

VELOCITY
S O F T W A R E

How fast is your EC12 / z13?

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

Copyright © 2015 Velocity Software, Inc. All Rights Reserved. Other products and company names mentioned herein may be trademarks of their respective owners.

What is CPU Utilization?

- What is important

What is CPU Measurement Facility

What makes your EC12 faster?

- **What do you need to know for z13?**

Measurements and results

YES, ZVPS (4.2) used for all measurements

One execution unit per IFL

- Two threads
- Neither will get 100%
- “idle time” on execution unit when thread takes cache miss

Objective:

- increase Instructions executed on execution unit

How much “idle” time is there?

- Cache miss
- Higher cache miss has more potential for improvement

Up to six instructions can be decoded per clock cycle.

Up to ten instructions can be in execution per clock cycle.

Instructions can be issued out-of-order.

Memory accesses might not be in the same instruction order (out-of-order operand fetching).

Several instructions can be in progress at any moment, subject to the maximum number of decodes and completions per cycle.

What is CPU Utilization used for?

- Capacity planning – How much resource for how much work
- Understanding performance
- Planning for z13....

What is important?

- Throughput (instructions executed, Cycles Per Instruction)
- Performance service levels

What are “MIPS”? Vs Gigahertz? **(CPI)**

- Barton's number 2003: 4 Mhz is about 1 mip
- (not 1.0 mip, 1 mip + or -)
- Based on measured workload on intel and p390

CPU Cycle time dropping, CPUs Faster

- 370/158, 8.696 MHz
- Z900 – 1.3Ghz (2002)
- Z990 – 1.2 Ghz (2004)
- Z9 – 1.7 Ghz (almost as fast as INTEL)
- Z10 – 4.4 Ghz (customer perception – SLOW)
- 196 – 5.2 Ghz
- EC12 5.5 Ghz
- Z13 – OOPS 5.0 Ghz

Question: Is the z13 faster than EC12?

Why you should be interested – what is a MIP?

Report: ESAMFC MainFrame Cache Analysis Rep
 Monitor initialized: 12/23/14 at 13:55:31 on 2964

```

-----
                <CPU Busy> <-----Processor----->
                <percent>  Speed/<-Rate/Sec->
Time           CPU Totl User  Hertz 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

CPU Cycle time dropping, Adding cache, function

- Z990: Faster (almost as fast as Intel)
 - (Execute up to 3 instructions per cycle)
- Z10: (more cache)
- 196: (more cache), Out of order execution
- EC12: (more cache), Enhanced out of order
- Z13: (more cache), Multi threads

LPAR – HiperDispatch

- attempts to align Logical CPs with PUs in same Book

Vertical vs Horizontal Scheduling

Affinity

Instructions per second = “work”

Cycles per instruction is “cost”

What helps (to reduce cycles per instruction)?

- Out of order execution
- Multi instructions per cycle

What hurts?

- L1 Cache miss
- L2 Cache miss
- TLB miss
- Memory access (20 cycles?)

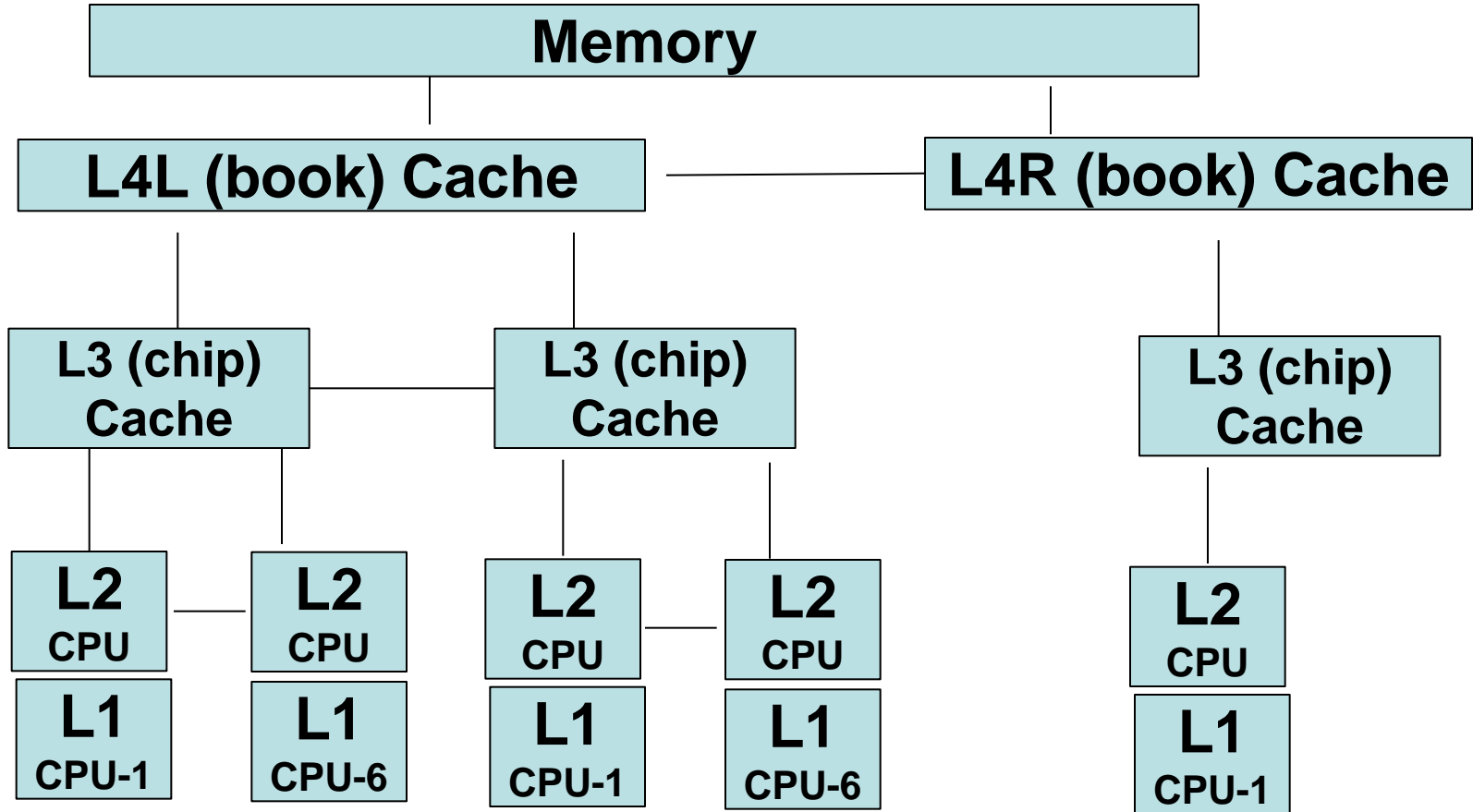
Up to 4 drawers, each SC (Drawer) up to 3 chips, 8 cores (PUs) per chip, 2 threads per core

- L4: 480 mb
- L3: 64mb (shared by 8 PU)
- L2: 2mb data, 2mb Instr
- L1: 96K Instr, 128K data

- Each L1 has TLB of 512 entries

▪ z13

$$\text{RNI} = 2.6 \times (0.4 \times \text{L3P} + 1.6 \times \text{L4LP} + 3.5 \times \text{L4RP} + 7.5 \times \text{MEMP}) / 100$$



What is the CPU Measurement Facility

- Hardware instrumentation
- Statistics by LPAR, all guests aggregated
- 5.18 Monitor records (PRCMFC) (Basic, Extended)
- “Extended” different for z10, 196, EC12 and z13
- Shows cycles used, instructions executed and thus CPI

```
Report: ESAMFC           MainFrame Cache Analysis Re
Monitor initialized: 02/27/15 at 20:00:00
```

```
-----
                <CPU Busy> <-----Processor----->
                <percent>  Speed/<-Rate/Sec->
Time           CPU Totl User  Hertz Cycles Instr Ratio
-----
20:01:00      0  0.7  0.4  4196M  30.8M  8313K  3.709
```

What is the CPU Measurement Facility (Basic)

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

```
-----  
                <CPU Busy> <-----Processor----->  
                <percent>  Speed/<-Rate/Sec-> CPI  
Time           CPU Totl User  Hertz Cycles Instr Ratio  
-----  
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
```

What is the CPU Measurement Facility (Extended)

Report: ESAMFCA MainFrame Cache Hit Analysis

Time	<-----Rate per 100 -----Data source----->			Instructions----->		
	L1	L2	L3	L4L	L4R	MEM
07:48:35	3.605	2.062	0.948	0.247	0.003	0.346
	3.281	1.935	0.831	0.195	0.002	0.319
	2.607	1.656	0.577	0.137	0.001	0.237
	2.913	1.678	0.786	0.249	0.002	0.198
	3.572	1.973	1.037	0.330	0.002	0.230
	3.188	1.815	0.889	0.272	0.002	0.210
	2.410	1.462	0.605	0.187	0.002	0.156
	3.729	1.793	1.220	0.654	0.035	0.026
	3.209	1.593	1.017	0.535	0.029	0.036
	2.182	1.222	0.602	0.307	0.018	0.034
System:	2.941	1.670	0.800	0.286	0.008	0.176

What to measure (Z10 – John Burg, WSC)

- L1MP – Level 1 Miss %
- L15P – % sourced from L1.5 cache
- L2LP – % sourced from Level 2 Local cache (on same book)
- L2RP – % sourced from Level 2 Remote cache (on different book)
- MEMP – % sourced from Memory

What to measure (EC12)

- L1MP – Level 1 Miss %
- L2P – % sourced from L2 cache
- L3P – % sourced from Level 3 Local (chip) cache
- L4LP – % sourced from Level 4 Local book
- L4RP - % sourced from Level 4 Remote book

- MEMP – % sourced from Memory

Cache sizes – EC12

- L1: 64k Instruction, 96k Data
- L2: 1MB Instruction, 1MB Data (private, cpu)
- L3: 48MB (Chip, shared 6 CPUs)
- L4: 384MB (Book, shared)

Cache Sizes – z196

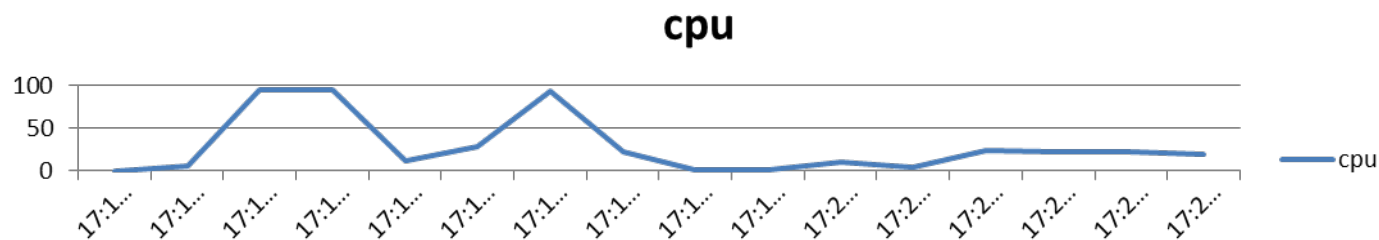
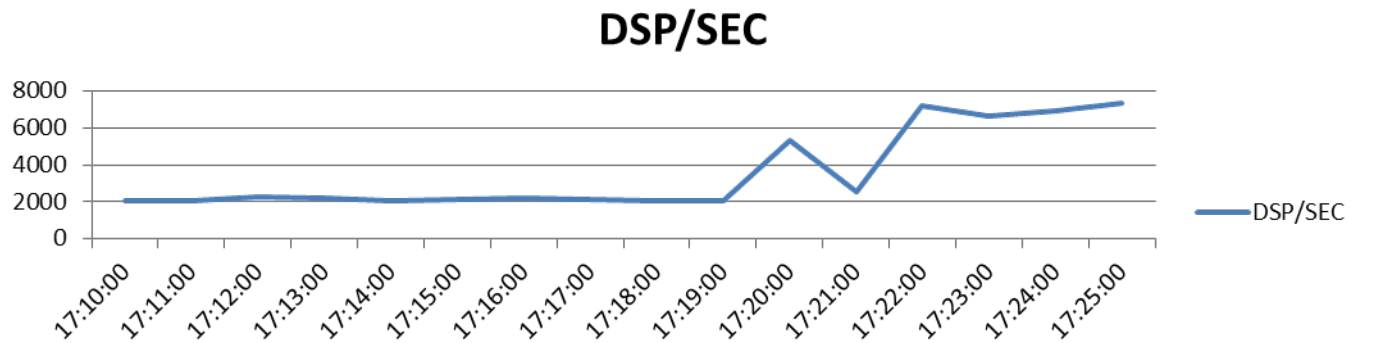
- L1: 64k Instruction, 128k Data
- L2: 1.5MB (private, cpu)
- L3: 24MB (Chip, shared 4 CPUs)
- L4: 192MB (Book, shared)

Cache Sizes – EC12 (pages)

- Level 1 cache, private (cpu), 16 pages instruction,
- Level 1 cache, private (cpu), 24 pages data
- Level 2 cache, private (cpu), 256 instruction
- Level 2 cache, private (cpu), 256 data
- Level 3 cache, shared (chip level) 12288
- Level 4 cache, shared (book) 98304

Workloads: Look at dispatches / second

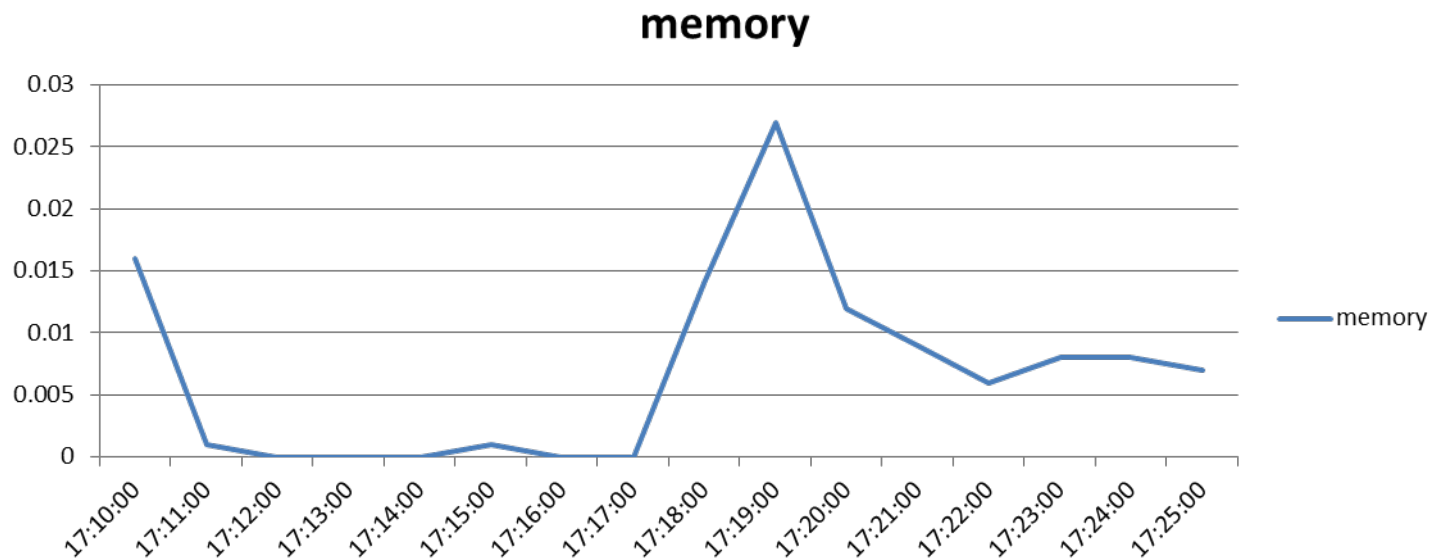
- 1st, pl1 loop (same as rexx loop)
- 2nd, run zmap against 50 sets of customer data



How long does cache last?

Average storage loaded per dispatch (idle time):

- Memory requests per 100 instructions: .01
- 2,000 dispatches per second
- Pages loaded from memory per dispatch: 66



MFC, memory loads?

Time	<CPU Busy>----->		CPI	<----Rate per 100 Instructions----->					
	<percent>			<-----Data source			read from----->		
	Totl	User	Ratio	L1	L2	L3	L4L	L4R	MEM
03/01/15									
17:10:00	0.4	0.1	4.105	5.100	3.521	1.301	0.262	0	0.016
17:11:00	6.5	6.2	1.501	0.131	0.094	0.030	0.006	0	0.001
17:12:00	94.9	94.6	1.431	0.019	0.015	0.004	0.000	0	0
17:13:00	94.6	94.3	1.432	0.019	0.015	0.004	0.000	0	0
17:14:00	12.0	11.8	1.470	0.078	0.057	0.017	0.003	0	0.000
17:15:00	28.4	28.1	3.995	0.086	0.063	0.018	0.004	0	0.001
17:16:00	94.1	93.8	4.001	0.028	0.022	0.005	0.001	0	0
17:17:00	22.4	22.2	3.991	0.106	0.075	0.026	0.005	0	0.000
17:18:00	0.5	0.3	3.430	3.204	2.288	0.751	0.151	0	0.014
17:19:00	0.6	0.3	3.517	3.179	2.251	0.750	0.151	0	0.027
17:20:00	11.4	9.8	2.032	1.506	1.267	0.177	0.049	0	0.012
17:21:00	4.0	3.3	2.298	1.999	1.649	0.279	0.061	0	0.009
17:22:00	24.1	22.4	1.935	1.233	1.039	0.121	0.066	0	0.006
17:23:00	23.2	21.4	1.953	1.213	1.017	0.124	0.064	0	0.008
17:24:00	22.4	20.8	1.950	1.235	1.038	0.121	0.068	0	0.008
17:25:00	18.8	17.0	1.937	1.382	1.168	0.143	0.063	0	0.007

Rexx loop

PL1 Loop – (check the other LPAR????)

zMAP batch

MFC, Double check

Screen: ESAMAIN Velocity Software - VSIVM4 ES
1 of 3 System Overview

Time	<---Users---> <-avg number-> On Actv In Q			Transact. per Avg. Sec. Time		<Processor> Utilization Total Virt.		
*-----	-----	-----	-----	-----	-----	-----	*-----	-----
17:26:00	140	71	27.0	27.1	0.23	2	86.5	83.9
17:25:00	140	80	31.0	27.0	0.27	2	35.8	33.1
17:24:00	140	75	34.0	27.8	0.40	2	36.1	33.4
17:23:00	140	73	47.0	25.9	0.34	2	35.5	32.8
17:22:00	140	88	29.0	26.4	0.31	2	35.6	32.9
17:21:00	140	73	31.0	26.9	0.27	2	35.4	32.7
17:20:00	140	82	28.0	25.9	0.25	2	35.0	32.3
17:19:00	140	87	30.0	25.8	0.34	2	36.7	33.9
17:18:00	140	77	31.0	25.2	0.29	2	34.0	31.3
17:17:00	140	91	31.0	25.1	0.30	2	35.2	32.4
17:16:00	140	72	29.0	25.1	0.32	2	74.5	71.7
17:15:00	140	81	28.0	26.8	0.29	2	34.7	32.1

Look at other LPAR

CPU Spike at 17:16

Run again: our “demo spike”

Yes, the “other lpar” makes a difference

MFC, memory loads?

Screen: ESAMAIN Velocity Software - VSIVM4 ES
1 of 3 System Overview

Time	<---Users---> <-avg number-> On Actv In Q			Transact. per Avg. Sec. Time		<Processor> Utilization CPUs Total Virt.		
*-----	-----	-----	-----	-----	-----	-----	*-----	-----
18:06:00	140	75	26.0	26.7	0.25	2	85.8	83.1
18:05:00	140	80	27.0	27.0	0.38	2	41.9	39.2
18:04:00	140	72	27.0	24.9	0.27	2	128.1	124.7
18:03:00	140	76	44.0	25.4	0.28	2	121.5	118.2
18:02:00	140	90	27.0	24.8	0.31	2	85.5	82.4
18:01:00	140	80	30.0	25.5	0.28	2	45.2	41.8
18:00:00	140	85	31.0	26.5	0.31	2	34.1	31.5
17:59:00	140	77	32.0	26.4	0.47	2	35.1	32.4

Time	<CPU Busy>-----> <percent> Totl User			<---Rate per 100 Instructions---> <---Data source read from---> CPI Ratio L1 L2 L3 L4L L4R MEM						
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
03/01/15 18:02:00	88.2	87.9	3.995	0.030	0.023	0.006	0.002			

Suse server: Look at other LPAR

CPU Spike at 17:16

Run again: This time oracle guest acting up

MFC, memory loads?

```
Report: ESAMFCA           Manalysis           Velocity Software
Monitor initialized: 03on 2828 serial 314C7    First record anal
-----
                <CPU Busy>-----> <-----Rate per 100 Instructions----->
                <percent>   CPI   <-----Data source read from----->
Time           CPU Totl User   Ratio L1    L2    L3    L4L    L4R    MEM
-----
09:24:00      0   0.6   0.4   3.443 3.216 2.304 0.747 0.141    0 0.025
09:25:00      0   0.6   0.4   3.371 3.154 2.296 0.690 0.142    0 0.026
09:26:00      0 61.0 60.7   3.998 0.047 0.035 0.009 0.002    0 0.000
09:27:00      0 83.6 83.3   4.003 0.042 0.031 0.010 0.001    0 0.000
09:28:00      0   0.6   0.3   3.386 3.156 2.294 0.699 0.140    0 0.024
09:29:00      0   1.6   1.3   2.387 1.459 1.105 0.274 0.058    0 0.022
09:30:00      0 27.2 26.9   1.439 0.040 0.030 0.008 0.001    0 0.000
09:31:00      0 93.2 92.8   1.432 0.019 0.015 0.004 0.000    0 0.000
09:32:00      0 87.0 86.7   1.433 0.020 0.015 0.004 0.001    0 0.000
09:33:00      0   1.8   1.4   2.345 1.378 1.051 0.251 0.059    0 0.017
09:34:00      0 72.8 72.5   4.003 0.042 0.031 0.010 0.002    0 0.000
09:35:00      0 93.9 93.6   4.006 0.035 0.026 0.008 0.001    0 0.000
09:36:00      0 93.4 93.2   4.001 0.031 0.024 0.006 0.001    0   0
```

Rexx consistently 1.4, PL1 consistently 4....

Looping program (BC12 IFL)

- 1.4 cycles per instruction
- Memory access zero by looper
- 2,000 dispatches per second for 'other work'

Idle Time Analysis

- 4 cycles per instruction
- 2,000 dispatches per second – Mystery work
- NOTE- z13 support adds vmdbk dispatch rate!!!!

ZMAP workload (99% of the load)

- 2.0 cycles per instruction – increased memory access
- 7,000 dispatches per second

Why is looper not CPI of 1:1?

Average time in dispatch (idle time):

- 2,000 dispatches per second
- 1% cpu utilization
- 5 microseconds per dispatch average
- Gigahertz: 4.196
- Cycles per microsecond: 4,196
- Cycles per dispatch: 20,000

- Memory requests per 100 instructions: .03
- Pages “touched”, loaded from memory per dispatch: 6

- The extra “.4” comes from the 2,000 dispatches/second

Average time in dispatch (load testing time):

- 15,000 dispatches per second
- 20% cpu utilization
- 13 microseconds per dispatch average
- Gigahertz: 4.196
- Cycles per microsecond: 4,196
- Cycles per dispatch: 260,000

- Memory requests per 100 instructions: .02
- Pages loaded from memory per dispatch: 50
- (Well tuned PL1 program)

IBM RNI calculations (per John Burg)

- **z196**

$$\text{RNI} = 1.67 \times (0.4 \times \text{L3P} + 1.0 \times \text{L4LP} + 2.4 \times \text{L4RP} + 7.5 \times \text{MEMP}) / 100$$

- **zEC12**

$$\text{RNI} = 2.3 \times (0.4 \times \text{L3P} + 1.2 \times \text{L4LP} + 2.7 \times \text{L4RP} + 8.2 \times \text{MEMP}) / 100$$

- **z13**

$$\text{RNI} = 2.6 \times (0.4 \times \text{L3P} + 1.6 \times \text{L4LP} + 3.5 \times \text{L4RP} + 7.5 \times \text{MEMP}) / 100$$

- **z10 RNI = (1.0 × L2LP + 2.4 × L2RP + 7.5 × MEMP) / 100.**

IBM RNI calculation analysis

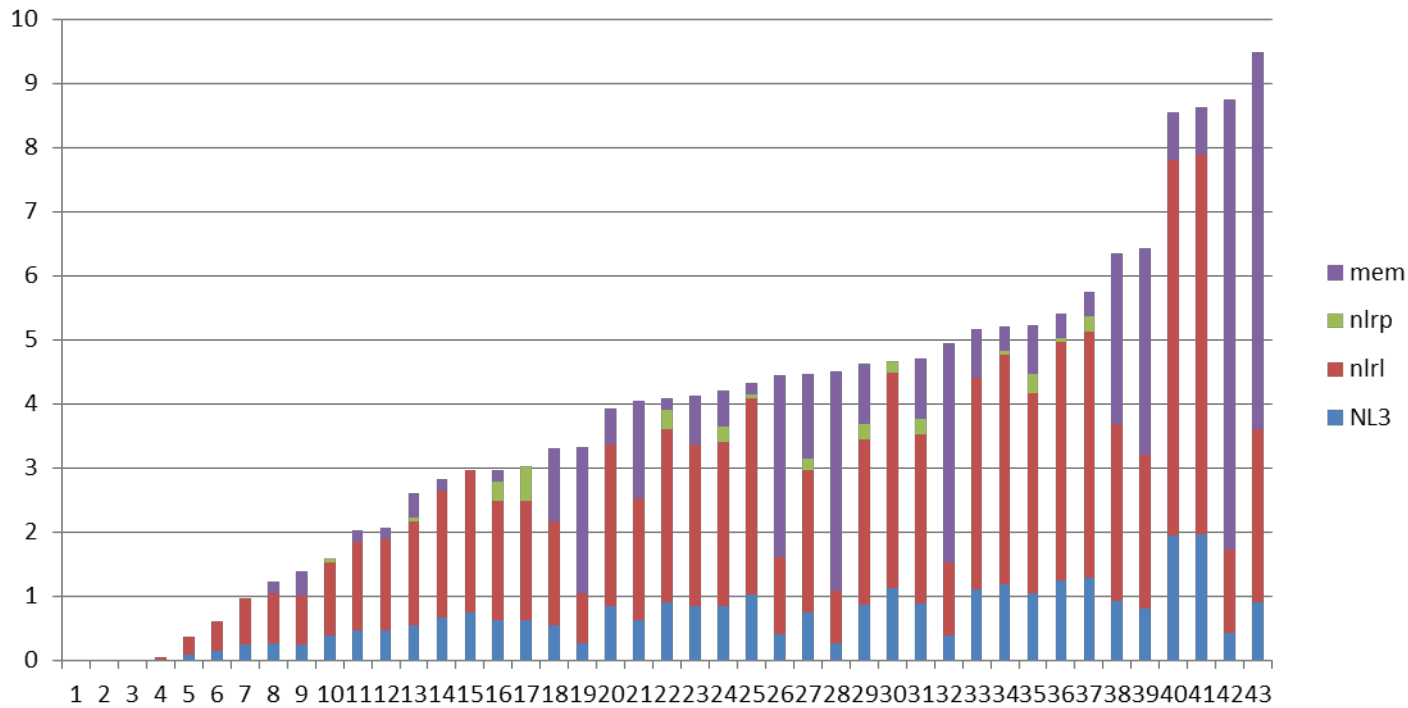
- **zEC12**

$$\text{RNI} = 2.3 \times (0.4 \times \text{L3P} + 1.2 \times \text{L4LP} + 2.7 \times \text{L4RP} + 8.2 \times \text{MEM}) / 100$$

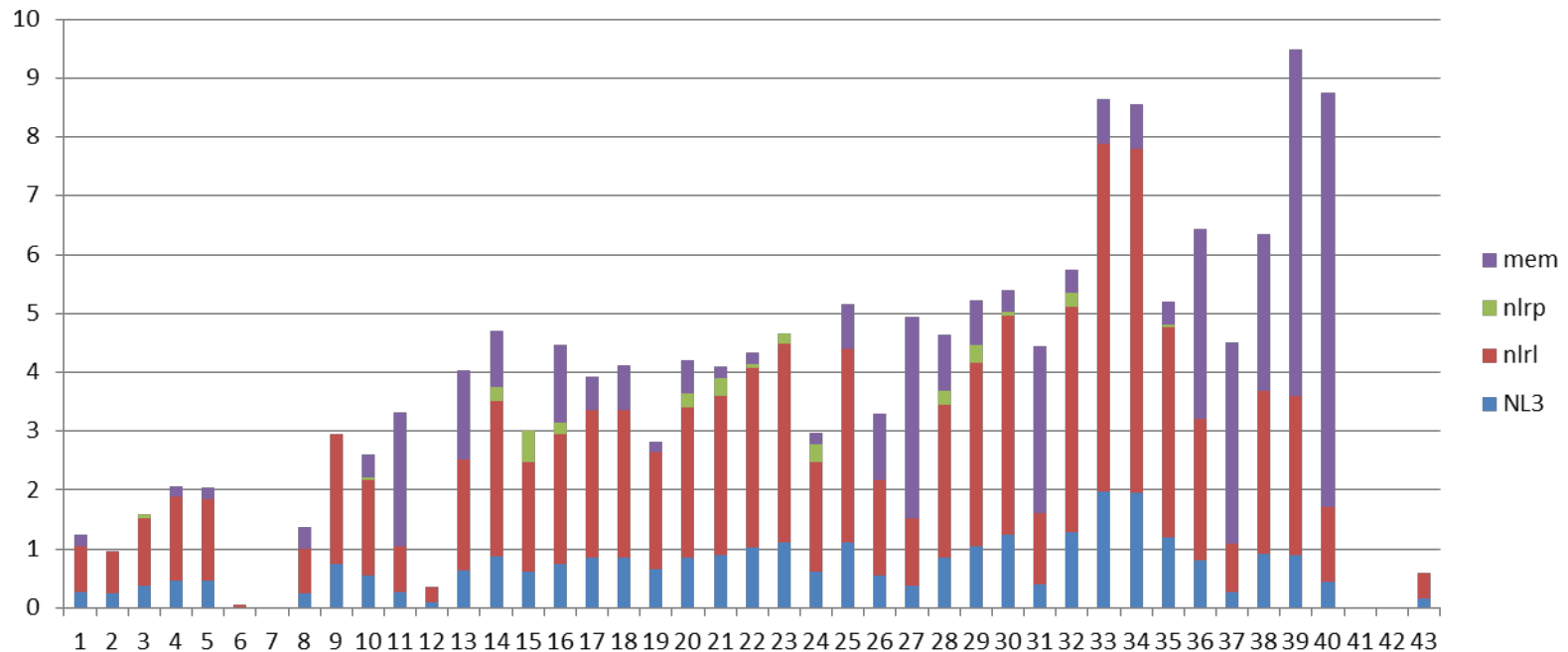
Cost analysis - ratio

- **L3P: 1**
- **L4LP: 3**
- **L4RP: 6**
- **MEM: 19**

Analysis, 50 LPARS, sort by Relative Nest Intensity
2 on right are z196 – (rest are ec12)

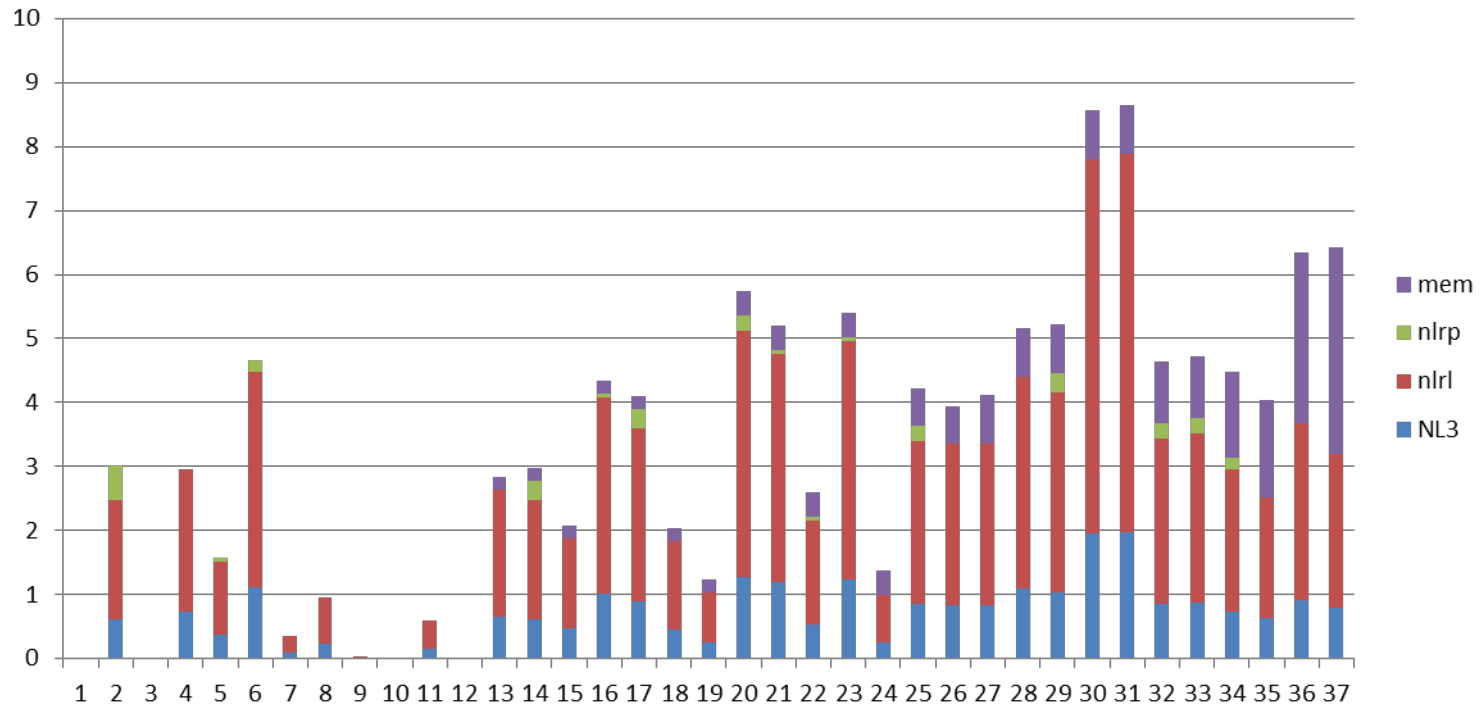


Analysis, 50 LPARS, sort by CPI
Slope of CPI and RNI “very close”
Higher RNI, lower mips



Impact of memory access?

Analysis, 50 LPARS, sort by “MEMP”, ec12 only
Common Characteristic of “high” ones???



EC12, 80 IFLs

LPAR: 32 IFLs (p210)

Report: ESAPLDV Processor Local Dispatch Vector Activity

```
-----
                                <-CPU Steals fr
Time          CPU  <VMDBK Steals  Moves/sec>  Dispatcher  <-From Nesting
-----          ---  -----  -----  -----  -----
14:06:00     0  3529.8      11.6      13104.2    1951  1198  380
              1  2908.6         0      11452.0    1626   976  306
              2  2751.9         0      10475.2    1630   855  267
              3  2671.7         0       9968.8    1653   775  244
              4  2522.7         0       9244.6    1615   693  215
              5  3382.6         0      13543.8    1259  1712  412
              6  2803.6         0      11705.6    1131  1339  334
              7  2460.3         0      10297.6    1114  1075  272
              8  3156.8         0      11949.7    1462  1366  329
              9  2702.0         0      10806.9    1283  1137  282
             10  2504.7         0       9849.8    1287   970  248
-----
```

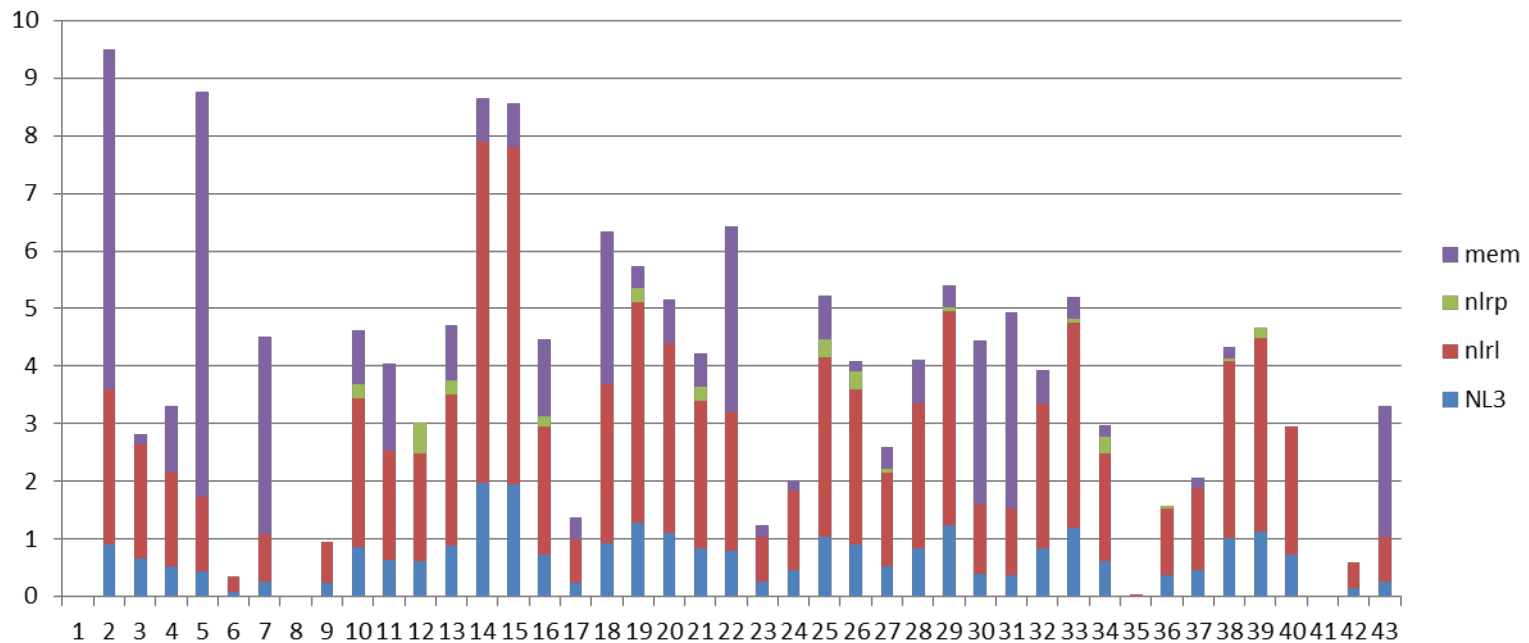
Steals: vmdblks moved to processor

Dispatcher Long paths: vmdblks dispatched

**Nesting level – CPU on chip, different chip(NL1),
different book(NL2)**

Analysis, 50 LPARS,

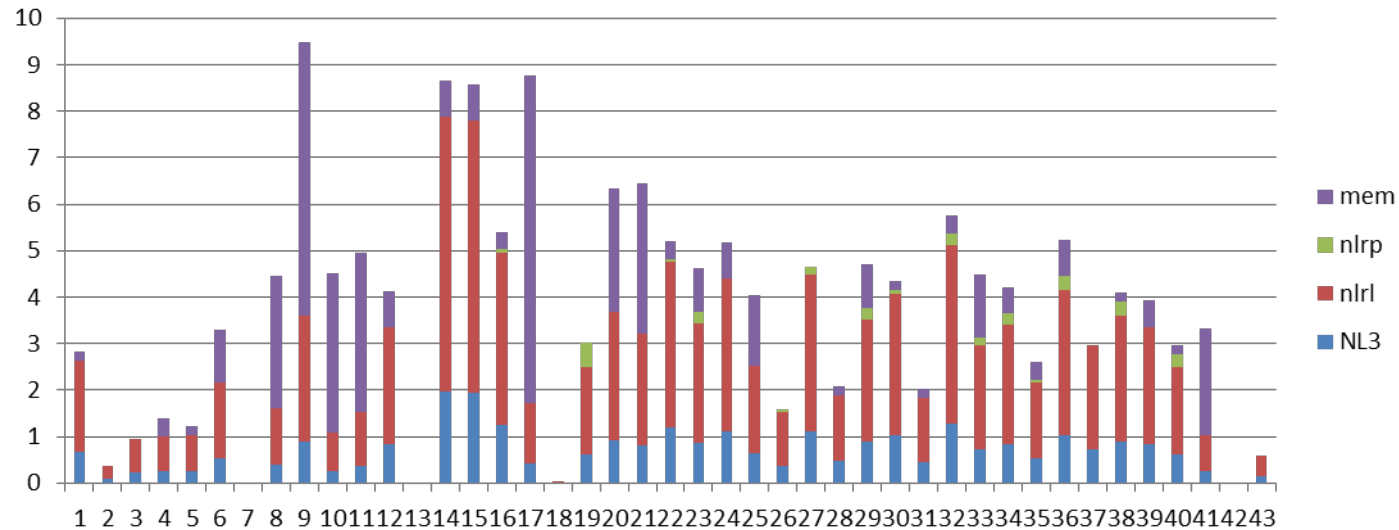
- sort by Dispatches/CPU/Second
- No expected pattern, more is better?



Impact of number of engines in LPAR

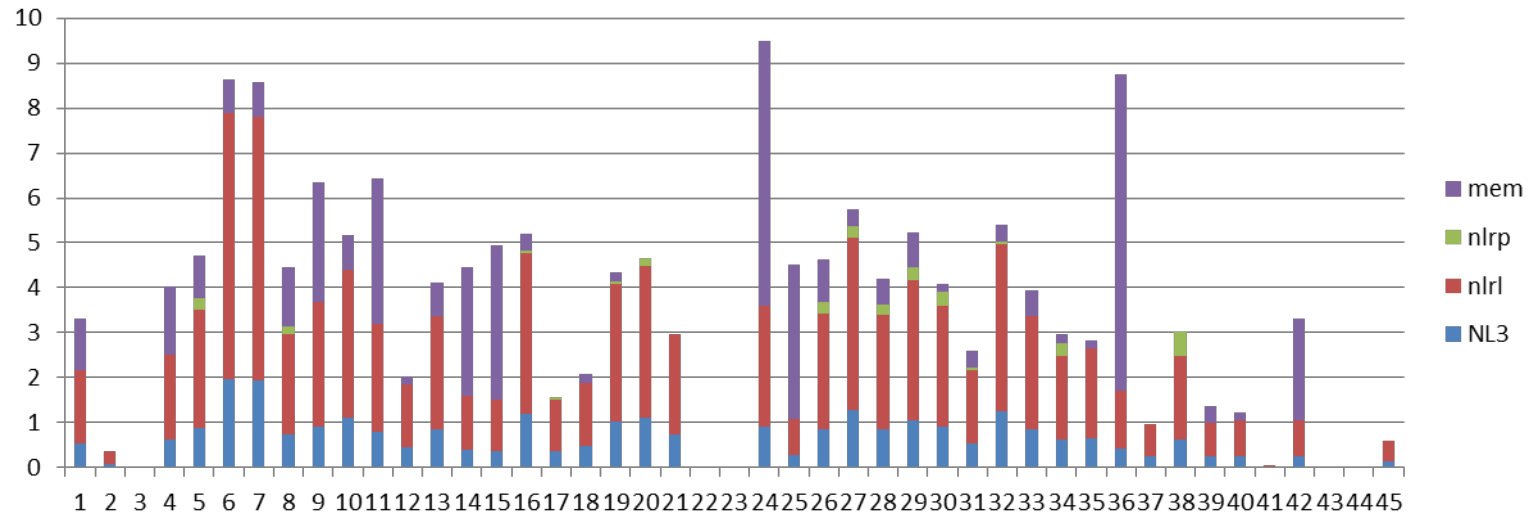
Analysis, 50 LPARS,

- sort by shared IFLs
- No expected pattern, more is better?



Analysis, 50 LPARS,

- sort by horizontal vs vertical
- No expected pattern, vertical slightly better?



CPU Cache Analysis

Report: ESAPLDV Processor Local Dispatch Vector Activity

Time	CPU	<VMDBK Steals	<Moves/sec> To Master	Dispatcher Long Paths	<-CPU Steals fr Same	<-From Nesting NL1	NL2
14:06:00	0	3529.8	11.6	13104.2	1951	1198	380

EC12, 80 IFLs

LPAR: 32 IFLs

Dispatch Analysis

- 13,104 dispatches per second per CPU

Steals: vmdblks moved to processor

Dispatcher Long paths: vmdblks dispatched

**Nesting level – CPU on chip, different chip,
different book**

CPU Cache Analysis

Report: ESAMFCA

MainFrame Cache Hit Analysis

Time	CPU	<CPU Busy <percent>		<-----Speed CPI		<----Rate per 100 Instructions			<-----Data source read from----->		
		Totl	User	Hertz	Ratio	L1	L2	L3	L4L	L4R	MEM
14:06:02	0	77.2	40.6	5504M	3.764	2.134	1.176	0.729	0.155	0.025	0.048
	1	76.1	42.5	5504M	3.625	2.112	1.183	0.714	0.146	0.022	0.046
	2	75.3	41.8	5504M	3.591	2.031	1.138	0.688	0.138	0.020	0.047
	3	74.8	42.3	5504M	3.539	2.001	1.118	0.679	0.136	0.020	0.048
	4	75.5	43.1	5504M	3.400	1.862	1.048	0.622	0.127	0.018	0.048

Cache source Analysis

- 3.7 cycles per instruction
- 2.1 % instructions from other level 1 cache on chip
- 1.2% instructions from other level 2 cache on chip
- .7% from level 3 on chip
- .2% from level 4 on book
- .02% from level 4 on remote book
- .05 % from memory (1 page from memory per 2,000 Inst)

BC12, 3 LPARS (2 physical IFLs)

- 1 IFL, z/VM 5.4
- 2 IFLs, z/VM 5.4
- 2 IFLs, z/VM 6.3

Run looper for 2 minutes on each

Expectation: less balance on z/VM 6.3

z/VM 5.4, looping on IFL, no affinity

Screen: ESACPUU Velocity Software - VSIVM4
1 of 3 CPU Utilization Analysis (Part 1)

Time	<--CPU-->		<-----CPU (percentages)----->			
	ID	Type	Total util	Emul time	Overhead User Syst	Idle time
07:35:00	1	IFL	16.3	15.0	1.0 0.3	83.1
	0	IFL	17.1	15.9	0.9 0.4	82.3
07:34:00	1	IFL	56.2	54.9	1.0 0.3	42.5
	0	IFL	60.1	58.9	0.9 0.3	39.0
07:33:00	1	IFL	67.0	65.8	0.9 0.3	32.1
	0	IFL	62.6	61.4	0.9 0.3	36.3
07:32:00	1	IFL	31.2	30.0	0.9 0.3	67.9
	0	IFL	34.0	32.6	1.0 0.4	65.5

z/VM 6.3, looping on IFL – a little affinity

Screen: ESACPUU Velocity Software ESA
1 of 3 CPU Utilization Analysis (Part 1) CPU

<----CPU (percentages)----->								
	<--CPU-->		Total	Emul	<-Overhd>		<CPU Wait>	
Time	Type	ID	util	time	User	Syst	Idle	Steal
-----	----	---	-----	-----	-----	-----	-----	-----
07:26:00	CP	0	10.5	7.5	0.5	2.5	89.3	0.1
	IFL	1	53.2	53.1	0.0	0.1	36.9	10.0
		2	34.5	34.3	0.1	0.1	56.8	8.7
07:25:00	CP	0	10.9	7.8	0.5	2.6	88.8	0.2
	IFL	1	55.5	55.3	0.1	0.1	40.4	4.1
		2	39.1	38.9	0.1	0.1	56.7	4.2
07:24:00	CP	0	10.0	7.7	0.5	1.8	89.8	0.2
	IFL	1	4.5	4.3	0.1	0.1	95.1	0.5
		2	10.0	9.8	0.1	0.1	88.9	1.1

Objectives of “affinity”

- Re-dispatch on same processor
- Utilize cache

Experiment: Looper for 2 minutes, measure CPU

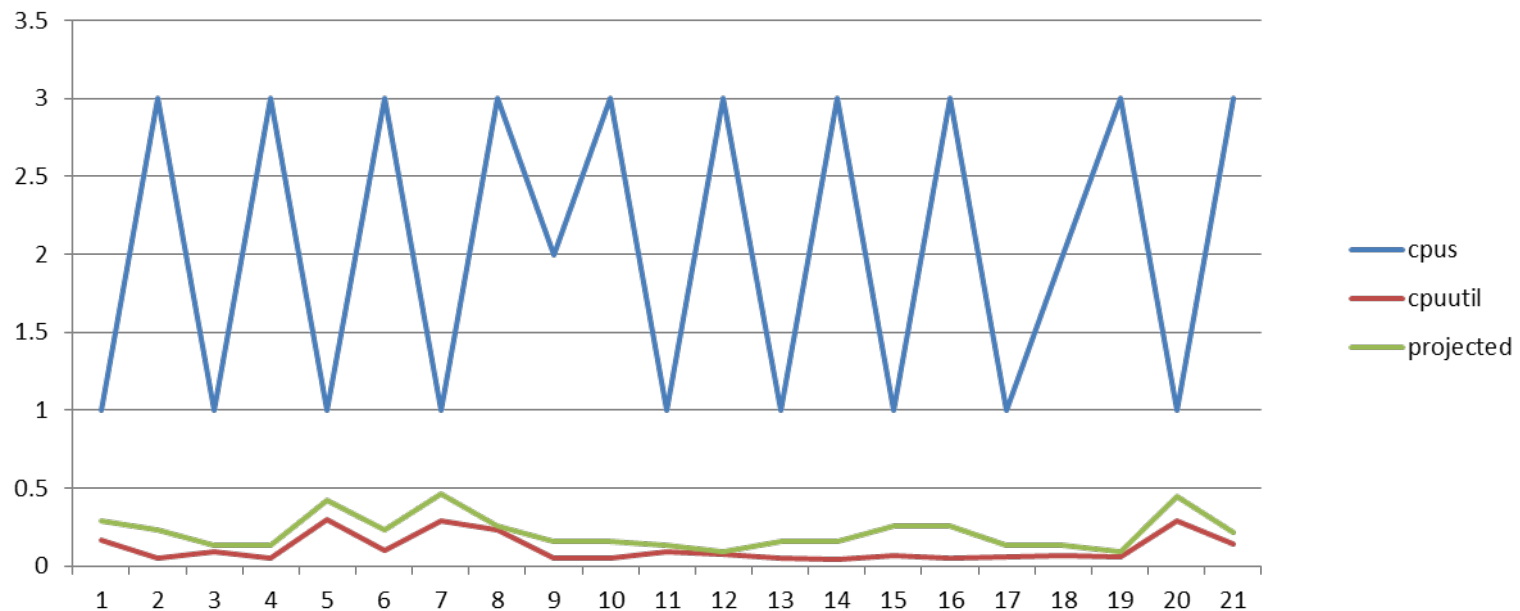
- z/VM 5.4, 1 IFL: 123.16 cpu seconds
- z/VM 5.4, 2 IFL: 123.14 CPU seconds
- z/VM 6.3, 2 IFL: 123.71 CPU seconds (ran twice)

Conclusion: 6.3 Affinity not helping with horizontal

Objectives of “vertical” scheduling

Localize work to cache

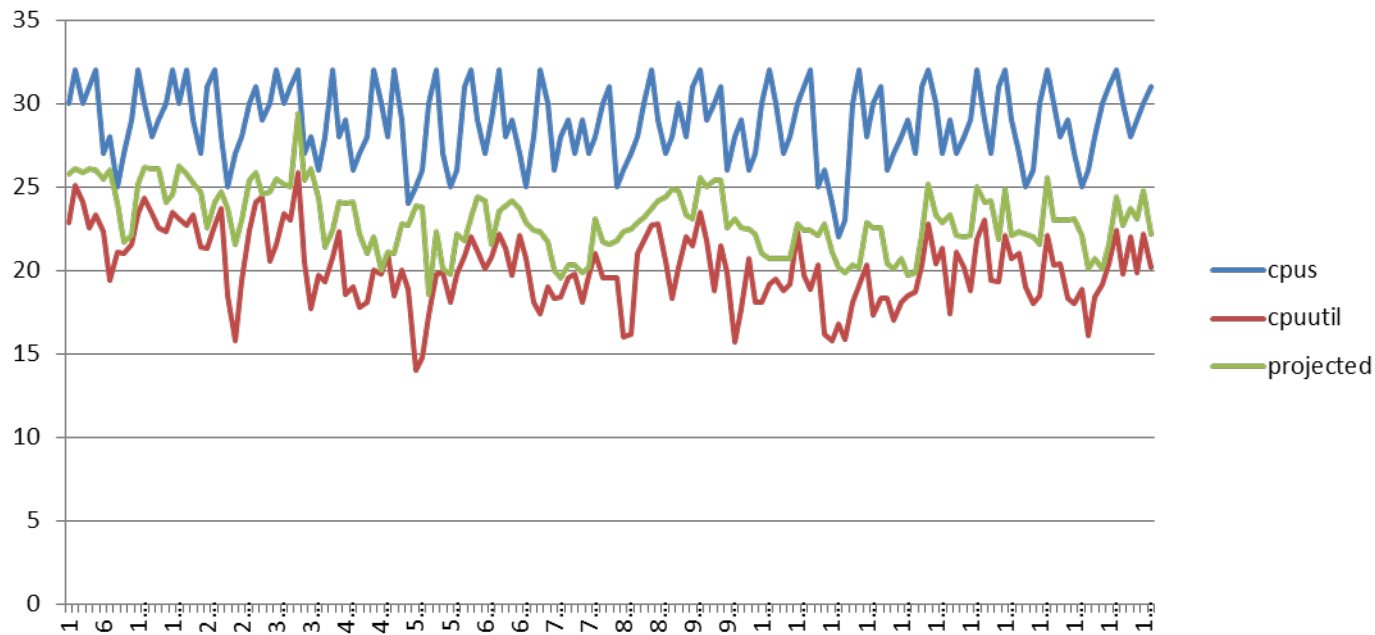
Monitor “event” – see parked cpus every 2 seconds



Objectives of “vertical”

Localize work to cache

Can we validate this has value?



Cycles per Instruction matters

Use of cache (all levels) has positive impact

- Need to dispatch fewer times on more processors
- Need further understanding

Please Send data for z13

Please come to z/VM Workshop

- June 25th to June 27th
- “[HTTP://VMWORKSHOP.ORG](http://vmworkshop.org)”

Velocity Software Performance Workshop

- June 23-24th