

Customer Experiences Saving MSUs Through CPC Optimization



Todd Havekost, IntelliMagic
Frank Kyne, Watson & Walker
SHARE Pittsburgh, Session 25634
August 7, 2019, Room 408



Agenda

- Key Processor Cache Concepts and Metrics
- Extent and Types of Savings Opportunities
- Detailed Customer Case Study
- Useful IBM Tools
- Highlights of z14 Processor Cache Design Changes

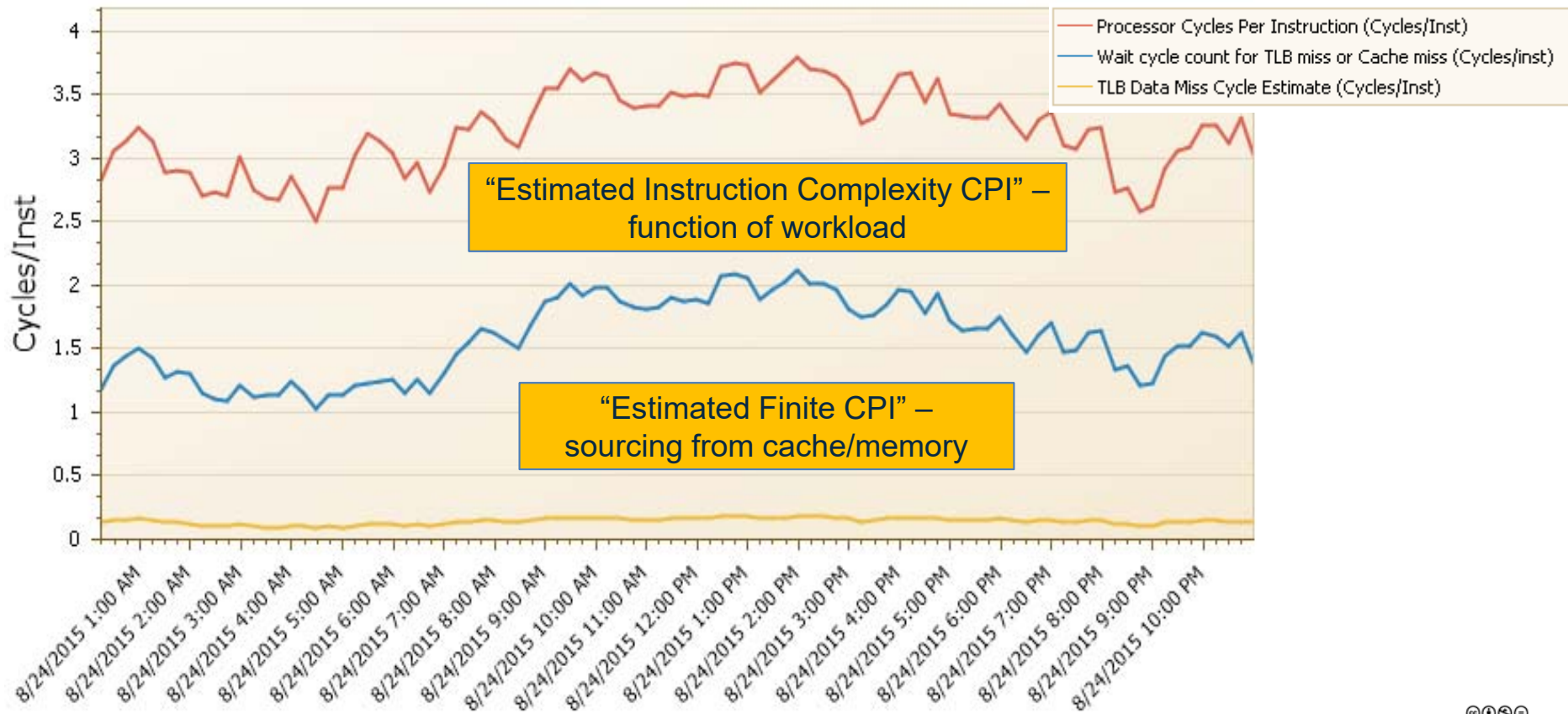


Key Processor Cache Concepts & Metrics

Cycles per Instruction (CPI)

- Number of processor cycles spent per completed instruction
- Processor cycles are spent
 - Productively – executing instructions present in L1 cache
 - Unproductively – waiting to stage data (L1 cache or TLB miss)
- Note: “Waiting” does not always mean waiting
 - Out Of Order (OOO) execution
 - Other pipeline enhancements

Cycles Per Instruction

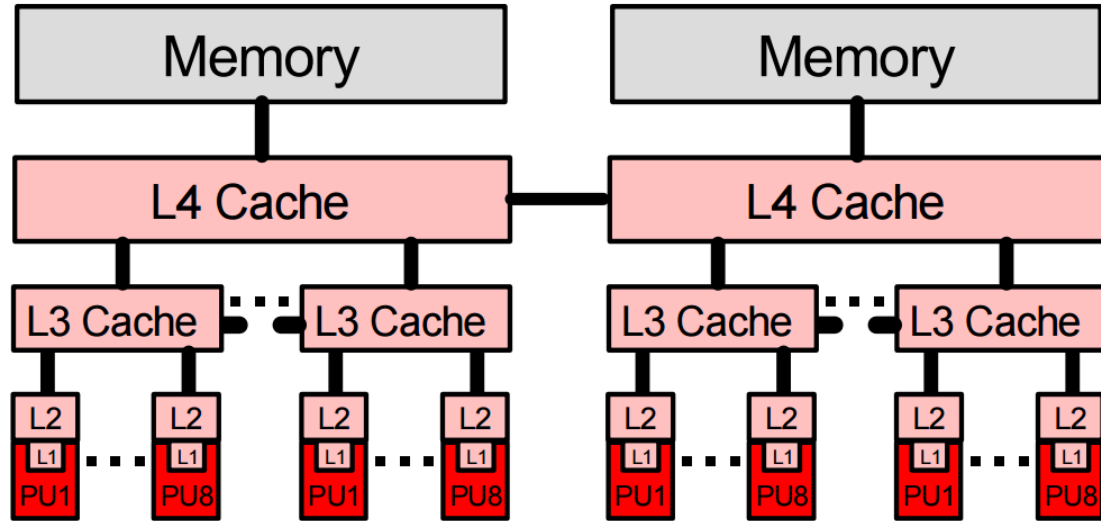


Relative Nest Intensity (RNI)

- How deep into the shared cache and memory hierarchy (“nest”) the processor must go to retrieve data
- Access time increases significantly with each additional level (increasing processor wait time)
- RNI formulas are processor dependent
 - z13: $2.3 * (0.4 * L3P + 1.6 * L4LP + 3.5 * L4RP + 7.5 * MEMP) / 100$
 - z14: $2.4 * (0.4 * L3P + 1.5 * L4LP + 3.2 * L4RP + 7.0 * MEMP) / 100$
- Reducing RNI improves processor efficiency

HiperDispatch

- Partnership between z/OS & PR/SM Dispatchers to align work to logical processors (LPs) & align LPs to physical CPUs
- Repeatedly dispatching the same work to the same or nearby CP is vital to optimizing processor cache hits



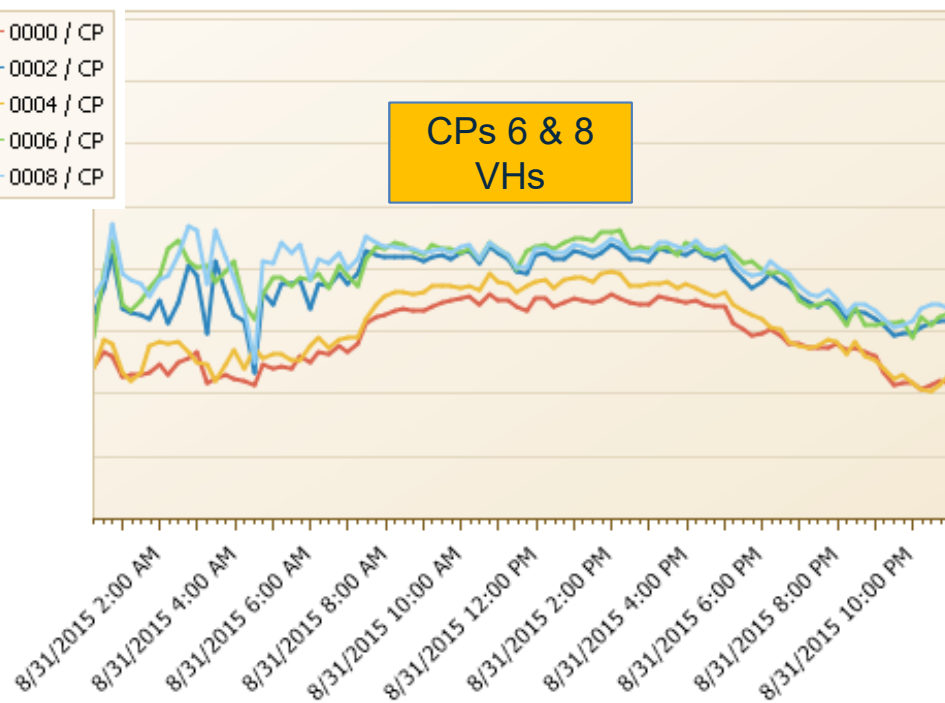
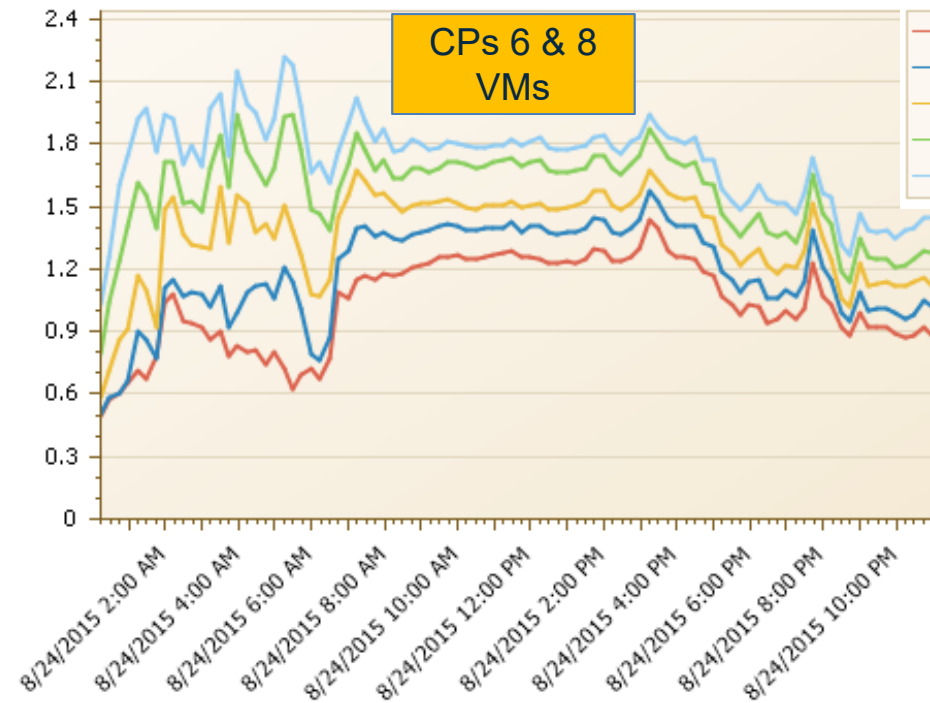
(© IBM)

Vertical CP Assignments

- Based on LPAR weights and the number of physical CPs PR/SM assigns logical CPs as
 - Vertical High (VH) – 1-1 relationship with physical CP
 - Vertical Medium (VM) – has at least 50% share of a CP
 - Vertical Low (VL) – no guaranteed share
- Work running on VHs has higher probability of cache hits
- Work running on VMs & VLs is subject to being dispatched on various CPs and contending with other LPARs

RNI Impact by Logical CP

Change: -19.11% Absolute change: -0.25



Optimizing Processor Cache – Recap

- CPU consumption decreases when unproductive cycles waiting for data to be staged into L1 cache are reduced
 - “Waiting” cycles represent significant component of total CPU
 - RNI metric correlates to unproductive waiting cycles
 - Reducing RNI reduces CPU (and thus MLC software expense)



Extent and Types of Savings Opportunities

Example 1

- The case that really brought these concepts to the forefront of z/OS performance discussions – Todd's work to address z13 performance issues in USAA.
- USAA moved from zEC12 711s to z13 711s and experienced an increase of 4K MIPS to do the same work.
- Moving from the z13 711s to 716s resulted in a net 5K decrease in MIPS consumed compared to the zEC12s.
- Moving from the 716s to 726s resulted in a *further* decrease of 4K MIPS.
- In addition to adding engines, they also optimized the LPAR and memory configuration.
- The bottom line is that optimizing the caches and the LPAR topology enabled USAA to reduce SW costs by 9K MIPS to do the same work.

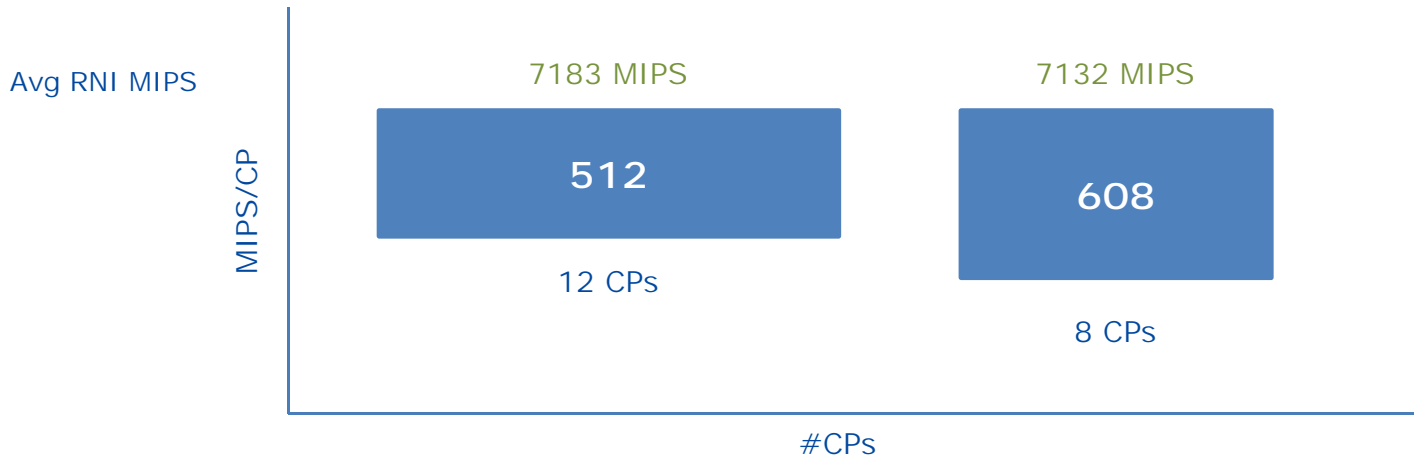
Example 2

- Large, European-based, international bank.
- Running zEC12s at the time.
- Turned on all CPs on CPC in preparation for DR test.
- Observed an immediate 25% drop in Actual MSUs as reported in RMF Partition Data Report.



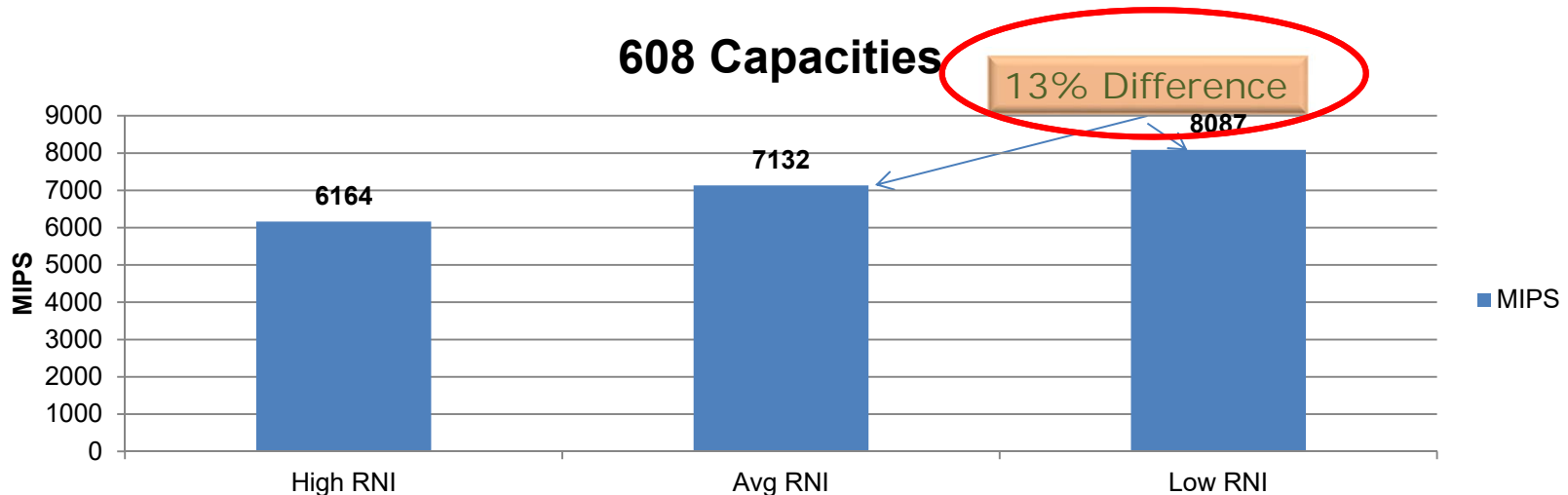
Example 3

- Regional bank in Europe.
- Normally run as z13 512s, but during performance test, one CPC is 'upgraded' to a z13 608.



Example 3

- ‘Upgraded’ CPC had approximately the same capacity, and ran roughly the same workload volumes as normal, however it had 1/3 fewer CPs.
- During performance test, RNI of production systems disimproves from average of 0.8 (Low RNI) to 1.0 (Avg RNI).



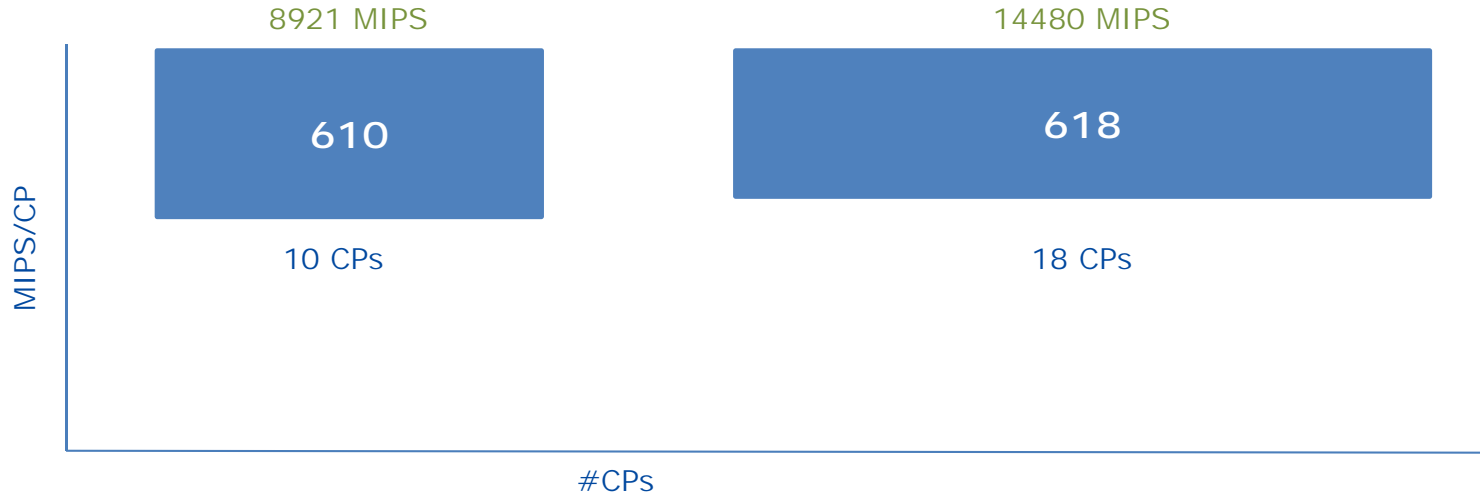
Example 4

- Large American Health Insurance company.
- Replaced zEC12-712 with z13-623.
 - *Total* MIPS increased from 14,166 on 712 to 17,020 on 623.
 - MIPS *per CP* dropped from 1180 on 712 to 740 MIPS on 623.
- Based on Engine speeds, you would expect CPU time for a given job to increase by about 59%.
- *Actual* CPU time increase varied by between 24% and 52%.
 - And despite the slower CP speed, batch job elapsed times *decreased* by an average of 5%.
- Overall, *actual* observed capacity was >25% more than projected by zPCR.
 - But upgrade included lots of additional memory, so that likely accounted for part of the improvement as well.

Example 5

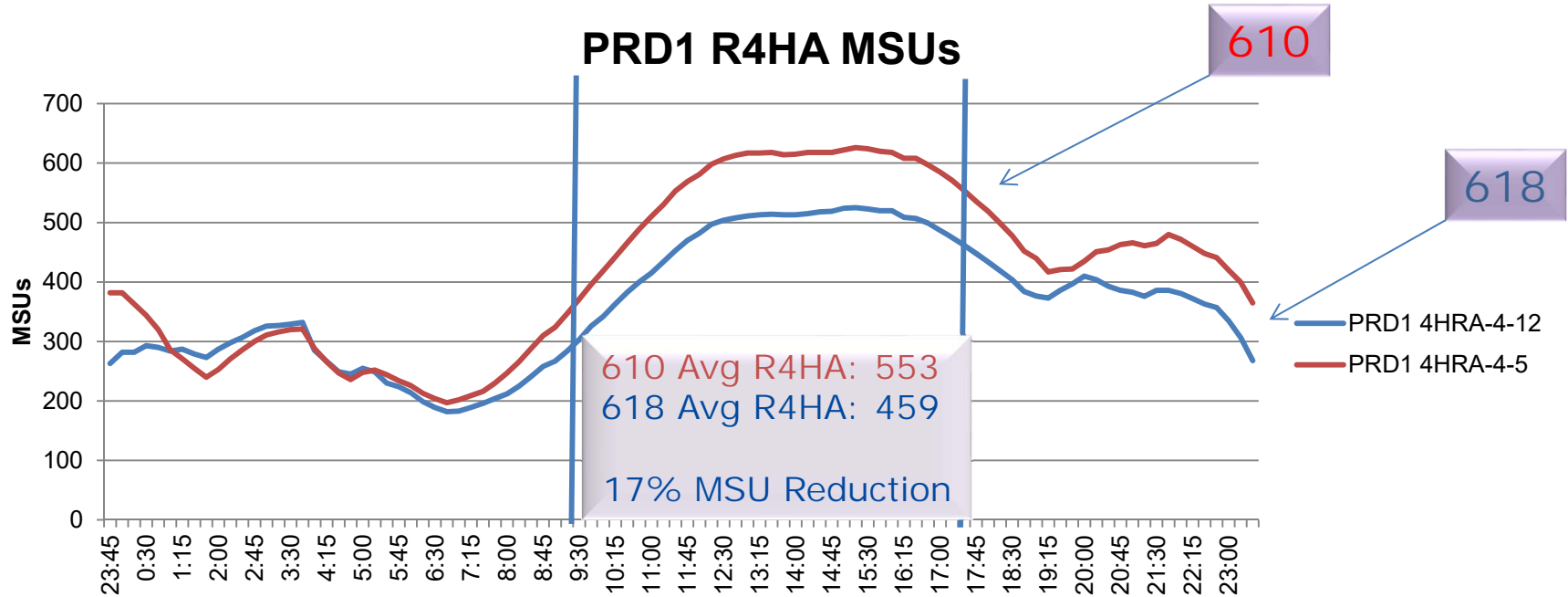
- African Financial Institution
- Normally run with z14 610s. One CPC temporarily upgraded to 618, but no workload change.

Avg RNI MIPS



Example 5

- Main Production LPAR



Example 5

- Based on LSPR numbers for Average RNI Workload on 610 & 618:
 - 610 MIPS – 8921
 - 618 MIPS – 14480
 - 618 utilization drop – $(1 - (14480 - 8921)) = 38.4\%$
 - For an average RNI Workload – $38.4\% * .4 = 15.4\%$ decrease.
 - Observed decrease was 16.99%.
- In *this* case, most of the MSU drop was because of the lower utilization on the 618.
 - Nearly all of the PRD1 work was already running on VH CPs, so adding more VH CPs really didn't help much.
 - Adding more logical CPs to the LPAR caused it to overflow to a 2nd chip, even though the additional capacity was not required.

Would you like to Help?

- In all these examples, the result was *generally* what you would expect, but we have no way today to provide a better prediction than “probably better” or “probably worse”.
- Neither zPCR or zCP3000 attempt to model impact of lower utilization or changing the stress on the caches.
- IBM’s David Hutton is helping us better understand what is going on, but we need more hard (SMF) data and less anecdotal results.
- If you would like to help us, please email us at technical@watsonwalker.com and we can explain what we need.



Detailed Customer Case Study

See “Customer Sub-capacity CPC Experience” article in Cheryl Watson’s Tuning Letter 2018 No. 3 for additional information.

CEC Configurations

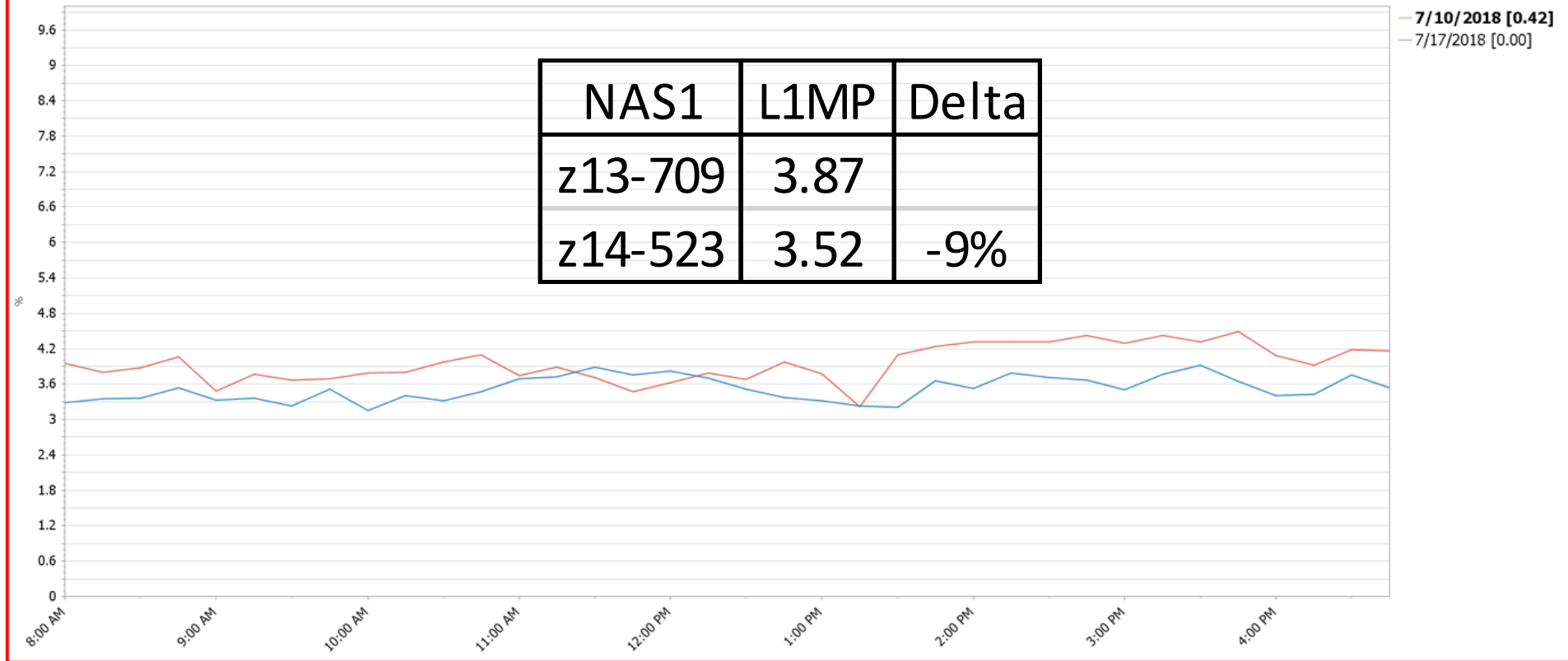
CECs	MSUs	# Phys CPs	# VHs	% on VHs
z13-709	1496	9	6	82.3
z14-523	1522	23	20	95.8

Total Cache Sizes

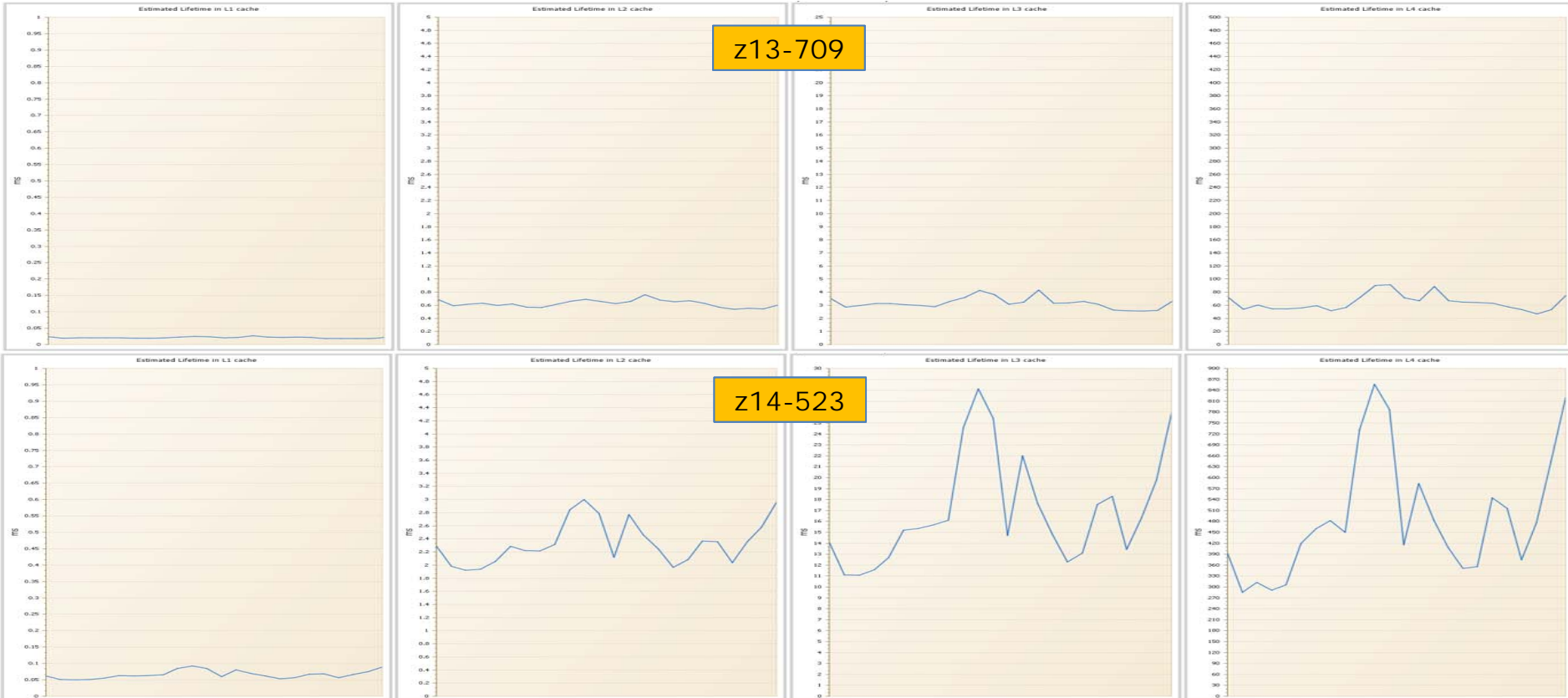
	Cache Sizes					
CEC	L1 KB	% Chg	L2 MB	% Chg	L3 MB	% Chg
z13-709	2016		36		128	
z14-708	2048	2%	48	33%	128	0%
z14-523	5888	192%	138	283%	384	200%

L1MP – Level 1 Miss Percentage

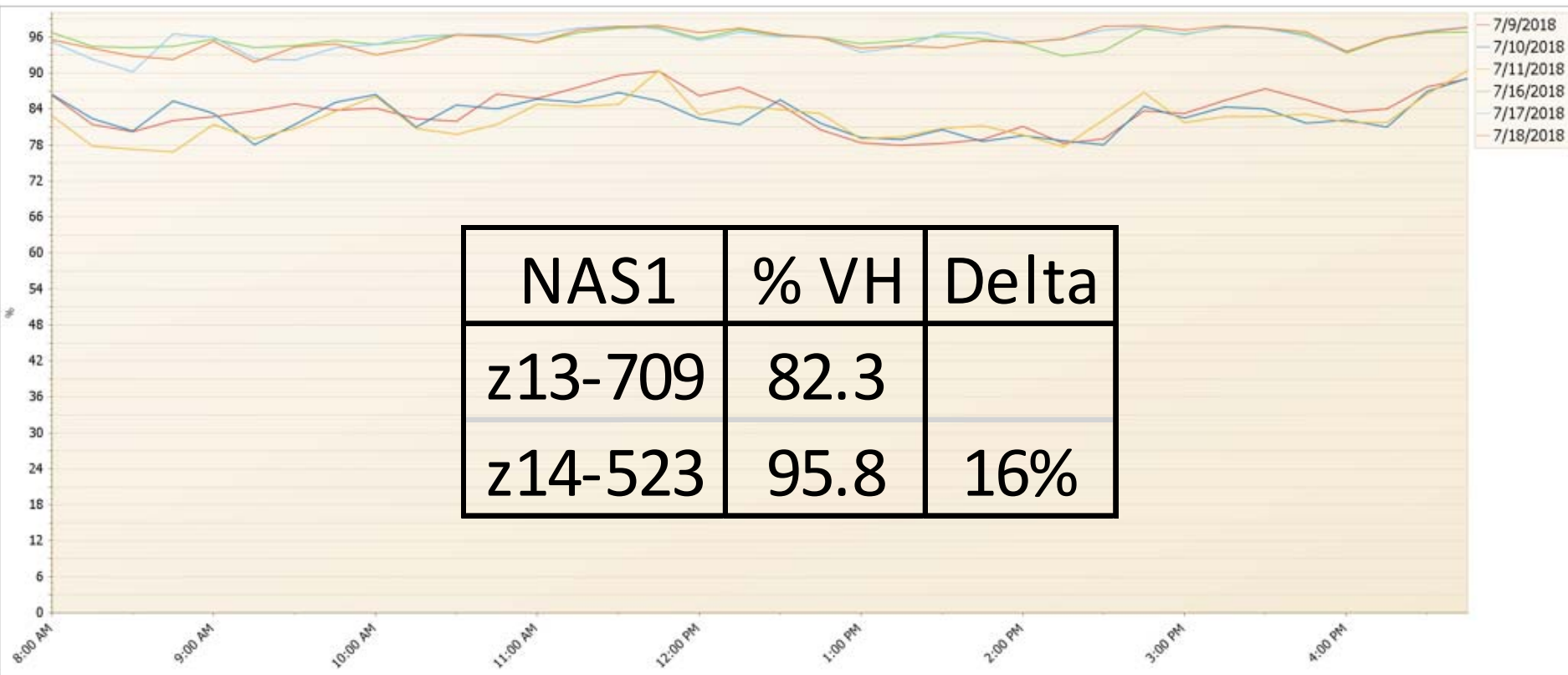
NAS1	L1MP	Delta
z13-709	3.87	
z14-523	3.52	-9%



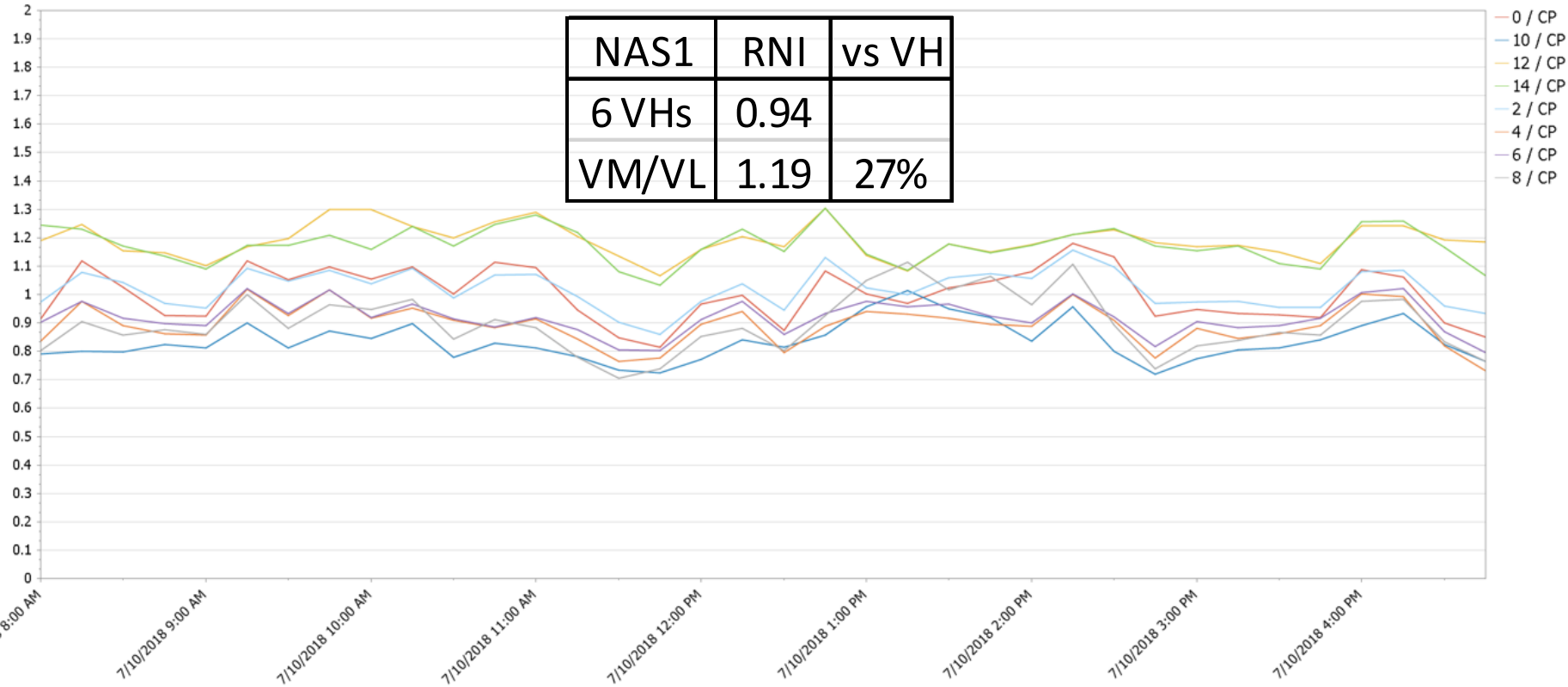
Cache Data Lifetime



% Workload Executing on VHs

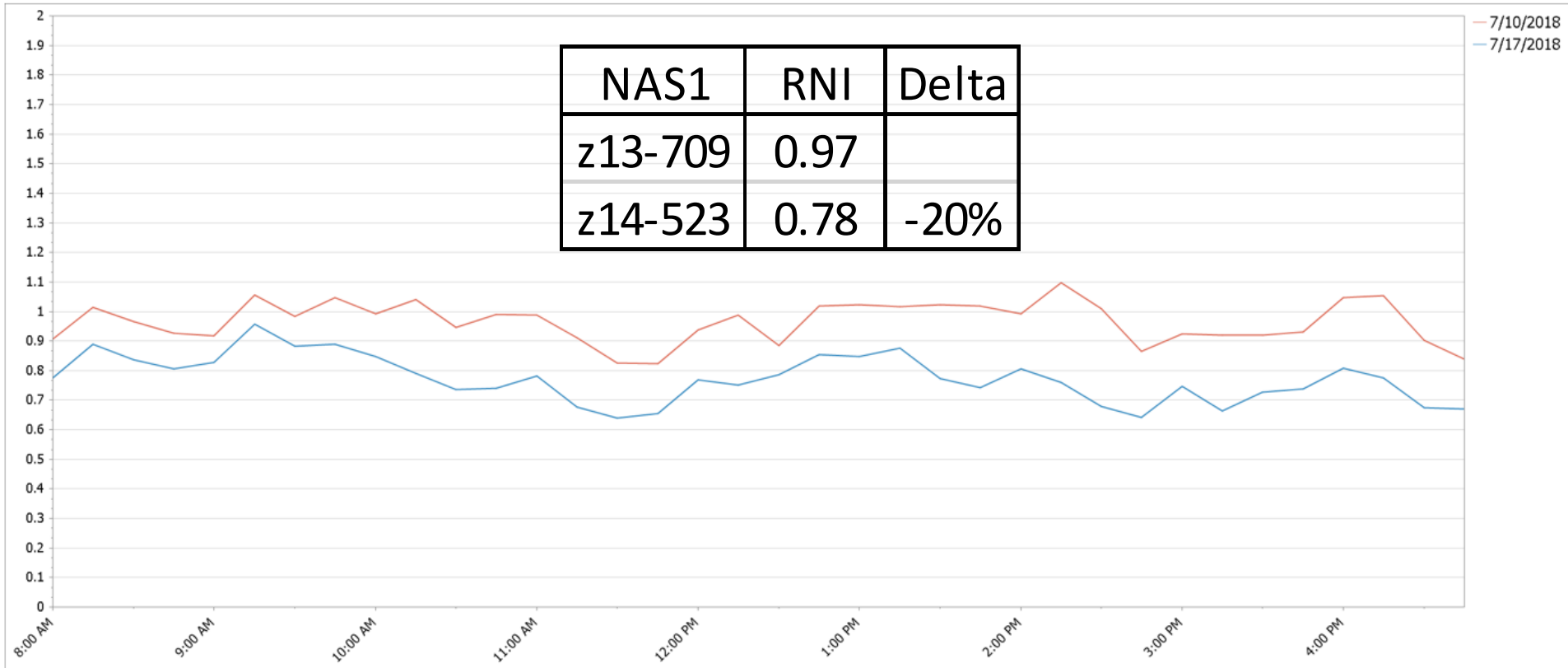


RNI by Logical CP – z13-709

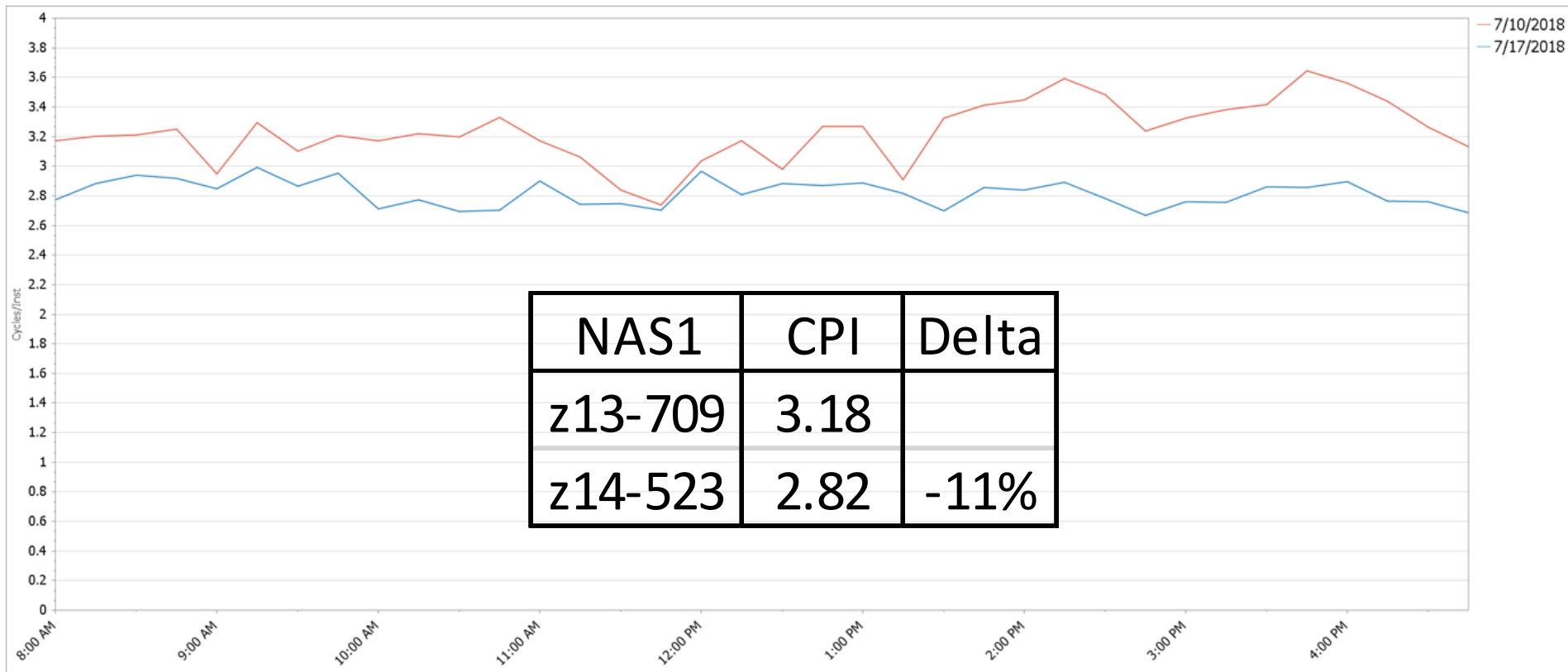


RNI by LPAR

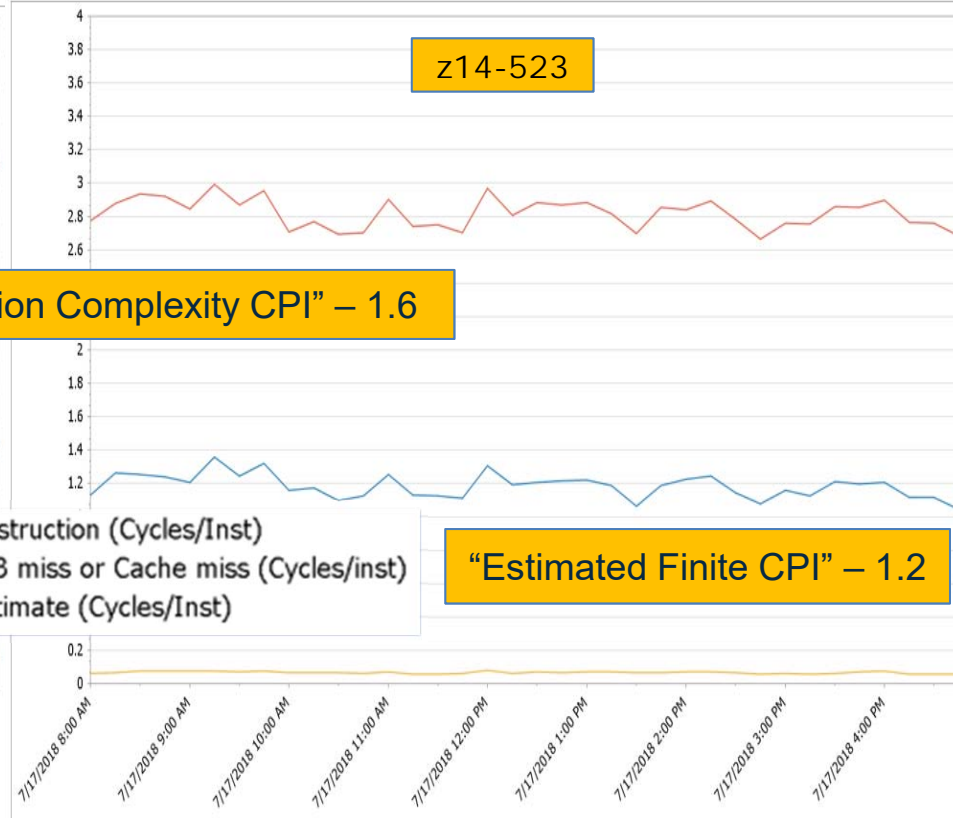
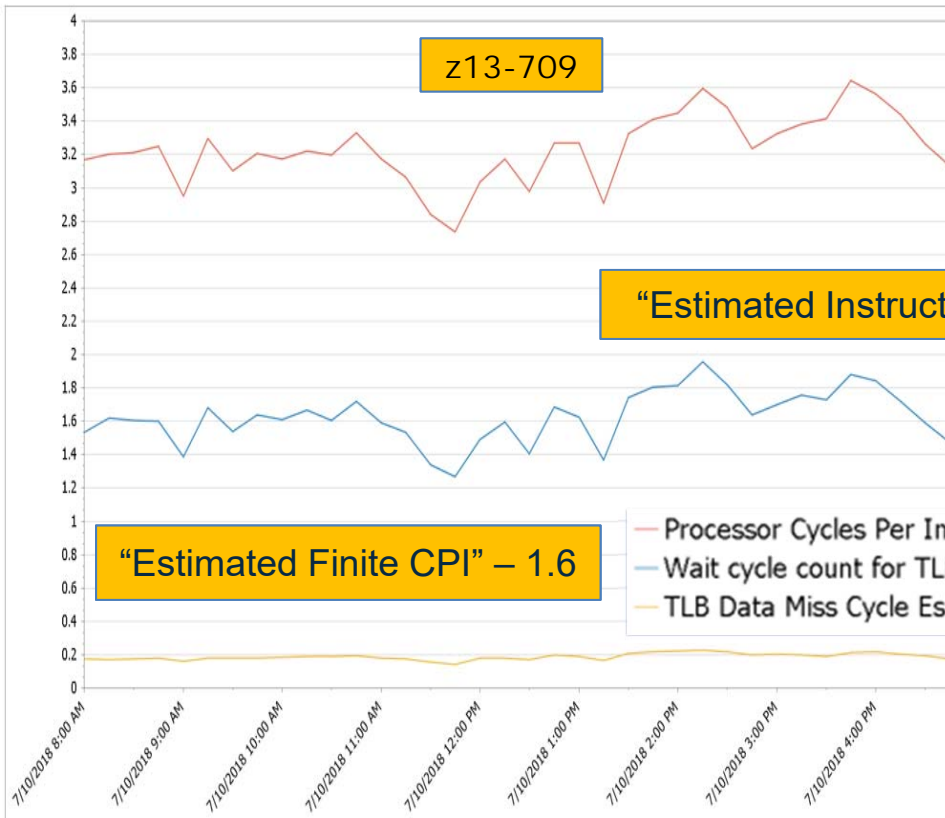
NAS1	RNI	Delta
z13-709	0.97	
z14-523	0.78	-20%



CPI – Cycles per Instruction



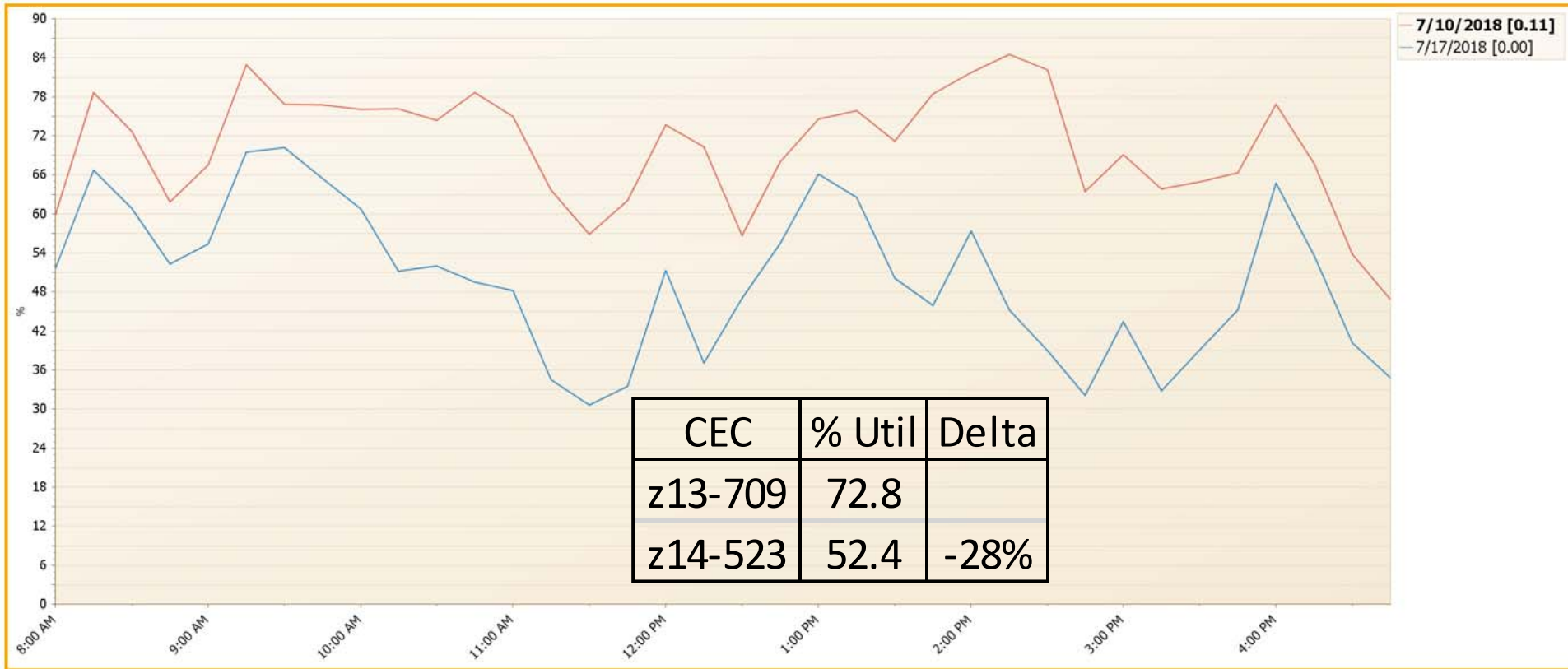
CPI Breakdown



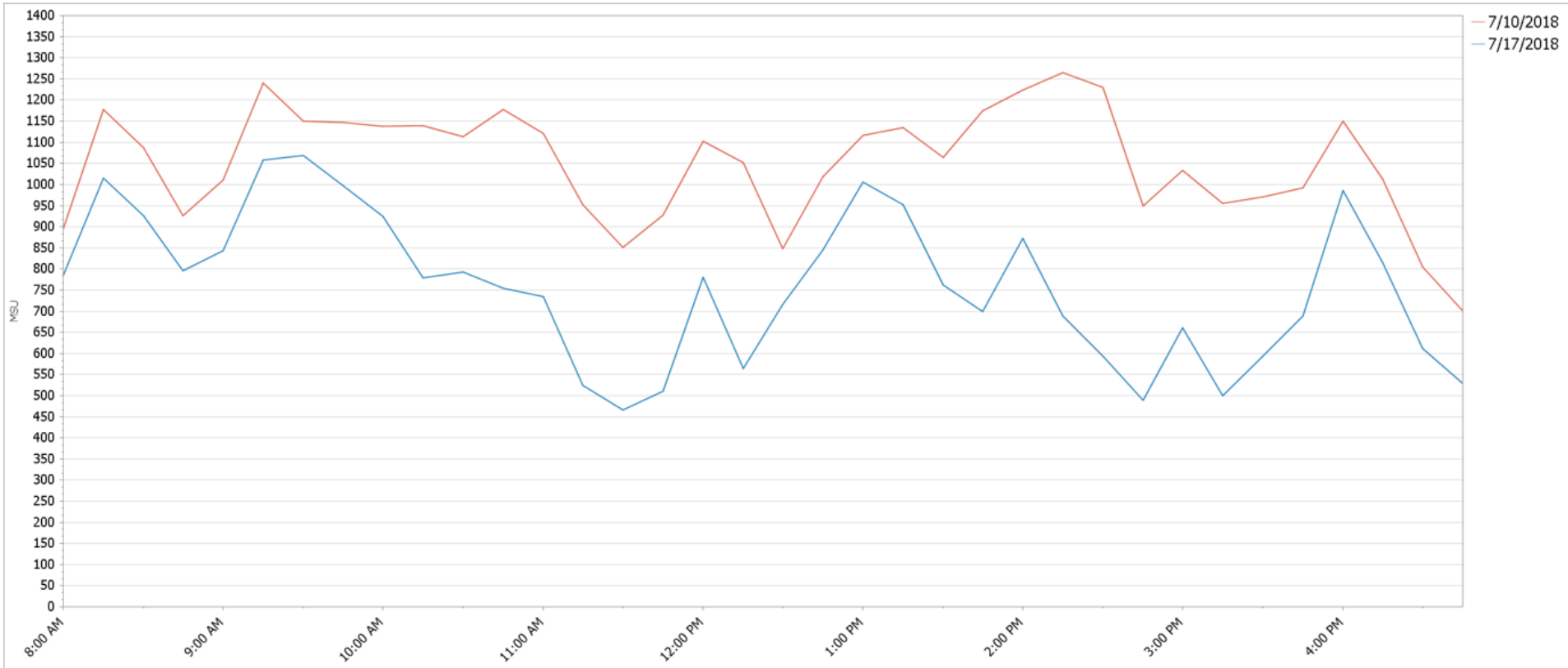
Impact on CPI Components

CPI	z13	z14
Inst Cmplx CPI	1.57	1.63
Finite CPI (Wait)	1.61	1.19
TLB Miss	0.19	0.07
Total CPI	3.18	2.82

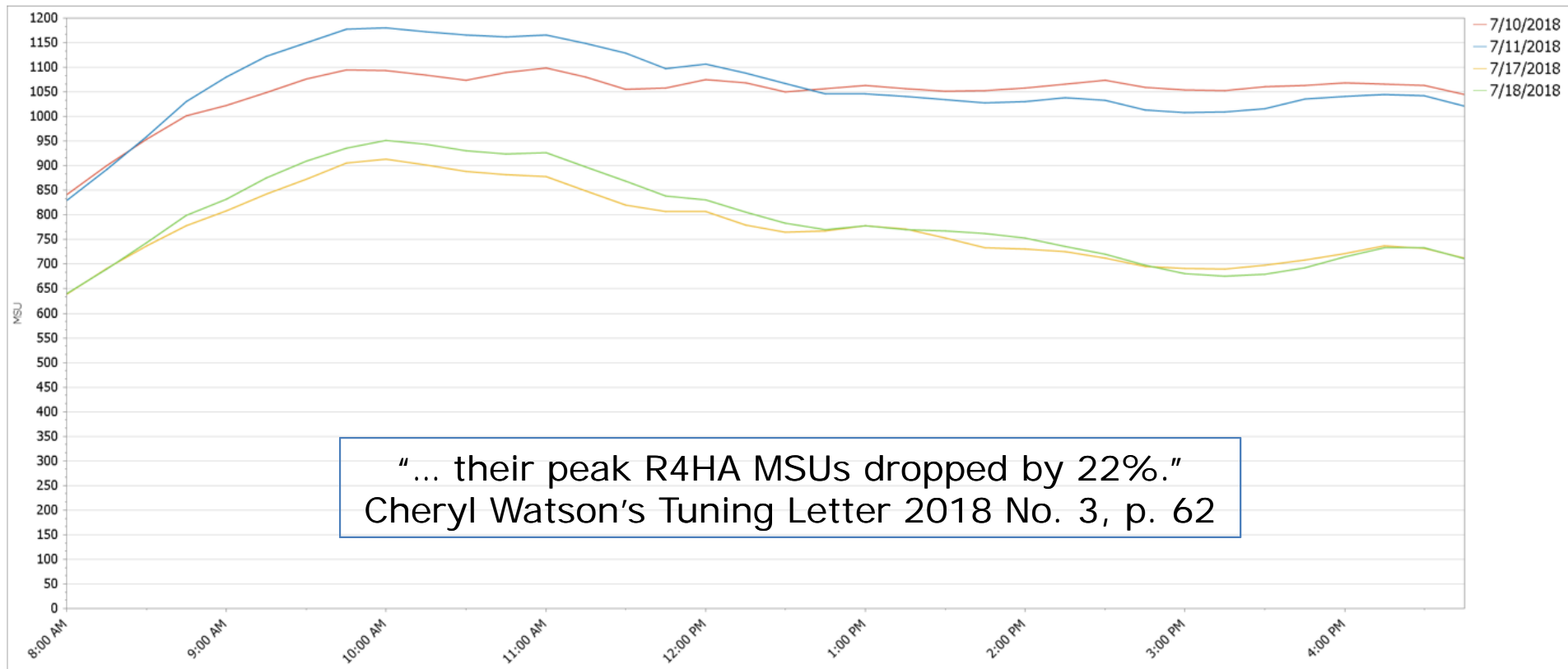
CEC Utilization



MSU Consumption



4 Hour Rolling Average



"... their peak R4HA MSUs dropped by 22%."
Cheryl Watson's Tuning Letter 2018 No. 3, p. 62



Useful IBM Tools

- IBM provides a number of tools to help you identify the ideal upgrade target CPC for you:
 - [zPCR](#)
 - zCP3000
 - [zBNA](#)
 - [TopoReport](#)
 - “View Partition Resource Assignments” on z14 SE
 - [LPAR Design Tool](#)
 - DIY CPU/Txn Tracking

- Available to everyone – [download](#) from Techdocs.
- Ideal input is EDF file created with CP3KEXTR from SMF type 70 **and 113** records.
- *You* select the interval to base your analysis on.
- Uses capacity numbers from LSPR and workload profile (your RNI) from SMF 113 records.
- Does NOT attempt to model the savings from running at lower utilization.
- Does NOT attempt to model impact of cache topology changes.

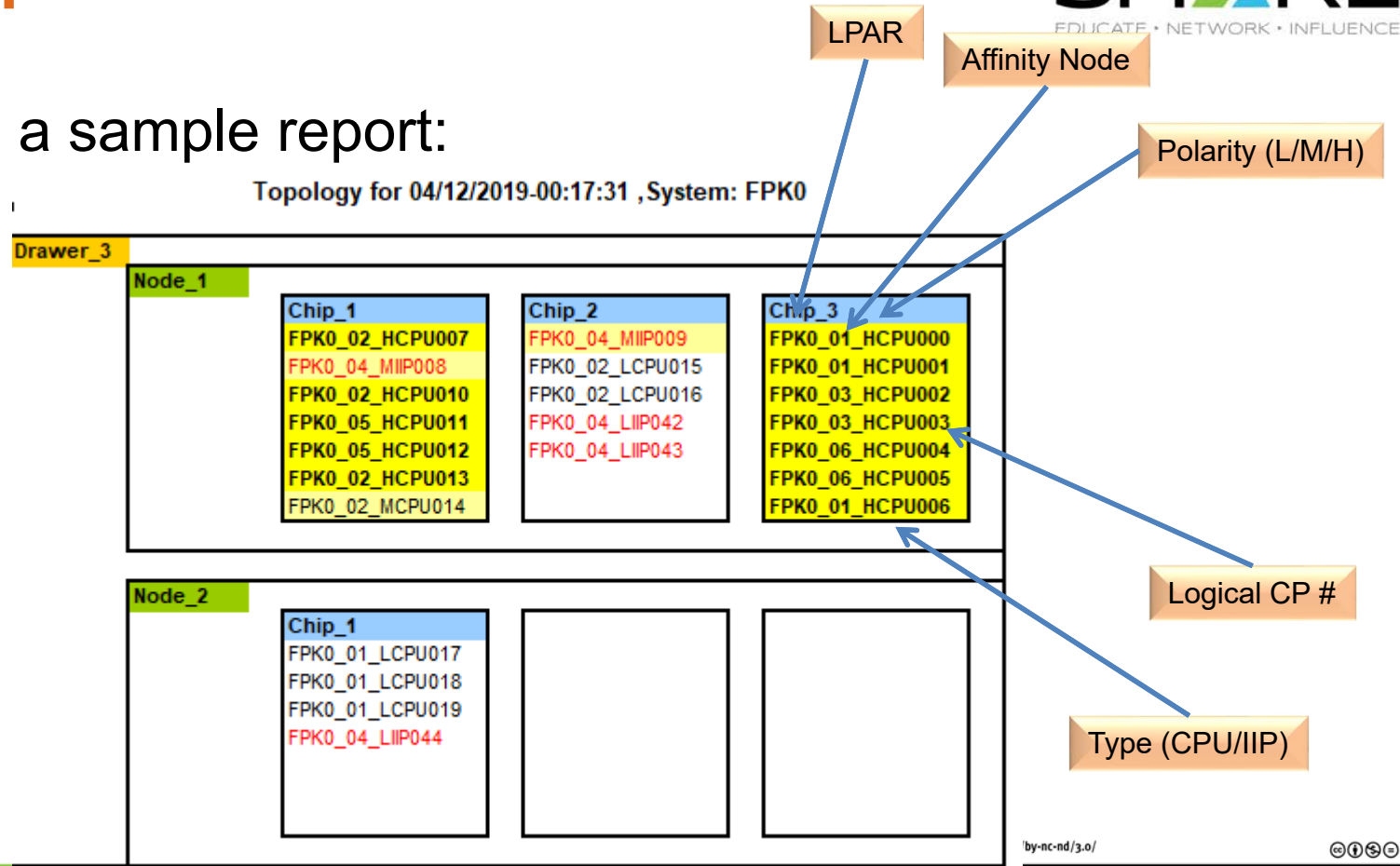
- Available to IBMers and Business Partners.
 - Does a lot more than zPCR, but it also provides a capacity planning/modeling capability.
 - *Requires* EDF files as input. Supports many more SMF record types than zPCR.
 - Also uses capacity numbers from LSPR and workload profile from SMF 113 records.
- Does NOT attempt to model the savings from running at lower utilization, except for one report (CEC1049).
 - Does NOT attempt to model impact of cache topology changes.

- Available to customers – download from [here](#).
 - See Session [25707](#) from Tuesday for info on new version of zBNA.
- Used for modeling various new IBM Z technologies (zHyperLinks, zEDC, Encryption, etc).
 - Also supports modeling the impact of changing the per-CP speed of your processor.
 - If you know which are your key critical path jobs, it will help you see how changing engine speed might affect that job.
 - Very helpful if you are considering a dramatic change in engine speed.
 - However, it is aimed at ‘normal’ programs. If you have programs that run multiple TCBs, zBNA can’t see the CPU consumption of each TCB.

- Available to customers – download from [here](#).
- The Topology report is a spreadsheet tool (created by the original creator of the RMF Spreadsheet Reporter), that reads SMF 99.14 records.
- It displays the relationship between logical CPs, WLM affinity nodes, and CPC chips.
- The information that it provides can be invaluable when contemplating CPC upgrades or LPAR configuration changes.

TopoReport

- Here is a sample report:



View Partition Resource Assignments

- This is a new function on z14 SE.
 - On earlier generations, you needed an LPAR dump to gather this info.
- On z14:
 - Logon to Support Element
 - Expand System Management group
 - Expand the CPC you are interested in
 - On bottom right, in Tasks area, expand Configuration
 - Click on View Partition Resource Assignments

View Partition Resource Assignments

IBM Support Element

Home View Partition Resource ... X

View Partition Resource Assignments - [REDACTED]

S [Pause] [Grid] [Print] Actions

[REDACTED] [HD] [REDACTED] [HD] [REDACTED] [HD] [REDACTED] [HD] [REDACTED] [HD]

- Node 00
 - Chip 0
 - 2 G SH
 - 2 G L
 - 1 Z L
 - 1 Z M
 - 1 G M
 - Chip 1**
 - 2 G SH
 - 3 G L
 - 1 G M
 - Chip 2
 - 8 G D
- Node 04
 - Chip 0
 - 1 I M
- Node 05
 - Chip 0
 - Chip 1
- Node 06
 - Chip 0

1 I M 1 I M 1 I M /30/

LPAR Design Tool

- One of the secrets to optimizing your use of the available processor cache is to have as many Vertical High CPs as possible.
- As Todd showed, the determination of how many High, Medium, and Low CPs an LPAR will have depends on its fair share of the total available capacity – and this is determined by the LPAR's relative weight.
 - A weight change as small as '1' can result in a VH logical CP being a VM one instead.

LPAR Design Tool

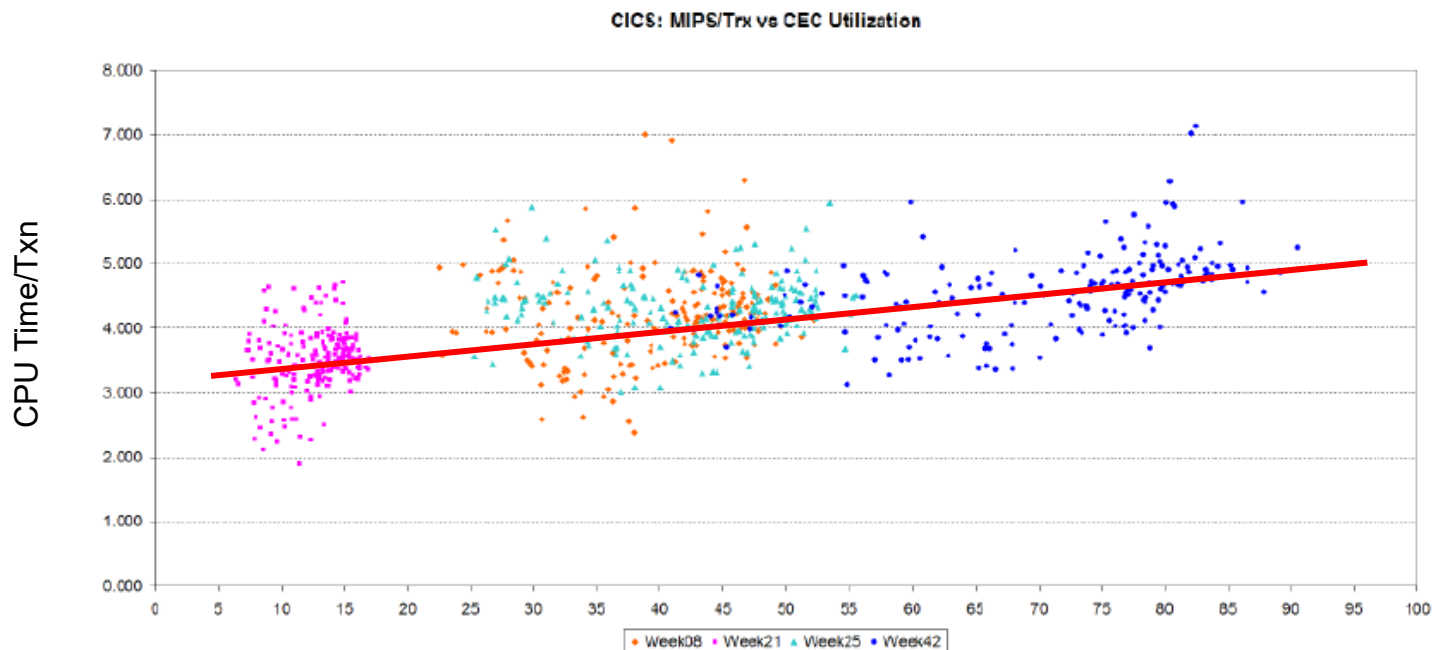
- The LPAR Design Tool is an excellent, free, tool to help you accurately model the impact of changing LPAR weights. It is written by Alain Maneville of IBM France.
- The tool can be downloaded from <https://github.com/AlainManeville/z-OS-LPARDesign>
 - For Tuning Letter subscribers, [*Tuning Letter 2017 No. 4*](#) includes [an article](#) that describes how to use the tool.
- All of our customers that try it say that they would never make an LPAR topology change again without modeling it with this tool first.

Tracking CPU/Transaction

- We get a lot of questions about the impact of high utilization on system overhead.
- Gary King's White Paper referenced earlier is a big help.
- However, the *best* way to get an accurate number for *your* system is to track the CPU per I/O for common, consistent, transactions at different times of the day, and plot those values against the physical CPC utilization at that time.
- Over time, you will build up a picture like this:

Tracking CPU/Transaction

- Using *your* data, you can calculate the impact of each additional x% of CPC utilization.



Getting the Goldilocks CPC

- Hopefully this information will help you get the most value from the IBM-provided tools, while at the same time adjusting for the effects that the tools do not allow for.
- It might seem like a lot of work, but CPC upgrades involve a lot of money, so investing some time in getting the right answer can pay back hundreds of times over.



z14 Processor Cache Design Changes

z14 Design Changes

- Improved PR/SM LPAR placement algorithms
 - Seeks to fit LPAR in single drawer & avoid remote accesses
 - Gives proximity to VH & VM GCPs (rather than VHs for GCPs & zIIPs)
- Unified L4 cache enables point-to-point access to remote drawers
- Strategic increases in cache sizes
- Level 1 TLB merged into Level 1 cache

System Design: z14 vs. z13 (Hutton, IBM)

z13

CPU

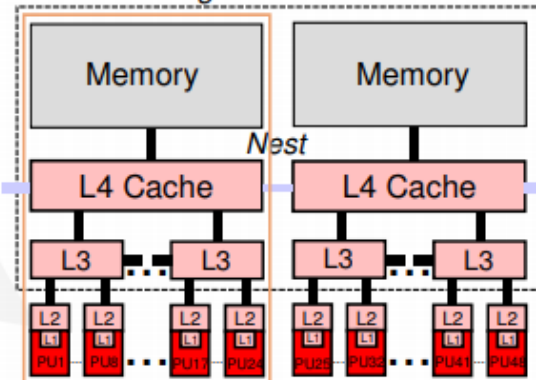
5.0 GHz
Major pipeline enhancements
1 picocoded translation engine

Caches

L1 private 96k i, 128k d
L2 private 2 MB i + 2 MB d
L3 shared 64 MB / chip
L4 shared 480 MB / **node**

Topology

8 cores + 1 L3 / CP chip
3 CP chips + 1 L4 / **node**
2 **nodes** / drawer
4 drawers / CEC
Book interconnect: NUMA



z14

CPU

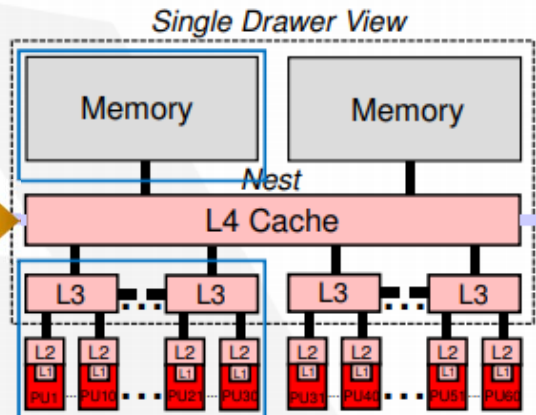
5.2 GHz
Logical directory w/ inclusive TLB
4 HW-implemented translation engines

Caches

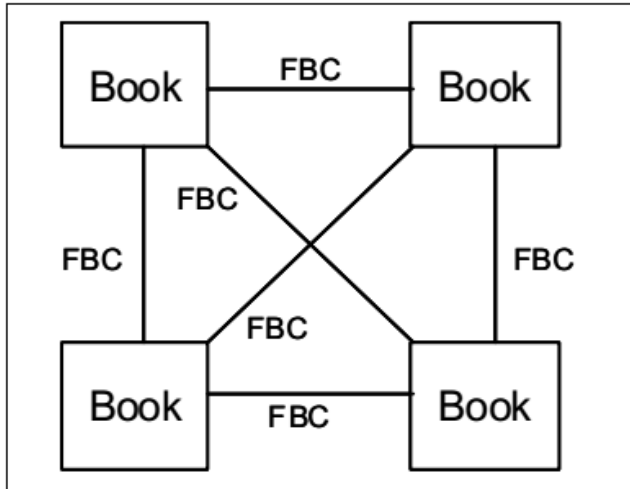
L1 private 128k i, 128k d
L2 private 2 MB i, 4 MB d
L3 shared 128 MB / chip
L4 shared 672 MB / **drawer**

Topology

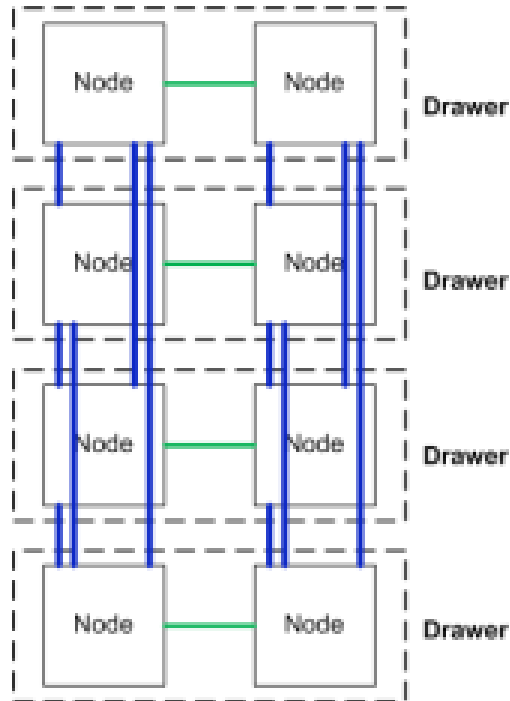
10 cores + 1 L3 / CP chip
3 CP chips / **cluster**
2 **clusters** + 1 L4 / drawer
4 drawers / CEC
Book interconnect: numa



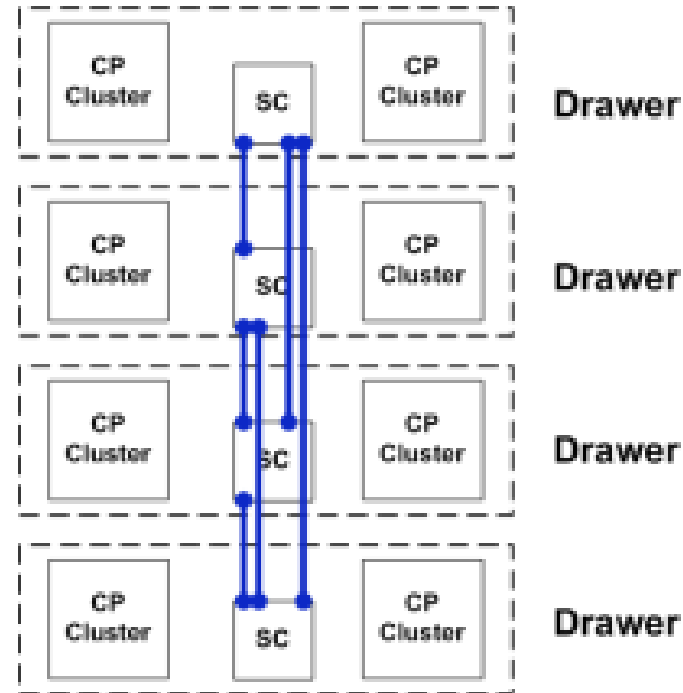
Drawer Interconnects



zEC12: Point-to-Point



z13: Multi-hop



z14: Point-to-Point

System Design: z14 vs. z13 (Hutton, IBM)

z13

CPU

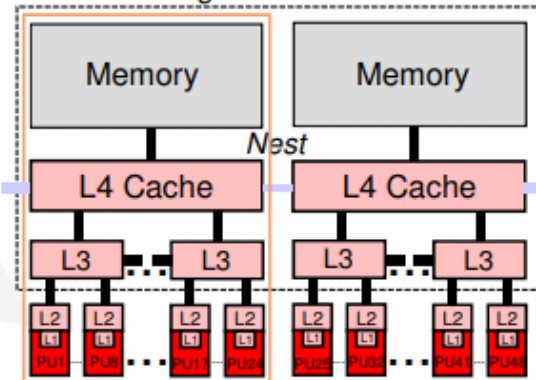
- 5.0 GHz
- Major pipeline enhancements
- 1 picocoded translation engine

Caches

- L1 private 96k i, 128k d
- L2 private 2 MB i + 2 MB d
- L3 shared 64 MB / chip
- L4 shared 480 MB / node

Topology

- 8 cores + 1 L3 / CP chip
- 3 CP chips + 1 L4 / node
- 2 nodes / drawer
- 4 drawers / CEC
- Book interconnect: NUMA



z14

CPU

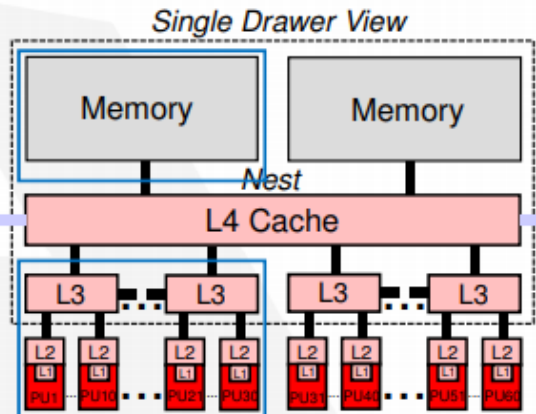
- 5.2 GHz
- Logical directory w/ inclusive TLB
- 4 HW-implemented translation engines

Caches

- L1 private 128k i, 128k d
- L2 private 2 MB i, 4 MB d
- L3 shared 128 MB / chip
- L4 shared 672 MB / drawer

Topology

- 10 cores + 1 L3 / CP chip
- 3 CP chips / cluster
- 2 clusters + 1 L4 / drawer
- 4 drawers / CEC
- Book interconnect: numa



z14 Cache Sizes

Cache		z13	z14	Mult
L1	Inst	96K	128K	1.33
L1	Data	128K	128K	none
L2	Inst	2M	2M	none
L2	Data	2M	4M	2
L3		64M	128M	2
L4		960M	672M	0.7

L1 & L2 caches per core; L3 per chip; L4 per drawer

Summary

- Key Processor Cache Concepts and Metrics
- Extent and Types of Savings Opportunities
- Detailed Customer Case Study
- Useful IBM Tools
- Highlights of z14 Processor Cache Design Changes

Sources

- Todd Havekost, “Achieving Significant Capacity Improvements on the IBM z13 Processor – User Experience”, SHARE 8/2016
- David Hutton, “The RNI-based LSPR and the Latest IBM Z Performance Brief”, SHARE 3/2018
- Gary King, “To MIPS or Not to MIPS”, SHARE 3/2017
- Frank Kyne, “A Holistic Approach to Capacity Planning”, Cheryl Watson’s Tuning Letter (CWTL) 2015 No. 4, pp. 55-75
- Kyne, “CPU MF Part 2 – Concepts”, CWTL 2017 No. 1, pp. 49-75
- Kyne, “Customer Sub-capacity CPC Experience”, CWTL 2018 No. 3, pp. 57-75



Backup Slides

Maximize Work on VHs – LPAR Weights

- Increase weights for high CPU LPARs
- Tailor weights to maximize assignment of VHs
- Adjust weights to reflect changes in workload (e.g., by shift)
- Configure fewer, larger LPARs
- Avoid activating “idle” LPARs with Production weights

Maximize Work on VHs – # of Physical CPs

- Utilize sub-capacity processor models
- Activate On/Off Capacity on Demand (CoD) during monthly peak intervals
- Install or deploy additional hardware