part of share
- **Linux process running on virtual machine vcpu**
 - Does NOT get virtual machine entitlement
 - **Does get virtual machine vcpu share**

Processes in Linux have “priority”, “nice” settings

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)

Managing Processor Distribution

Objective: Operate at **high utilization**

- Requires management decisions, prioritization
- Alternative to management is more hardware/software

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**

LPAR Configuration

z/VM entitlement of IFLs (zvmqa, 15% of 10 IFLs)

Report: ESALPARS

Logical Partition Summary

Time	<--Complex-->		<-----Logical Partition----->						<--Assigned	
	Phys CPUs	Dispatch Slice	Name	Nbr	Virt CPUs	Type	<%Assigned> Total	Ovhd	<---LPAR--> Weight	Pct
00:15:00	23	Dynamic	Totals:	0	22	CP	506.0	4.5	99	100
			Totals:	0	23	IFL	903.1	8.6	100	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----->			<-Shared Processor busy->			
	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

SMT Configuration

Bug or feature? SMT enabled on CP? Overhead?

CPs: Core is assigned, two threads assigned, one idle

IFLs: If SMT is being used, "IDLE" will be less

<-----Logical Partition----->						<--Assigned Shares---->				<--Thread-->		Entitled	
Name	Nbr	Virt CPUs	CPUs Type	<%Assigned> Total	Ovhd	<---LPAR--> Weight	Pct	<VCPU Pct> /SYS /CPU	Idle	cnt	CPU	Cnt	
Totals:	00	250	CP	991.9	12.0	1000	100						
Totals:	00	12	IFL	81.6	2.2	1000	100						
RV3LNX1	31	4	IFL	19.1	0.7	300	30.0	7.50 60.0	9.37	2	2.40		
RV3MGTA	01	10	CP	6.1	0.2	38	3.8	0.38 14.4	5.88	2	1.44		
RV3MGTA	01	6	ZIP	0.3	0.0	127	12.7	2.12 59.3	0.29	2	3.56		
RV3MGTV	02	10	CP	5.7	0.2	50	5.0	0.50 19.0	5.53	2	1.90		
RV3MGTV	02	5	ZIP	0.4	0.0	76	7.6	1.52 42.6	0.36	2	2.13		
RV3MHWC	08	12	CP	7.3	0.2	19	1.9	0.16 6.02	7.11	2	0.72		
RV3MHWF	09	12	CP	9.2	0.4	166	16.6	1.38 52.6	8.86	2	6.31		
RV3MHWF	09	15	ZIP	0.4	0.0	179	17.9	1.19 33.4	0.40	2	5.01		
RV3MHHW	0A	38	CP	7.1	0.3	208	20.8	0.55 20.8	6.75	2	7.90		
RV3MHHW	0A	13	ZIP	0.7	0.0	120	12.0	0.92 25.8	0.67	2	3.36		
RV3MHWO	0C	38	CP	3.4	0.2	70	7.0	0.18 7.00	3.25	2	2.66		

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

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 is **allocated** 150/1000 of 10 SHARED IFLs (6 vcpu)
- ZVMQA is **using** 37.5% of 10 SHARED IFLs
- IFLs running 91.6% busy
- ZVMQA using more than entitled, ZVMPRD using less.

Each LPAR gets a weight,

- each vcpu in lpar gets part of weight

LPAR's entitlement:

- $(\text{LPAR Weight}) / \text{SUM}(\text{LPAR Weights})$

Processor share of system (**horizontal**):

- $(\text{LPAR entitlement}) / (\text{Number CPUs in LPAR})$

Processor share of a CPU is

- $(\text{Processor share of system}) * (\text{Number physical processors})$

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

Time	Phys CPUs	Dispatch Slice	Complex Name	Nbr	Virt CPUs	<%Assigned> Total	Ovhd	Assigned LPAR Weight	Pct	<VCPU 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

Each virtual cpu allocated 2% of system (8 CPUs)

Each processor rated at 16% speed of real processor

LPAR Weights Example

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"

Time	Phys CPUs	Dispatch Slice	<--Logical--> <-Partition> Name	<--Logical--> No.	<-----Logical VCPU Addr	<%Assigned> Total	Processor 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 – Why HYPERdispatch needed

Processor Details

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

Guaranteed processor share (speed)

- (Share of system / nbr logical processors) * nbr phys
- $((10 / 60) / 8) * 8 = .16$

**Each virtual cpu at peak runs at 16% rated speed
(go back to processor performance concepts)**

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
- Faster master processor for z/VM
- L1/L2 cache impact negligible in Linux environment

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

```

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

```

LPAR with HYPERDispatch

HYPERDispatch requires Vertical scheduling

- Now Exposed on ESALPAR

```
Report: ESALPAR          Logical Partition Analysis
Monitor initialized: 05/11/21 at 03:36:13 on 8561 serial XXXXXX
-----
-
      CEC  <-Logical Partition-> <-----Logical Processor-
-
Time      Phys      Pool      VCPU <%Assigned> VCPU Weight/
-----  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
```

Time Slice: Dynamic, used exclusively

Weights: Sets priority 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)

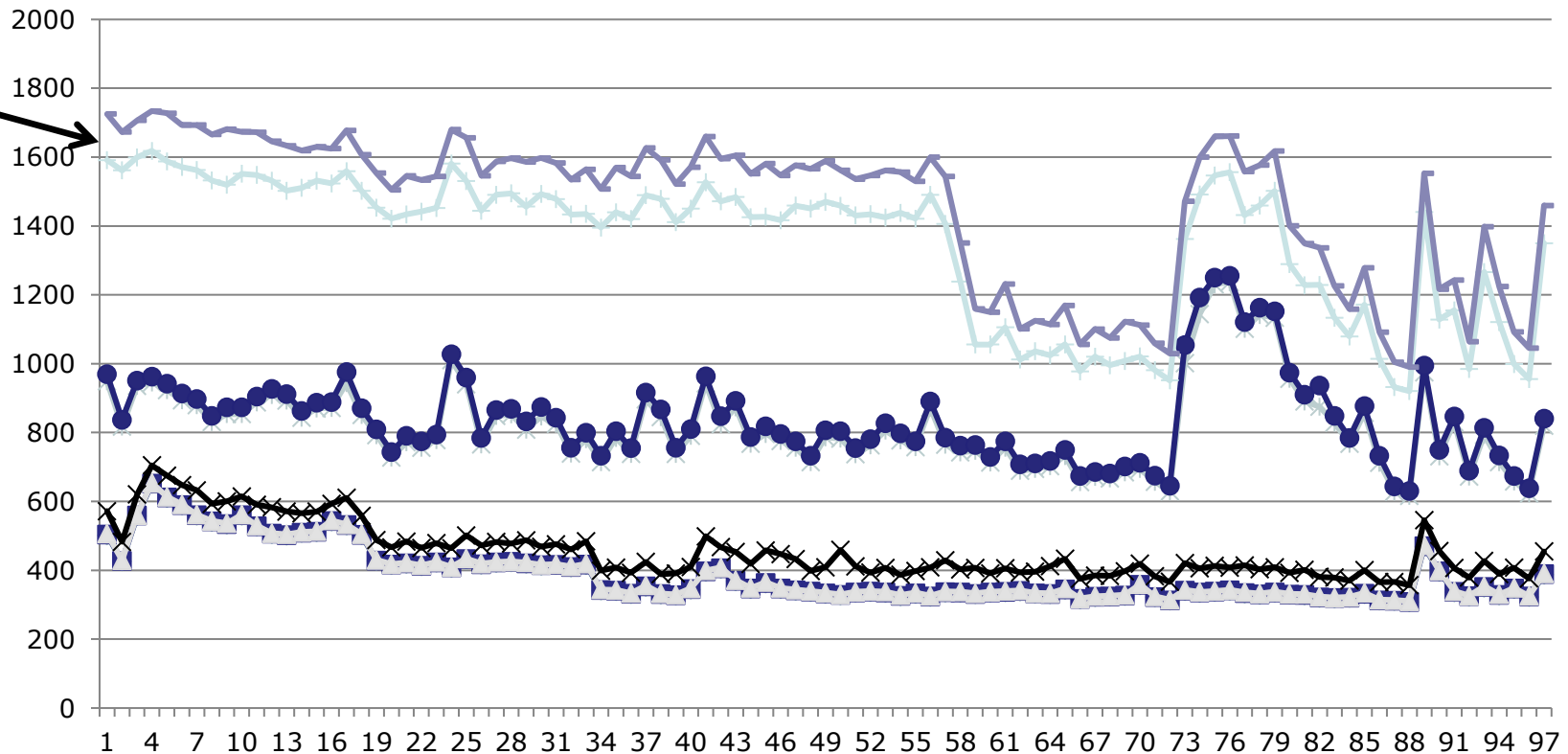
Linux Servers

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

Hardware

- 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 lpar, 17 vcpus each, **7:1 overcommit**

Physical Overhead significant from real processor overcommit

Managing Distribution - What is my share?

Entitlement / Share analysis:

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

"z" Processor Overview (ESAHDR)

```
Machine Model/Type                z13:2964/725
Multithreading Status:Enabled
System Sequence Code              000000000000B9177
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 (IFLs)

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

Processor Measurements User View

ESAUSP2:

CPU Consumption in 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
TCP/IP2   0.02  0.01 2.0  1142  1142  48   198   198   198
```

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

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
-----
```

Time	<----Load---->			<-----CPU (percentages)----->						<-----External (per second)----->					
	<-Users->	Tran		Total	Emul	User	Sys	Idle	<---Page---	<---Spool---	RSCH+				
	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
```

Time	<----Users----->			Tran /sec	CPU	<VMDBK Moves/sec>		<-----PLDV Lengths----->				Dispatcher Long Paths	
	Logged	Actv	In Q			Steals	To Master	Avg	Max	Mstr	MstrMa		%Empty
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.

Effects of Logical Partitioning – Case Study

Report: **ESASSUM Subsystem** Activity Velocity Software, Inc.

Time	<---Users--->			Transactions		<Processor>		Storage (MB)		<-Paging-->		<-----I/O----->		<MiniDisk>		Spool		
	<-avg number->	Per	Avg.	Utilization	Fixed	Active	<pages/sec>	<-DASD-->	Other	<-Cache-->	Page	Rate	Rate	Rate	%Hit	Rate		
	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

Effects of Logical Partitioning – Case study

Report: **ESACPUU CPU** Interval Analysis Velocity Software, Inc.

Time	<----Load---->				<-----CPU (percentages)----->						<---Internal (per second)---->				
	<-Users-> Actv	Tran In	Q /sec	Tran CPU	Total util	Emul time	User ovrhd	Sys ovrhd	Idle time	Diag- nose	Inst. sim.	SIE intrcp	Fast path	Page 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?**

Effects of Logical Partitioning – Case Study

Report: ESAXACT Transaction Analysis Velocity Software, Inc.

```

-----
                <-----Percent non-dormant----->
UserID   <-Samples->
/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 even greater.

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.

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 broken....

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

```
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
```

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						
Time	<---Users--->			Transactions		<Processor>		Storage (MB)		<-Paging-->		<-----I/O----->		
	<-avg number->	Per	Avg.	Per	Avg.	Utilization	Fixed	Active	<pages/sec>	<-DASD-->	Other	Rate	Resp	Rate
	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

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

Time	<--Complex-->		<-----Logical Partition---->					<-Assigned Shares---->				Proce		
	Phys CPUs	Dispatch Slice	Name	Nbr	Virt CPUs	<%Assigned> Total	Ovhd	<---LPAR--> Weight	<VCPU Pct> Pct	/SYS	/CPU	Cap- ped	Wait Comp	Type
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									

Limiting Processor Case Study

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----->																		
UserID	<-Samples->													<Asynch>		Lim		Pct		Times
/Class	Total	In Q	Run	Sim	CPU	SIO	Pag	SVM	D- SVM	T- SVM	CF	Idl	I/O	Pag	Ldg	Oth	Lst	Elig	I/O Thrott	
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	

Processor Case Study

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

UserID	ClassID	Account Code	ACI Grp Name	<-----SHARE----->				
				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

Processor Measurements SMT

```
Report: ESAUSR5          User SMT CPU Consumption Analysis
-----
          <----Raw CPU Seconds Consumed (Total)---->
UserID   <Traditional> <MT-Equivalent> <MT Prorated>
/Class   Total      Virt      Total      Virtual      Total      Virtual
-----
10:32:00 660.4    641.7    476.0     462.5     432.0     420.0
  ***User Class Analysis***
TheUsers 660.2    641.6    475.9     462.4     431.9     419.9
  ***CPU POOL User Analysis***
DB2      15.63    15.42    12.13     11.97     12.23     12.09
EEMSCSP  9.03     8.97     6.91      6.87      6.59      6.55
IIB      498.7    488.6    360.4     353.2     321.8     315.4
```

ESAUSR5/ESAUSP5 show SMT user data

Three CPU measures

- 1) Traditional: Time assigned and dispatched on a thread
- 2) Time Would take if non-SMT (MT-Equivalent)
- 3) Cycles really used (approximately, prorated)

How do you do capacity planning? What is 100% busy?

SMT Processor Utilization

SMT 2 has two threads

Always assigned concurrently

What if one thread is active, 2nd thread is idle?

Is "thread idle" additional capacity?

- It is not one for one. What does this mean?
- 384% - 10.0% assigned, 298% one thread idle....

<-----Logical Partition----->						<-Assigned Shares----->						
		Virt CPU		<%Assigned>		<---LPAR-->		<VCPU Pct>		Wait	<-Thread	
Name	Nbr	CPUs	Type	Total	Ovhd	Weight	Pct	/SYS	/CPU	Comp	Idle	c
-----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	---
Totals:	00	61	IFL	729.9	20.8	1001	100					
VLB4	04	14	IFL	384.1	10.0	475	47.5	3.39	122	No	298.5	
VLBX	0F	1	IFL	0.6	0.1	50	5.0	4.99	180	No	0	
VLB1	01	24	IFL	2401	0.1	Ded	40.0	0	0	Yes	0	
VLB6	06	14	IFL	326.5	10.3	475	47.5	3.39	122	No	254.4	
VLB8	FF	0										
ZS01	0E	8	IFL	18.7	0.4	1	0.1	0.01	0.45	No	17.79	

SMT – When to use it?

SMT Announced on z13 without valid guidance

A few installations said “good stuff”

- **Oracle, SAP workloads**

Others said “bad stuff”

- **Java, Websphere workloads**

The question is why?

And why is z14 so much better? (Prove it)

(z15 is better)

CPU Measurement Facility for z/VM

What is the CPU Measurement Facility (Basic) (smf 113)

CPI: Cycles per Instruction (RNI does not track....)

Report: ESAMFCA MainFrame Cache Hit Analysis
Monitor initialized: 12/10/14 at 07:44:37 on 282

Time	CPU	<CPU Busy>		<-----Processor----->			CPI Ratio
		Total	User	Speed/ Hertz	<-Rate/Sec-> Cycles	Instr	
07:48:35	0	20.8	18.4	5504M	1121M	193M	5.807
	1	21.6	19.6	5504M	1161M	221M	5.264
	2	24.4	22.5	5504M	1300M	319M	4.078
	3	22.4	19.7	5504M	1248M	265M	4.711
	4	19.6	17.6	5504M	1102M	194M	5.683
	5	20.4	18.6	5504M	1144M	225M	5.087
	6	23.9	22.0	5504M	1341M	341M	3.935
	7	17.6	15.4	5504M	949M	160M	5.927
	8	18.5	16.5	5504M	1005M	194M	5.195
	9	22.5	20.6	5504M	1259M	347M	3.629
System:		212	191	5504M	10.8G	2457M	4.733



EC12...

Why you should be interested – what is a MIP?

Report: ESAMFC MainFrame Cache Analysis Rep

Time	CPU	<CPU Busy>		<-----Processor----->			
		Total	User	Speed/ Hertz	<Rate/Sec-> Cycles	Instr	Ratio
14:05:32	0	92.9	64.6	5000M	4642M	1818M	2.554
	1	92.7	64.5	5000M	4630M	1817M	2.548
	2	93.0	64.7	5000M	4646M	1827M	2.544
	3	93.1	64.9	5000M	4654M	1831M	2.541
	4	92.9	64.8	5000M	4641M	1836M	2.528
	5	92.6	64.6	5000M	4630M	1826M	2.536
System:		557	388	5000M	25.9G	10.2G	2.542
14:06:02	0	67.7	50.9	5000M	3389M	2052M	1.652
	1	67.8	51.4	5000M	3389M	2111M	1.605
	2	69.0	52.4	5000M	3450M	2150M	1.605
	3	67.2	50.6	5000M	3359M	2018M	1.664
	4	60.8	44.5	5000M	3042M	1625M	1.872
	5	70.1	53.8	5000M	3506M	2325M	1.508
System:		403	304	5000M	18.8G	11.4G	1.640

1830 mips
(at 100%)

2828 Mips
(at 100%)
Doing 10%
more work



z13...

TLB Analysis – z13

DAT Translation consumes 30% of cycles for ONE thread

- Two threads on one core leaves very little for real work
- Problem is very high Linux polling / dispatch rates

Report: ESAMFC MainFrame Cache Magnitudes Report ZMAP 4.2.4

Time	CPU	<CPU Busy>		<----->		<- Translation Lookaside buffer (TLB) ->				CPU Cost	Cycles Lost
		Totl	User	Speed/ Hertz	Ratio	<cycles/Miss> Instr	<Writs/Sec> Data	Instr	Data		
07:45:01	0	25.9	24.4	5000M	1.704	159	742	473K	244K	19.77	257M
	1	35.9	34.7	5000M	1.491	138	731	530K	249K	14.17	255M
	2	15.8	13.9	5000M	2.868	206	826	419K	245K	36.30	289M
	3	16.6	15.4	5000M	2.508	212	825	411K	247K	34.90	291M
	23	18.1	17.0	5000M	2.144	197	815	412K	229K	29.44	268M
	24	21.4	19.9	5000M	1.865	114	533	598K	302K	21.35	229M
	25	26.2	24.9	5000M	1.742	98	503	736K	346K	18.71	246M
	26	12.9	11.6	5000M	2.050	154	631	378K	214K	29.92	194M
	27	13.1	11.9	5000M	1.987	156	630	378K	217K	29.64	195M
System:		514	476	5000M	2.257	176	724	14M	7641K	30.69	7917M

One Thread

TLB Analysis – Should SMT be Enabled?

Evaluate other data points from other z13 systems:

- z/VM Linux workloads issue: VERY HIGH dispatch
- Why z14 should be great....
- Don't enable SMT if one thread is consuming your DAT

Report: ESAMFC

MainFrame Cache Magnitudes Report

```
<CPU Busy> <---- <-Translation Lookaside buffer(TLB)->
<percent>   Speed <cycles/Miss><Writs/Sec> CPU  Cycles
##  Totl  User  Hertz  Instr Data  Instr Data  Cost  Lost
----  ----  ----  -----  -
Mem1  907   874   5504M      54    232    117M    36M  29.55  14.8G
Mem2  1188  1140   5000M     147    364     30M    26M  23.62  14.0G
VLB4  1703  1366   5000M     185    567     66M    46M  44.59  38.2G
z13N   216   212   5000M     192    598   3084K   1802K  15.94  1669M
TCPN   892   757   5000M     217    947     32M    17M  51.46  23.0G
MTRN   947   868   5000M     265   1283     33M    17M  65.25  30.8G ←
```

TLB Analysis – Should SMT be Enabled?

Z14 is “Awesome....”

- IBM doesn't sell value of z14 chip,
- DAT gives a lot of cycles back.
- 12% DAT cycles (SMT) vs. 30% z13, NO SMT....

ESAMFC MainFrame Cache Magnitudes Rate ZMAP 5.1.0
initialized: 04/08/19 at 19:00:00 on 39064/08/19 19:00:00

```
-----  
<CPU Busy> <-----Processor-----> <-Translation Lookaside buffer(TLB)->  
<percent> Speed/<-Rate/Sec-> <cycles/Miss><Writs/Sec> CPU Cycles  
CPU Totl User Hertz Cycles Instr Ratio Instr Data Instr Data Cost Lost  
-----  
0 29.5 28.0 5208M 1535M 822M 1.867 177 284 243K 364K 9.55 147M  
1 26.8 25.3 5208M 1399M 748M 1.871 178 294 248K 359K 10.71 150M  
2 37.3 35.1 5208M 1945M 877M 2.219 135 210 446K 818K 11.91 232M  
3 36.9 34.8 5208M 1925M 914M 2.107 136 212 449K 821K 12.19 235M  
4 22.6 20.9 5208M 1181M 530M 2.228 158 263 316K 445K 14.18 167M  
5 23.4 21.8 5208M 1219M 590M 2.066 156 260 316K 449K 13.63 166M  
6 23.9 21.5 5208M 1248M 615M 2.030 170 284 236K 364K 11.49 143M  
7 26.9 25.5 5208M 1402M 730M 1.921 166 265 237K 391K 10.19 143M  
8 31.2 29.5 5208M 1628M 792M 2.055 163 257 338K 507K 11.39 185M  
9 32.9 31.3 5208M 1715M 878M 1.954 159 247 326K 508K 10.34 177M  
10 20.9 19.4 5208M 1093M 504M 2.171 166 276 257K 391K 13.79 151M
```

Z13 Problem:

- z/VM does NOT support large pages
- Linux with java/websphere has VERY high dispatch
- Direct address translation (dat) required for all parts of instruction
- No cycles left after dat....
- IBM Hardware development “notified”

z14

- The fix seems like a secret
 - published by at least one IBMer for z/OS
- If one dat per core is the bottleneck, put on 4.....
- Z14 – no complaints about SMT
- Z15 the same

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
- CP SET SRM UNPARKING LARGE
- cp set srm excessuse type ifl high

```
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 Shares---->
      Virt CPU  <%Assigned>  <---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)

LPAR Entitlement

Entitlement field added...

ESALPARS Logical Partition Summary

```
-----<-----Logical Partition-----> <-Assigned Shares-----> Entitle
          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
```

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