

Exploring the SMF 113 Processor Cache Counters



Instructor: Peter Enrico

Email: Peter.Enrico@EPStrategies.com

Enterprise Performance Strategies, Inc.
3457-53rd Avenue North, #145
Bradenton, FL 34210
<http://www.epstrategies.com>
<http://www.pivotor.com>

Voice: 813-435-2297
Mobile: 941-685-6789

Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 1

Reports / SMF 113 Processing/Discussion Offer !!!

☐ Special Reports Offer!

- See your SMF 113 records in chart and table format
- Please contact me, Peter Enrico for instructions for sending raw SMF data
 - ☐ Send an email to peter.enrico@epstrategies.com
- Deliverable:
 - ☐ Dozens of SMF 113 based reports (charts and tables)
 - Summary by system
 - Summary by CPU
 - Before / After comparison reports
 - Raw counter reports
 - Much more...
 - ☐ One-on-one phone call to explain your SMF 113 measurements

Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 2



Current 2011 Class Schedule

- [WLM Performance and Re-evaluating of Goals](#)
 - Instructor: Peter Enrico
 - June 6 - 10, 2011 Columbus, Ohio USA
 - September 12 - 16, 2011 Baltimore, Maryland, USA

- [Essential z/OS Performance Tuning](#)
 - Instructor: Peter Enrico and Tom Beretvas
 - May 9 - 13, 2011 St. Louis, Missouri, USA

- [Parallel Sysplex and z/OS Performance Tuning](#)
 - Instructor: Peter Enrico
 - May 16 - 20, 2011 Omaha, Nebraska USA
 - September 19 - 23, 2011 Dallas, Texas, USA

- [z/OS Capacity Planning and Performance Analysis](#)
 - Instructor: Ray Wicks
 - August 15 - 17, 2011 Columbus, Ohio, USA

Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 3

Abstract and Reports Offer

- [Abstract](#)
 - The new SMF 113 measurements record measurements are designed to provide insight into the movement of data and instruction among the processor cache and memory areas. These measurements will be invaluable to help quantify the net effect of everything from turning on HiperDispatch to making critical application change. In addition, the SMF 113 measurements have become the basis for IBM's LSPRs for processor sizing.
 - During this presentation Peter Enrico explain concept of processor caching on zArchitecture processors, the counters available in the SMF 113 record, formulas that make the counters come alive, examples of how the counters could be used.

- [Thank you to John Burg of the IBM Washington System Center for his insights and thoughts about the very interesting measurements in this SMF record.](#)

Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 4



Contact, Copyright, and Trademark Notices

Questions?

Send email to Peter at Peter.Enrico@EPStrategies.com, or visit our website at <http://www.epstrategies.com>.

Copyright Notice:

© Enterprise Performance Strategies, Inc. All rights reserved. No part of this material may be reproduced, distributed, stored in a retrieval system, transmitted, displayed, published or broadcast in any form or by any means, electronic, mechanical, photocopy, recording, or otherwise, without the prior written permission of Enterprise Performance Strategies, Inc. To obtain written permission please contact Enterprise Performance Strategies, Inc. Contact information can be obtained by visiting <http://www.epstrategies.com>.

Trademarks:

Enterprise Performance Strategies, Inc. presentation materials contain trademarks and registered trademarks of several companies.

The following are trademarks of Enterprise Performance Strategies, Inc.: **Health Check®**, **Reductions®**, **Pivotor®**

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries: IBM®, z/OS®, zSeries®, WebSphere®, CICS®, DB2®, S390®, WebSphere Application Server®, and many others.

Other trademarks and registered trademarks may exist in this presentation

Presentation Overview

- Why the SMF 113
- Overview of contents of SMF 113
- Primary basic formulas for SMF 113 usage
- New 'Nest' related formulas

Note: this presentation has not been totally updated for z196

- However all concepts are the same
- Just remember
 - z10 L1.5 cache is similar in concept to the L2 cache on the z196
 - z10 L2 cache is similar in concept to the L4 cache on the z196
 - Z196 has a new cache called L3 cache between L2 and L4



Introducing the SMF 113

Peter Enrico : www.epstrategies.com

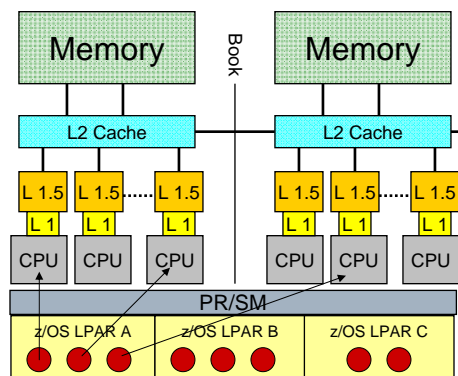
© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 7

Performance Analyst View of z10 Processor

It take more cycles to fetch information from further up in cache hierarchy

- L1 (Private level)
 - Data: 128KB
 - Instruction: 64KB
- L1.5 (Private level)
 - Unified for Data and Instruction
 - 3MB
- L2 (up to 4 shared caches)
 - Unified for Data and Instruction
 - 48MB
- Memory
 - Up to 384GB per book
 - (up to 1.5TB per machine)
 - Option to spread memory into multiple books



Peter Enrico : www.epstrategies.com

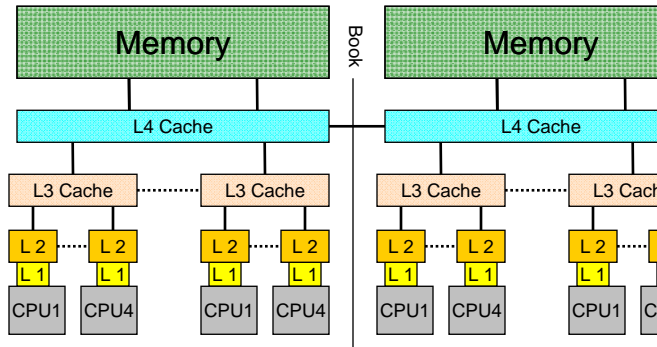
© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 8

Performance Analyst View of z196 Processor

It takes more cycles to fetch information from further up in cache hierarchy

- L1 (Private level)
 - Data: 128KB
 - Instruction: 64KB
- L2 (Private level)
 - 1.5MB
- L3
 - 24MB / chip
- L4
 - 192MB / book
- Memory



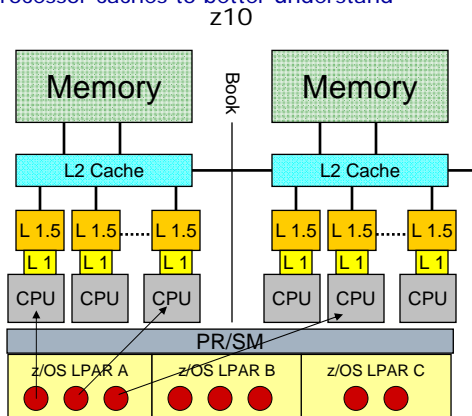
Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 9

Greatest Usage of SMF 113

- IBM's LSPR Workloads
 - See next slide (but fully explained later)
- Used to illustrate the usage of the processor caches to better understand before and after changes
 - Not good for benchmarking
 - But good to assuage concerns or gain insights
- Usage of standard SMF records still required for full processor evaluations
 - SMF 30
 - SMF 70
 - SMF 72.3
 - Etc..



Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 10

LSPRs and SMF 113s and RNI Hint

- SMF 113 measurements are now used to provide guidelines / hints for LSPR and zPCR processor sizing
- This RNI Hint table was documented in the Large System Performance Reference (LSPR)
 - Document Number SC28-1187-14
- The next slide shows an example of an LSPR chart used for processor sizing
- Using the SMF 113 records you now need to calculate
 - L1MP - L1 Miss Percentage
 - RNI - Relative Nest Intensity
- Note: This table and these guidelines are expected to change as more is learned from the SMF 113 records

L1MP	RNI	Workload Hint
<3%	>= 0.75	AVERAGE
	< 0.75	LOW
3% to 6%	>1.0	HIGH
	0.6 to 1.0	AVERAGE
	< 0.6	LOW
>6%	>=0.75	HIGH
	< 0.75	AVERAGE

Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 11

LSPR Table Example

IBM System z9 EC
(System z9 2094-701 = 1.00)

Processor	#CP	PCI**	MSU***	Low*	Average*	High*
2094-601	1	454	65	0.81	0.81	0.81
2094-602	2	880	127	1.6	1.57	1.53
2094-603	3	1303	184	2.38	2.33	2.23
2094-604	4	1720	240	3.13	3.07	2.92
2094-605	5	2109	292	3.87	3.77	3.58
2094-606	6	2482	339	4.59	4.43	4.21
2094-607	7	2842	385	5.3	5.08	4.81
2094-608	8	3188	428	5.99	5.69	5.37
2094-701	1	560	81	1	1	1
2094-702	2	1086	158	1.98	1.94	1.89
2094-703	3	1607	229	2.93	2.87	2.75
2094-704	4	2122	298	3.86	3.79	3.6
2094-705	5	2601	363	4.78	4.65	4.42
2094-706	6	3062	422	5.67	5.47	5.19
2094-707	7	3505	479	6.54	6.26	5.93
2094-708	8	3932	532	7.38	7.02	6.62

Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 12



New z10 (and higher) CPU Measurement Facility

- **Configure the z10 Server to collect CPU MF Data**
 - Update LPAR Security Tabs on HMC
- **Configure HIS facility on z/OS to collect CPU measurements**
 - Set up HIS Proc (See next slide)
 - Set up OMVS file system for *.CNT, *.MAP, and *.SMP files
 - Collect SMF 113s via SMFPRMxx
- **Collect CPU MF Data**
 - Start HIS proc
 - Use console modify command to begin/end counters and sampling
 - See next slide for syntax
 - Example: F HIS,B,TT='Text',PATH='/his/',CTRONLY,CTR=ALL
- **Analyze the CPU MF Data**
 - Sampling data
 - SMF 113 data
 - Note: Either is optional

Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 13

Setup Instruction Summary

- **Washington System Center Techdoc**
 - Collecting CPU MF (COUNTERS) on z/OS – Detailed Instructions***
- <http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TC000041>

Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 14

Output of CPU Measurement Facility

- For Counters enabled
 - CNT file
 - Named SYSHISyyyymmdd.hhmmss.CNT
 - HIS Start : collects begin measurements, and HIS End : collect end measurements
 - This text file is a summary of the delta between begin and end
 - SMF 113 records
 - Cut every 15 minutes (but not synced to SMF interval, and interval length not adjustable)
- If Sampling enabled (optional and explained towards end of this presentation)
 - Map file
 - Named SYSHISyyyymmdd.hhmmss.MAP
 - Text file contains load module mapping information
 - Sample data files
 - Named SYSHISyyyymmdd.hhmmss.SMP.cpu#
 - Large / voluminous files written for each z/OS logical processor on which data collection has been run
 - Contains sample data of the addresses on the instructions found executing during the sample, as well as some state information about the logical processor

Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 15

Counter Sets for z10 Machines Stored in SMF 113

- Basic Counters
 - Supervisor state + Problem state counters
 - Used to understand the activity of the CPU and L1 cache
- Problem Counters
 - Problem state counters (subset of Basic Counters)
 - Used to understand the activity of the CPU and L1 cache
 - These will be our stability measurements
- Crypto Counters
 - PRNG, SHA, DEA, AES counters
 - Crypto processor function calls and blocks broken down by algorithm
- Extended Counters
 - Used to understand the 'sourcing' of L1 from L1.5, L2 (local and remote), and memory (local and remote)

Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 16

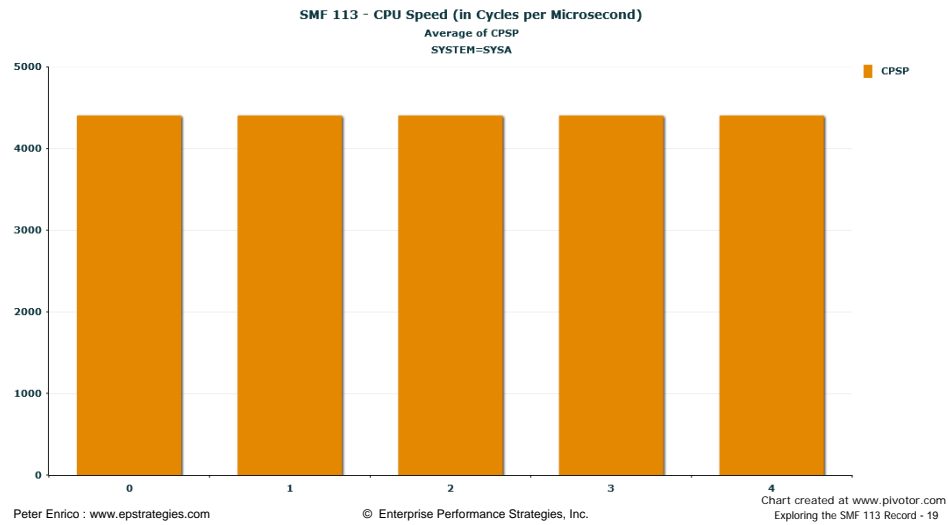
Highlights of SMF 113 Record

Processor Speed Information

- **SMF113_2_CPSP**
 - CPU speed in cycles per microsecond
 - Recorded for each logical CPU (but is really the physical CPU speed)
 - Example:
 - z10 : 4404 Cycles/Mic (i.e. 4.4 GHz)
 - Z196: 5208 Cycles/Mic (i.e. 5.2 GHz)
- **For knee capped processors**
 - Will reflect the reduced speed
 - But zIIPs and zAAP on the machine will show full speed numbers

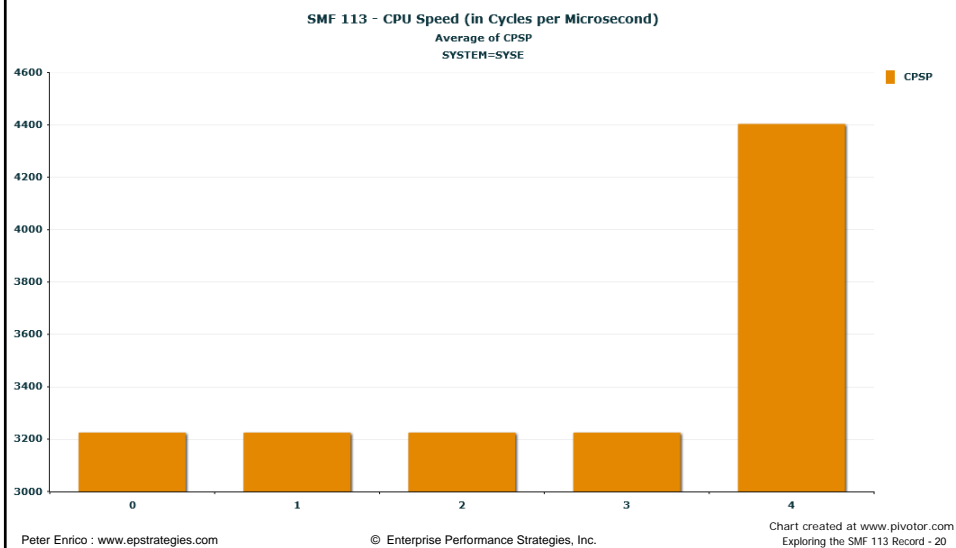
CPU Speed

- 5-CP way LPAR on 2097-706, E26, no zIIPs or zAAPs



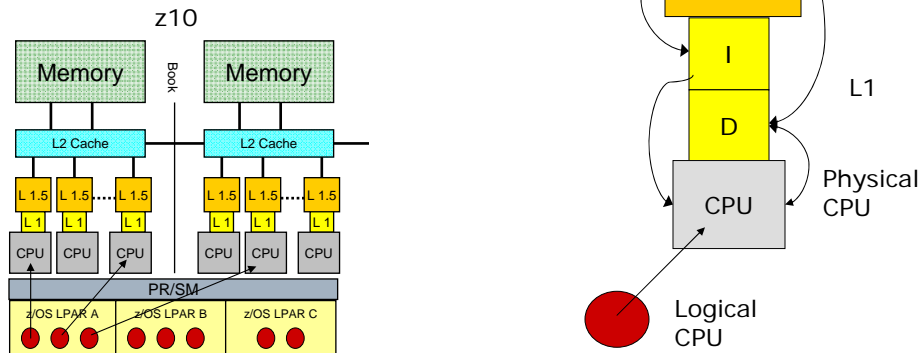
CPU Speed

- 4-CP way LPAR on 2097-604, E26, with 1 zIIP



COUNTER SET= BASIC / PROBLEM-STATE

- BASIC and Problem counters contain
 - L1 cache sourcing activity for both Data (D-cache) and Instruction (I-cache)
 - Contain instruction and cycle counters
 - Note 'Penalty' = 'Sourcing' = data/instruction gotten from somewhere and placed into L1 cache



Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 21

COUNTER SET= BASIC

- Activity count for CPU when in both problem and supervisor state
 - Counters for general purpose processors, zIIPs, and zAAPs
- 0: CYCLE COUNT
 - Number of CPU cycles, excluding the number of cycles CPU is in wait state
- 1: INSTRUCTION COUNT
 - Number of supervisor and problem state instructions executed by the CPU
- 2: L1 I-CACHE DIRECTORY-WRITE COUNT
 - Number of writes to instruction cache (and includes data cache if unified cache)
- 3: L1 I-CACHE PENALTY CYCLE COUNT
 - Instruction cache penalty cycle count (and includes data cache if unified cache)
- 4: L1 D-CACHE DIRECTORY-WRITE COUNT
 - Number of writes to data cache (and zero if unified cache)
- 5: L1 D-CACHE PENALTY CYCLE COUNT
 - Data cache penalty cycle count (and zero if unified cache)

Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 22

COUNTER SET= BASIC *.CNT file excerpt example

- CNT file contains delta values from begin to end of HIS modify begin/end
 - Note a very easy report to use, so it is recommended to use SMF 113 since will get measurements on an interval basis rather than begin/end delta

```
COUNTER SET= BASIC
COUNTER IDENTIFIERS:
  0: CYCLE COUNT
  1: INSTRUCTION COUNT
  2: L1 I-CACHE DIRECTORY-WRITE COUNT
  3: L1 I-CACHE PENALTY CYCLE COUNT
  4: L1 D-CACHE DIRECTORY-WRITE COUNT
  5: L1 D-CACHE PENALTY CYCLE COUNT

START TIME: 2010/03/16 11:25:21  START TOD: C5AFCE3D7E54909C
END TIME:   2010/03/16 14:44:15  END TOD:   C5AFFAB2674B2F8C
COUNTER VALUES (HEXADECEIMAL) FOR CPU 00 (CPU SPEED = 4404 CYCLES/MIC):
  0-  3 000007316AE25823 000000BE06CB4472 00000003CA983E8B 000001DE4B1B3543
  4-  7 00000004C2AE05DC 000003E785375A3C -----
START TIME: 2010/03/16 11:25:21  START TOD: C5AFCE3D7E586F9C
END TIME:   2010/03/16 14:44:15  END TOD:   C5AFFAB2674C708C
COUNTER VALUES (HEXADECEIMAL) FOR CPU 01 (CPU SPEED = 4404 CYCLES/MIC):
  0-  3 0000072B802120E8 000000B9C8DD0373 00000003CCC89351 000001E6B45F71C7
  4-  7 000000004BA302723 000003E608E79159 -----
```

Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 23

COUNTER SET= PROBLEM-STATE

- Activity count for CPU when in both problem state
 - Counters for general purpose processors, zIIPs, and zAAPs
- 32: PROBLEM-STATE CYCLE COUNT
 - Number of CPU cycles, excluding the number of cycles CPU is in wait state
- 33: PROBLEM-STATE INSTRUCTION COUNT
 - Number of problem state instructions executed by the CPU
- 34: PROBLEM-STATE L1 I-CACHE DIRECTORY-WRITE COUNT
 - Number of writes to instruction cache (and includes data cache if unified cache)
- 35: PROBLEM-STATE L1 I-CACHE PENALTY CYCLE COUNT
 - Instruction cache penalty cycle count (and includes data cache if unified cache)
- 36: PROBLEM-STATE L1 D-CACHE DIRECTORY-WRITE COUNT
 - Number of writes to data cache (and zero if unified cache)
- 37: PROBLEM-STATE L1 D-CACHE PENALTY CYCLE COUNT
 - Data cache penalty cycle count (and zero if unified cache)

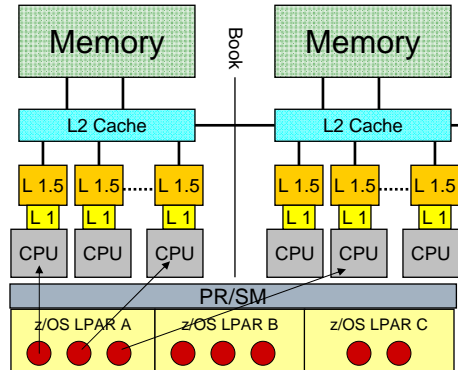
Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 24

z10 COUNTER SET= EXTENDED

- L1 from L1.5 cache movement
 - 128: Dir write to L1 I-cache dir from L1.5 cache (Instruction)
 - 129: Dir write to L1 D-cache dir from L1.5 cache (Data)



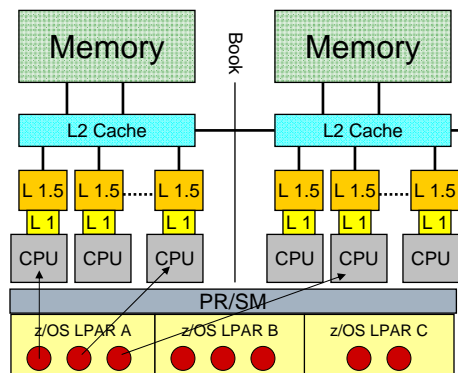
Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 25

z10 COUNTER SET= EXTENDED

- L1 from L2 Local cache movement
 - 130: Dir write to L1 I-cache dir from Local L2 cache (same book)
 - 131: Dir write to L1 D-cache dir from Local L2 cache (same book)
- L1 from L2 Remote cache movement
 - 132: Dir write to L1 I-cache from Remote L2 cache (not same book)
 - 133: Dir write to L1 D-cache from Remote L2 cache (not same book)



Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 26

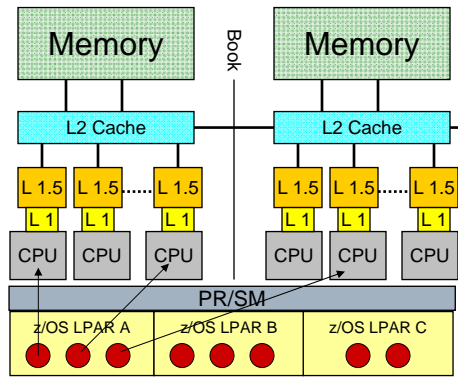
z10 COUNTER SET= EXTENDED

□ L1 from Local memory cache movement*

- 134: Dir write to L1 D-cache from memory same book (Local Memory)
- 135: Dir write to L1 I-cache from memory same book (Local Memory)

□ L1 from Remote memory cache movement

- Count does not exist, but see next slide for calculation



*Footnote: Notice, for some reason, reversal of 134 (D) and 135 (I) whereas all other counters are in (I) then (D) order.

z10 COUNTER SET= EXTENDED

□ Note: Previous slide was cache sourced from Local Memory

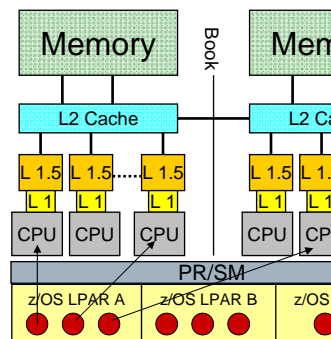
- But no count for L1 cached from Remote Memory
- Need to calculate

□ L1 I-cache source from Remote Memory =

- B2 : L1 I-Cache Dir-Write
 - (EC128 : write L1 I-cache from L1.5
 - +EC130 : write L1 I-cache from L2 local
 - +EC132 : write L1 I-cache from L2 remote
 - +EC135 : write L1 I-cache from Local Memory

□ L1 D-cache sourced from Remote Memory =

- B4 : L1 D-Cache Dir-Write
 - (EC129 : write L1 D-cache from L1.5
 - +EC131 : write L1 D-cache from L2 local
 - +EC133 : write L1 D-cache from L2 remote
 - +EC134 : write L1 D-cache from Local Memory



Using Key SMF 113 Metrics

Using Summarized Data in Formulas

- When using the SMF 113 records, insights could be gained by summarizing the counters and formulas based on the following:
 - By system
 - Example: For SYSA, for all CPs combined, all zIIPs, all zAAPs
 - By system, by CPU type (i.e. for all CPUs of a given type combined)
 - Example: For SYSA, for CP or zIIP or zAAPs, by CPU number
 - By machine (but remember that counters only collected for z/OS images)
 - Example: For CEC1 for all Systems

CPI – Cycles Per Instruction

- Key metric to gauge processor contention
 - Useful when doing a before / after comparison
 - Over time, useful to understand instruction mixture consistency
- When CPI increases
 - It is taking more cycles to execute the instruction mix
 - Shows an increase in contention
- When CPI decreases
 - It is taking less cycles to execute the instruction mix
 - Shows a decrease in contention
- Cycles / Instruction
 - Counters needed
 - B0: Cycle Count
 - B1: Instruction Count

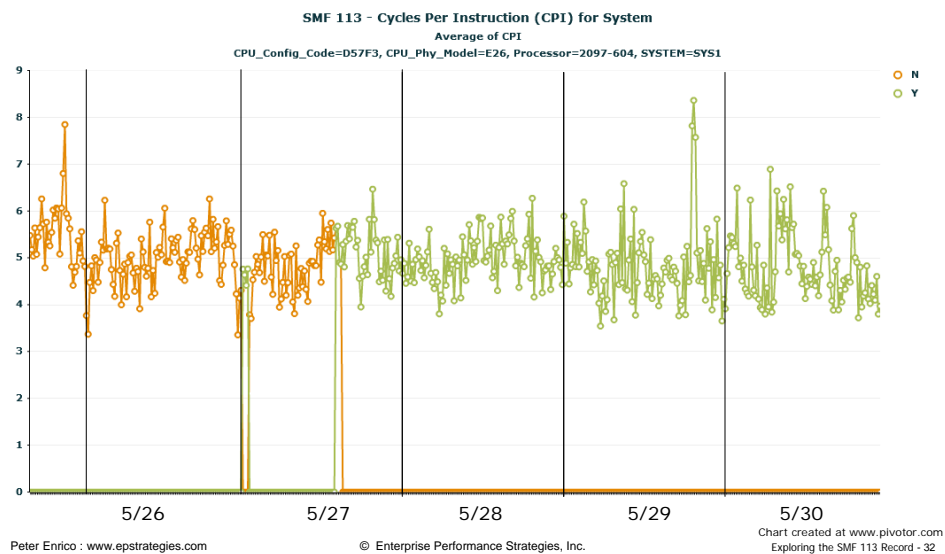
$$\begin{aligned}CPI &= (Total\ Cycles / Total\ Instructions) \\ &= (B0/B1)\end{aligned}$$

Peter Enrico : www.epstrategies.com

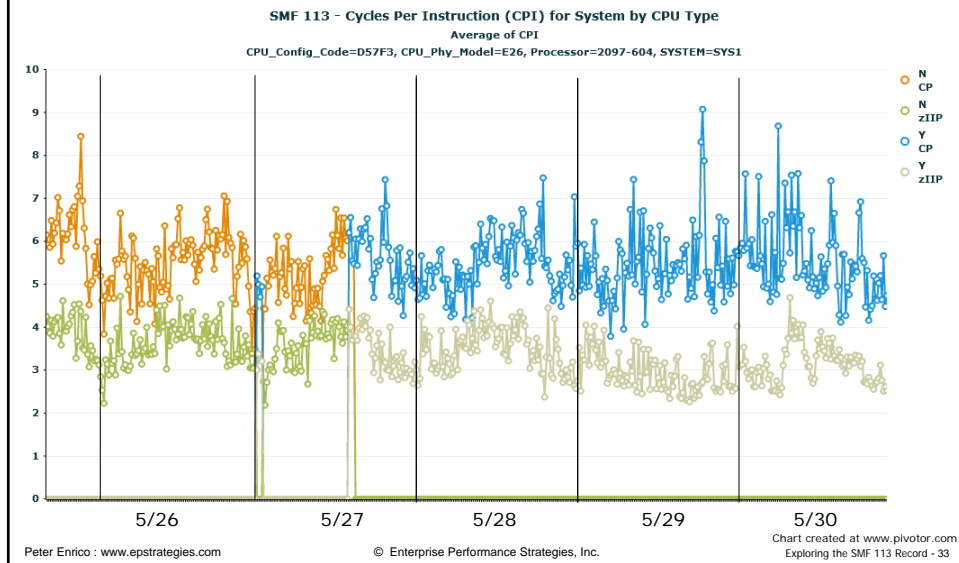
© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 31

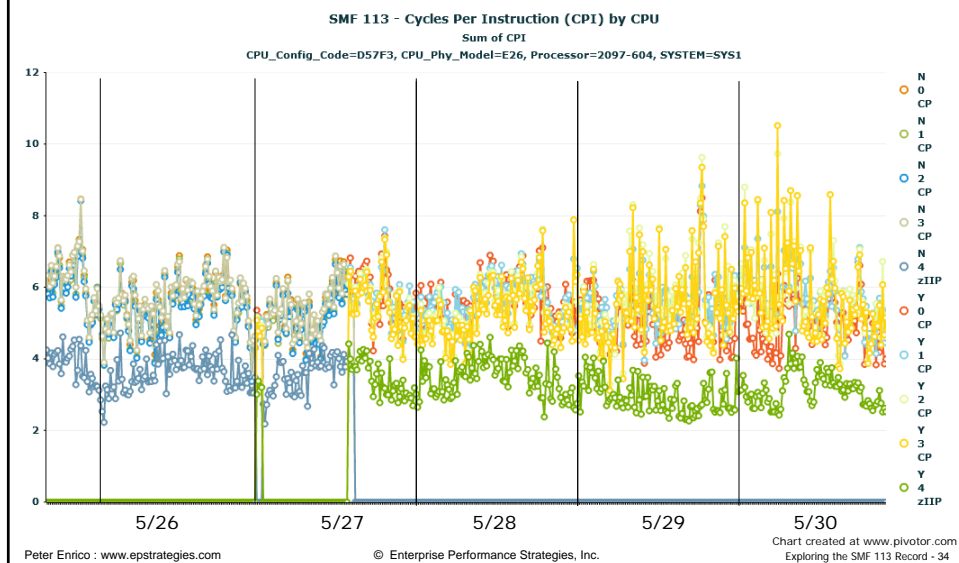
HD Before/After Example – Small MP



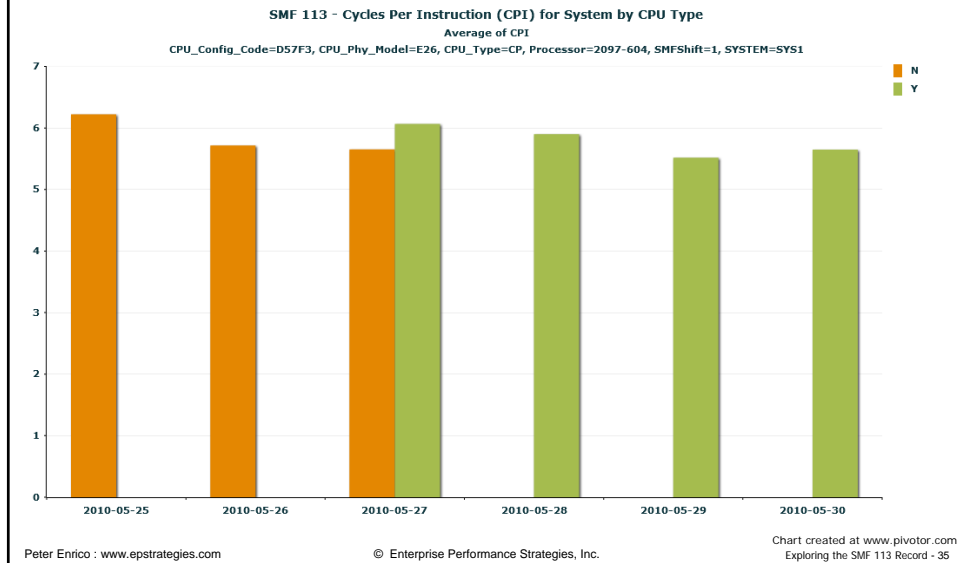
HD Before/After Example – Small MP



HD Before/After Example – Small MP



HD Before/After Example – Small MP



LPARCPU – LPAR Physical Busy %

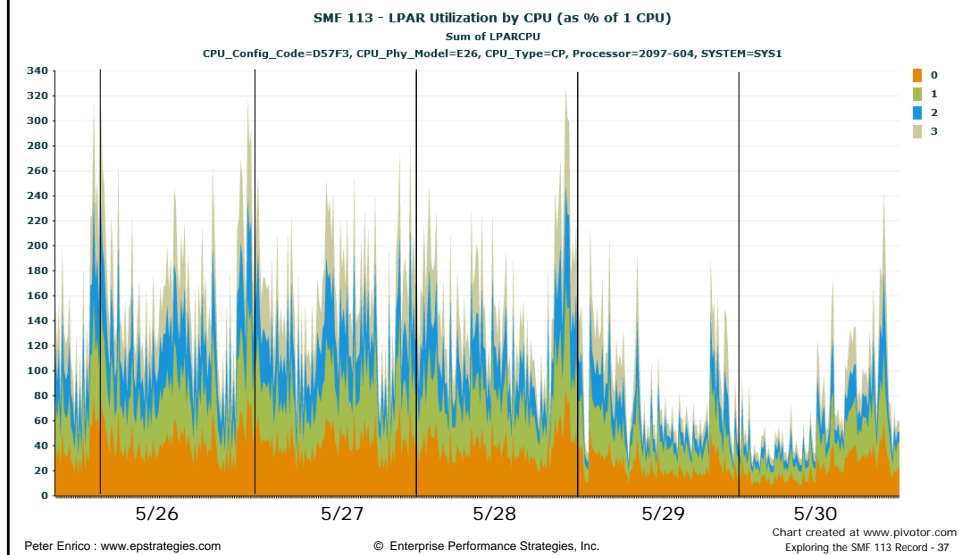
□ LPAR CPU

- LPARCPU (Cycle CPU %) (based on Cycle CPU seconds captured and un-captured)
- Counters needed
 - Processor Speed (cycles per microsecond) = SMF113_2_CPSP
 - B0: Cycle Count

$$LPARCPU = ((1/CPSP/1,000,000) * B0) / Interval\ Seconds * 100$$

- Example: Say for CPU0
 - If (Speed = 4404 Cycles/microsecond) &
 - (Executed 3,305,217,446,122 cycles executed in 900 seconds)
 - Then CPU utilization of CPU0 = 83.4%
- Add for each CPU to get utilization as a percent of 1 CPU
- Or Average for all CPUs to get a LPAR Counter Utilization %

HD Before/After Example – Small MP



L1 Sourcing CPU %

□ L1 Sourcing CPU%

- The portion of LPARCPU due to sourcing
- Counters needed
 - Processor Speed (cycles per microsecond) = SMF113_2_CPSP
 - B3: Instruction L1 Sourcing Penalty Cycle Count
 - B5: Data L1 Sourcing Penalty Cycle Count
- *Note: Not yet sure if this formula is valid due to the overlapping of sourcing cycles.*
 - *It is very likely the below result would need to be scaled (as discussed later in this presentation).*

$$LPARCPU = \left(\left(\frac{1}{CPSP/1,000,000} \right) * (B3+B5) \right) / \text{Interval Seconds} * 100$$

PRBSTATE (Problem Instruction to Total Instruction)

□ Problem to Total Instruction Ratio

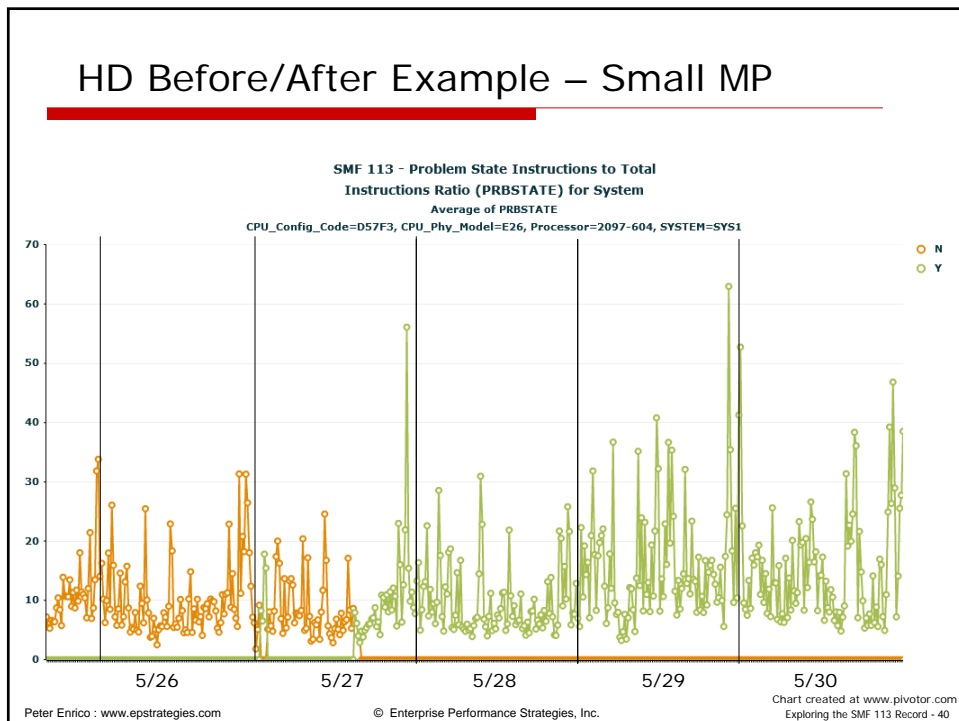
- Ratio of Problem State instructions to Total instructions
- Counters needed
 - B1 = (Supervisor State Instructions + Problem State Instructions)
 - P33 = (Problem State Instructions)

$$PRBSTATE = P33 / B1$$

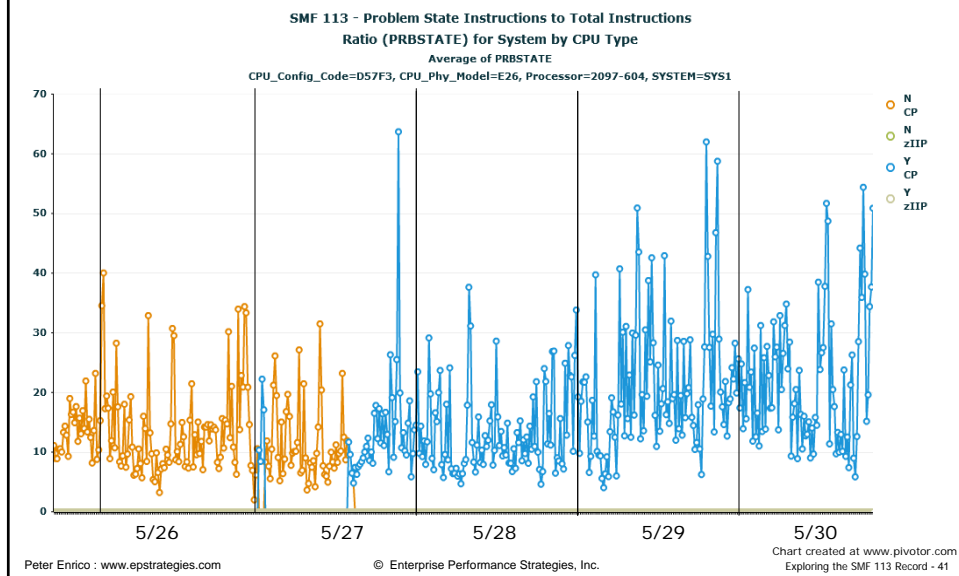
□ This is our stability factor

- Useful to help indicate if the before / after workload is consistent

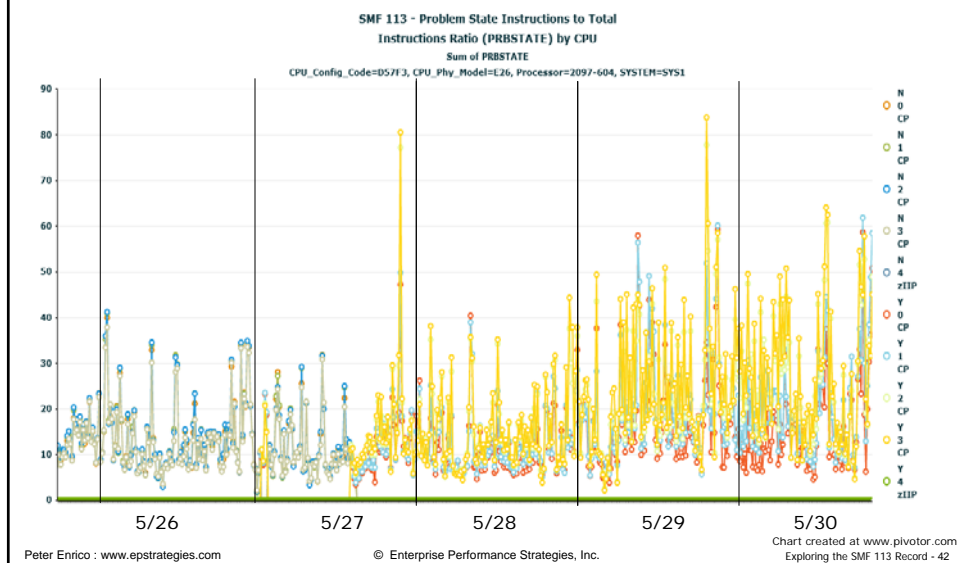
HD Before/After Example – Small MP



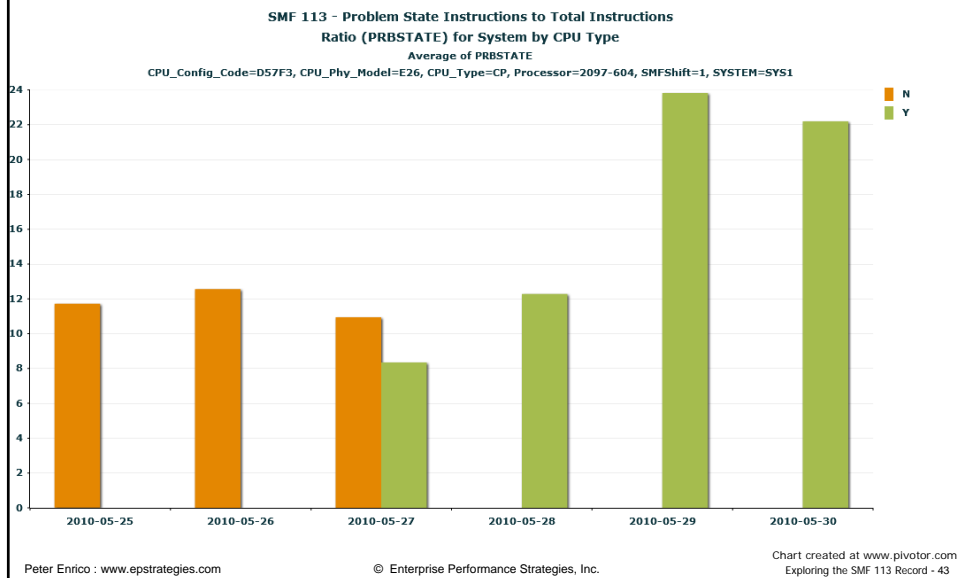
HD Before/After Example – Small MP



HD Before/After Example – Small MP



HD Before/After Example – Small MP



Useful Formulas

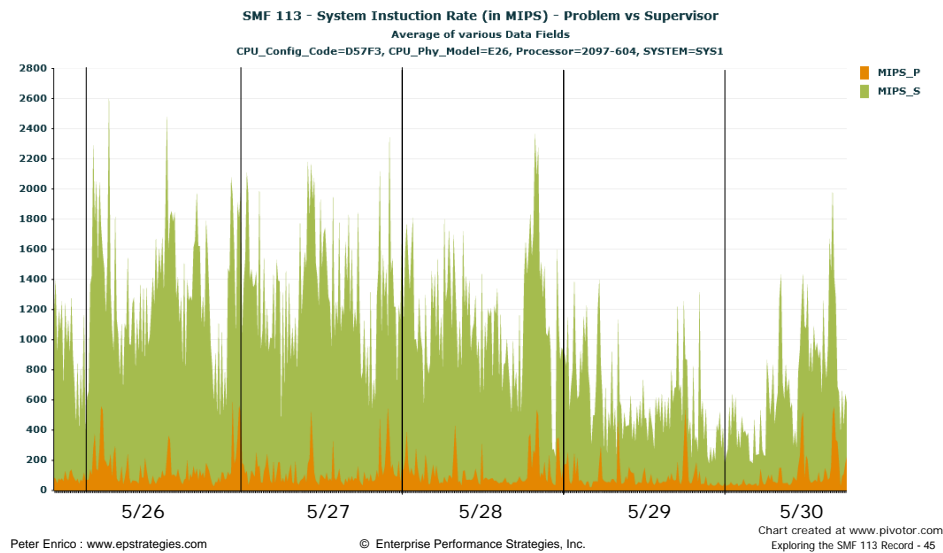
□ Executed Instructions Rate (in Million Instructions per Second)

- This is really the inverse of the CPI number (cycles per instruction)
- So recommend using CPI to compare changes rather than this MIPS number
- Counters needed
 - B1: Instruction Count
 - Measurement length in seconds

$$MIPS = (B1 / Interval\ Seconds) / 1,000,000$$

- This will not, and is not expected to, match any sort of MIPS table value or MIPS number you are utilizing today
- This MIPS number has absolutely nothing to do with capacity!

Instruction Rate - HiperDispatch Example



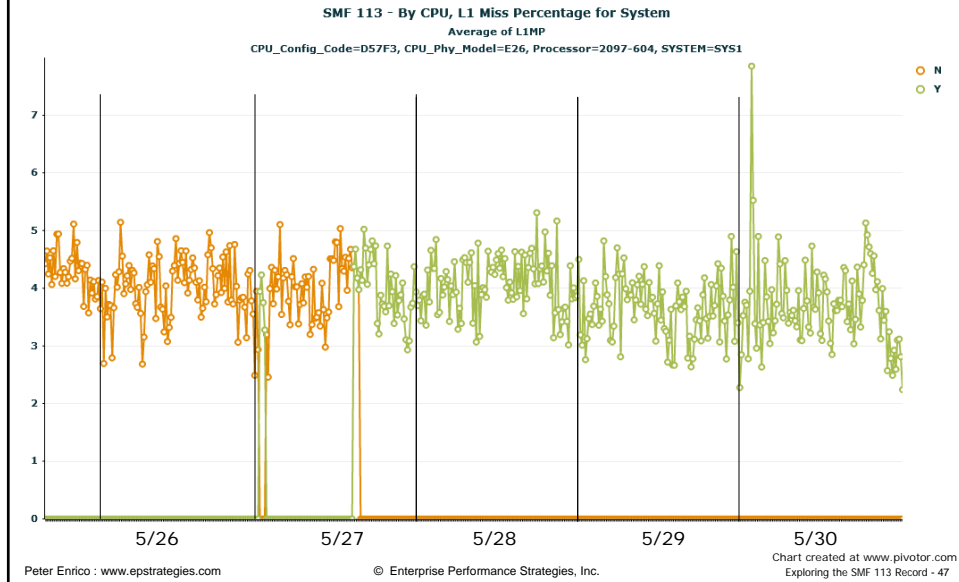
L1 Cache Miss %

□ L1 Cache Miss %

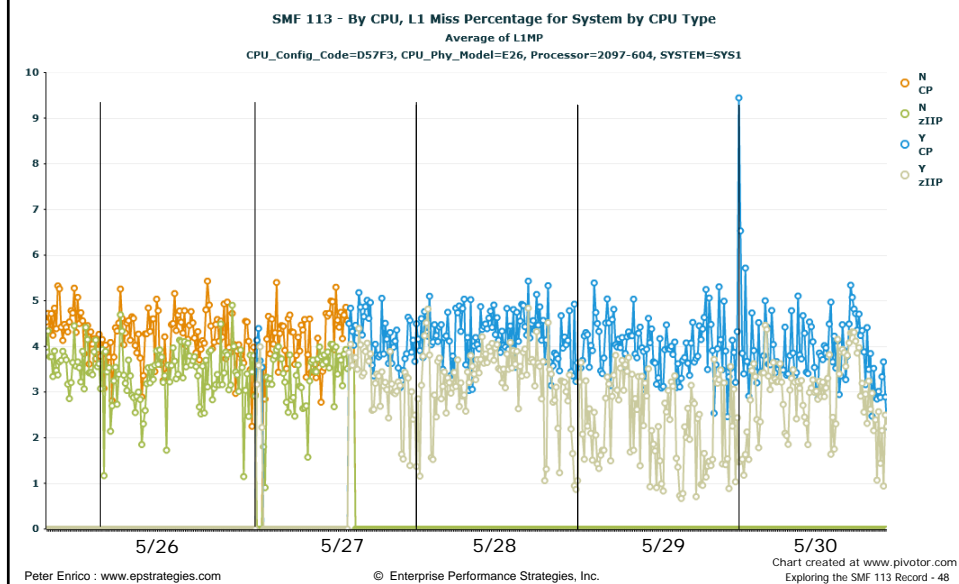
- Means percentage of counters when L1 I-cache or L1 D-cache got a cache miss
- Opposite of the hit percentage
 - Calculate miss rather than hit since the source % numbers (presented in subsequent slides) will be a breakdown of this miss %
- Based on counters
 - B2: L1 I-Cache Dir-Write Count
 - B4: L1 D-Cache Dir-Write Count
 - B1: INSTRUCTION COUNT

$$L1MP = ((B2 + B4) / B1) * 100$$

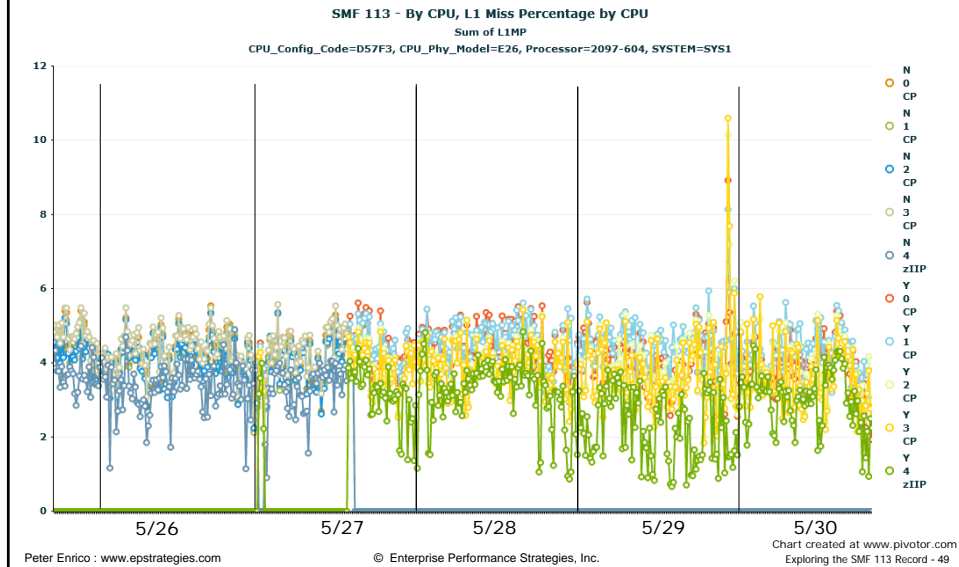
HD Before/After Example – Small MP



HD Before/After Example – Small MP

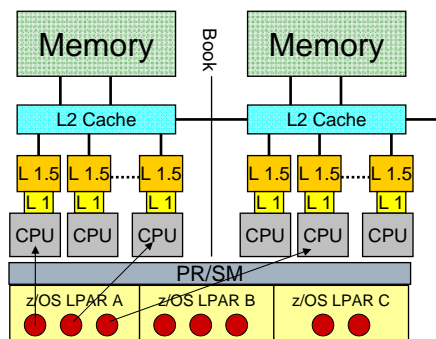


HD Before/After Example – Small MP



z10 Components of L1 Sourced

- If an L1 Miss Occurs then the Instructions and Data needs to be Sourced from some other cache / memory location
- Question to be answered
 - From where did the L1 get sourced?
 - Or to put it another way, what is the breakdown of how L1 Misses were resolved



Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 50

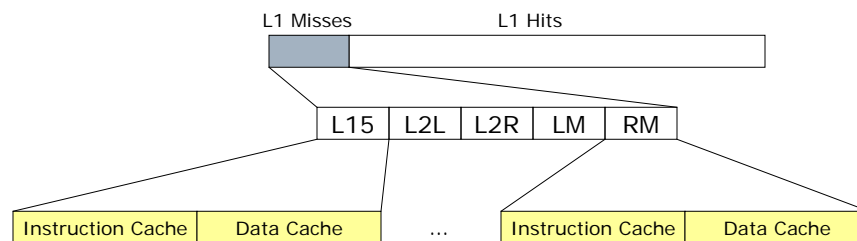
z10 From Where is L1 Sourced?

□ Answer

- From L1.5 (Instruction and Data)
- From L2 Local (Instruction and Data)
- From L2 Remote (Instruction and Data)
- From Local Memory (Instruction and Data)
- From Remote Memory (Instruction and Data)

□ Can calculate by area

□ Can calculate by Instruction or Data



Peter Enrico : www.epstrategies.com

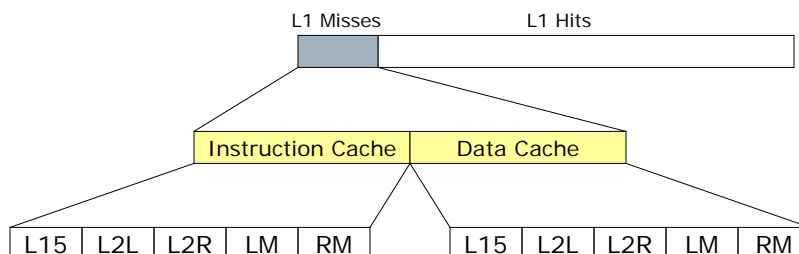
© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 51

z10 From Where is L1 Sourced?

□ Another interesting ways to look at the data

- Breakdown misses further to understand impact of instruction and data caches



□ Unfortunately Extended counts not granular to allow a better understand of L1 source affects to problem state or supervisor state

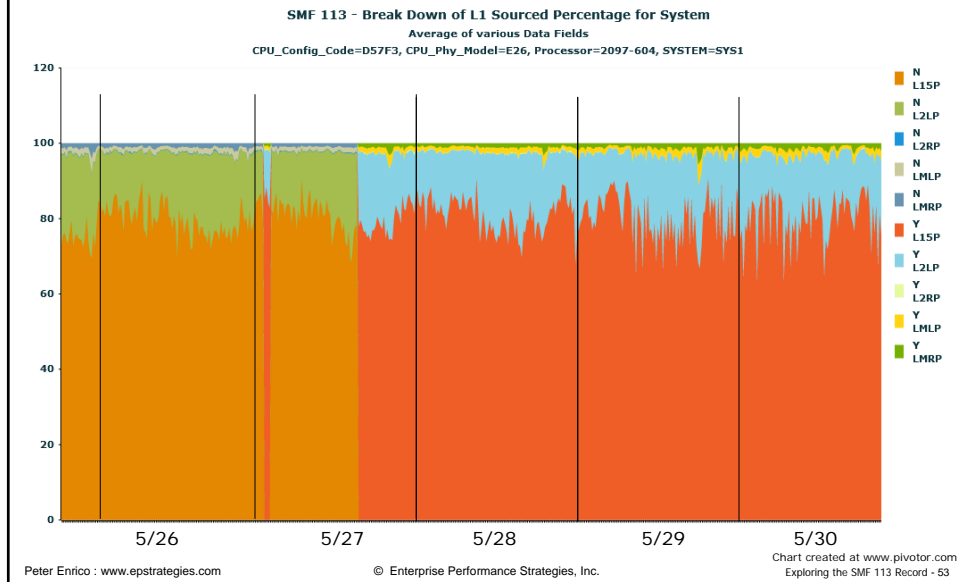
- So can only get based on Basic Counters (see next slides)

Peter Enrico : www.epstrategies.com

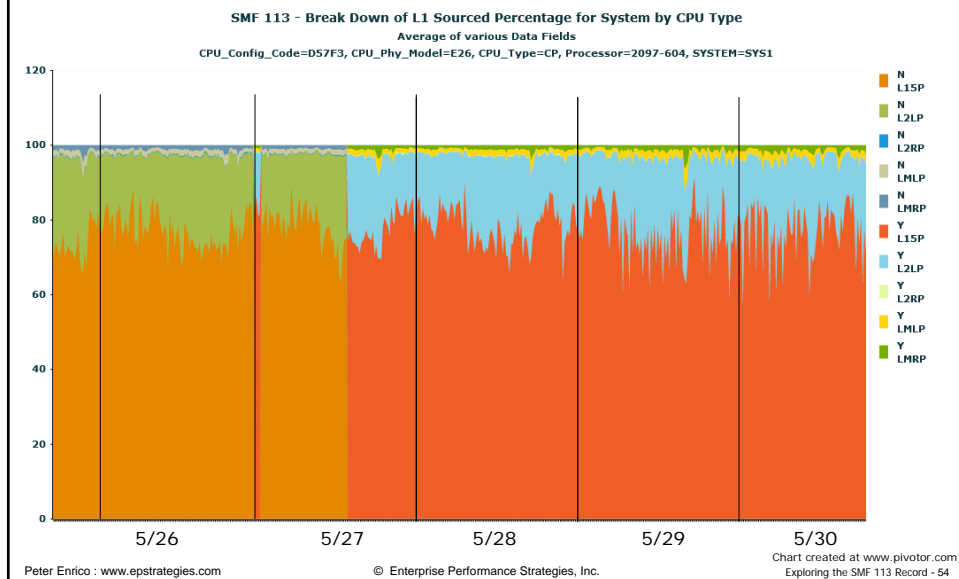
© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 52

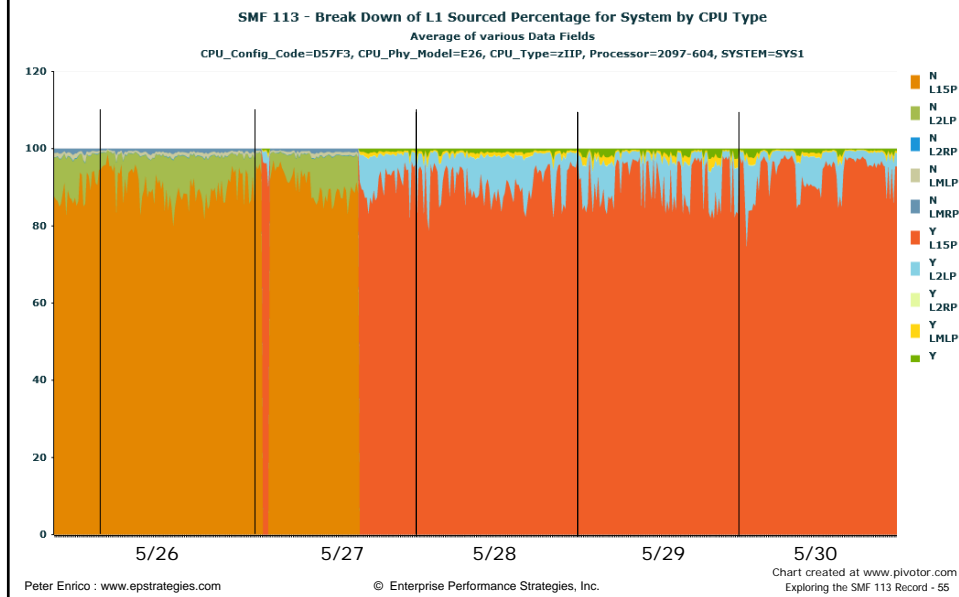
z10 HD Before/After Example – Small MP



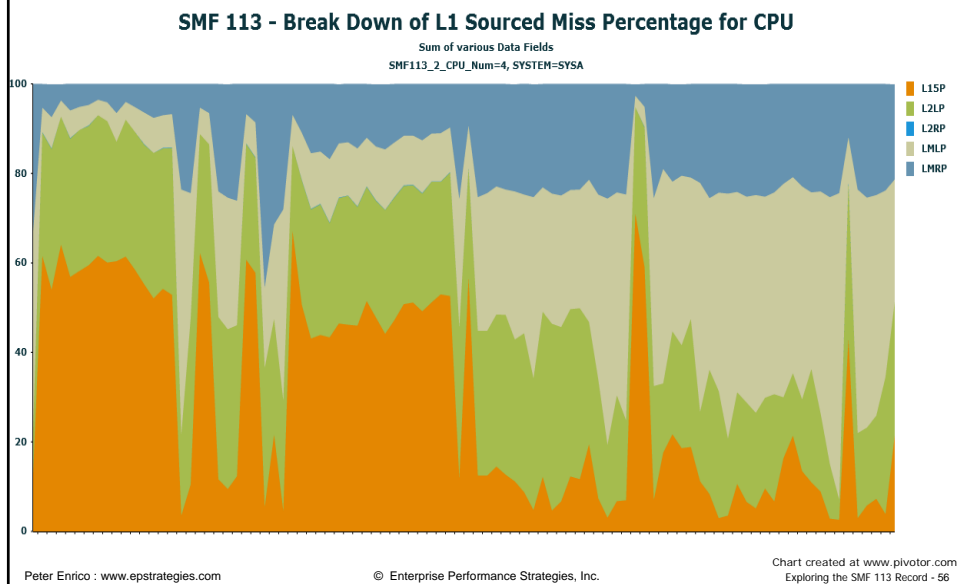
z10 HD Before/After Example – Small MP



z10 HD Before/After Example – Small MP



z10 CPU 4 (Med Pool) L1 Sourced Breakdown



Using the SMF 113 Record

- Before and After comparisons and evaluations
 - The contention index
 - CPI – Cycles per Instruction
 - Used to gauge relative increases and decreases in processor effectiveness
 - The stability index
 - PRBSTATE (Problem instruction to Total instruction ratio)
 - Used to gauge the before / after stability of the workload
 - L1 Cache Miss %
 - Effectiveness of the CPU caches
 - Breakdown of L1 Cache Miss %
 - Sourced L1.5, L2 Local, L2 Remote, Local Memory, Remote Memory
 - Improvements will show increased sourcing from areas of memory closer to the L1 cache (and CPU)

New 'Nest' Related Formulas

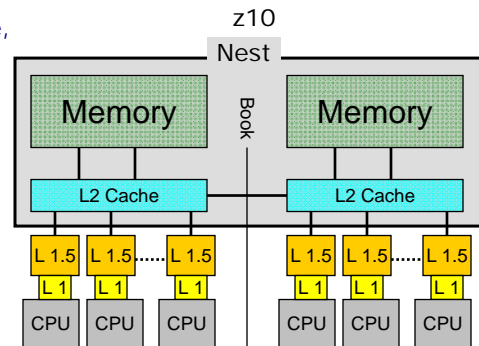
Initial information from John Burg at SHARE Summer Boston 2010

Explained by me (Peter Enrico)



Nest View of z10 Processor

- Caches can be thought as divided into
 - Private Area Caches which are part of the processor design
 - Shared Area Caches which are part of the memory hierarchy
 - This is called 'The Nest'
- During processor evaluations the Nest has many design alternatives, is very workload variable, and is influenced by many workload and configuration factors
 - Whereas Private Area Cache is more stable for a workload (since not influenced as much by configuration)
- For this reason, the new LSPR workloads focus on the Nest



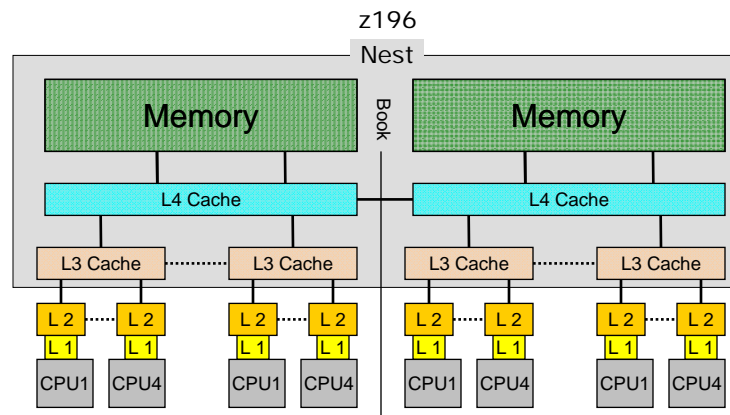
Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 59

Nest View of z196 Processor

- Caches can be thought as divided into
 - Private Area Caches which are part of the processor design
 - Shared Area Caches which are part of the memory hierarchy
 - This is called 'The Nest'



Peter Enrico : www.epstrategies.com

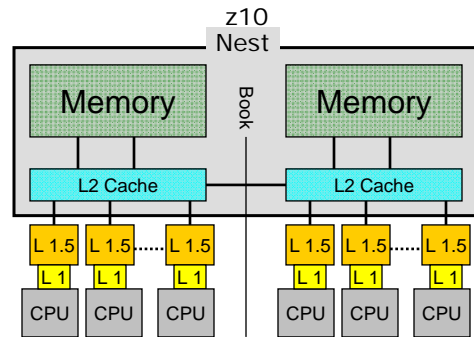
© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 60



Evaluating Processor / Nest / Workload Relationship

- ❑ There is a desire to understand the variability of processor capacity relative to the workload 'usage' of the Nest
 - L2 local, L2 remote
 - Local Memory, Remote Memory
- ❑ Good News: The SMF 113 provides insight into the number of times L1 was sourced from the Nest
 - L2 local, L2 remote
 - Local Memory, Remote Memory
- ❑ Less Than Good News: The SMF 113 does not provide the penalty cycles for the individual levels of cache
 - Only total penalty cycles for all L1 sourcing (for I and D cache)
 - Thus outside Nest L1.5 included
- ❑ Good News: We have 'Gary King' constants to estimate



Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 61

COUNTER SET= BASIC (Reminder)

- ❑ Activity count for CPU when in both problem and supervisor state
 - Counters for general purpose processors, zIIPs, and zAAPs
- ❑ 0: CYCLE COUNT
 - ❑ Number of CPU cycles, excluding the number of cycles CPU is in wait state
- ❑ 1: INSTRUCTION COUNT
 - ❑ Number of supervisor and problem state instructions executed by the CPU
- ❑ 2: L1 I-CACHE DIRECTORY-WRITE COUNT
 - ❑ Number of writes to instruction cache (and includes data cache if unified cache)
- ❑ 3: L1 I-CACHE PENALTY CYCLE COUNT
 - ❑ Instruction cache penalty cycle count (and includes data cache if unified cache)
- ❑ 4: L1 D-CACHE DIRECTORY-WRITE COUNT
 - ❑ Number of writes to data cache (and zero if unified cache)
- ❑ 5: L1 D-CACHE PENALTY CYCLE COUNT
 - ❑ Data cache penalty cycle count (and zero if unified cache)

Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 62

Z10 New CPI Formulas (for Contention Index)

- **Total Machine Cycles per Instruction (Actual CPI)**

$$CPI = (Total\ Cycles / Total\ Instructions)$$

$$= (B0/B1)$$
- **Estimate Finite CPI (Est Finite CPI)**

$$EF_CPI = (Penalty\ Cycles / Total\ Instructions)$$

$$= ((B3+B5)/B1) * .84 \quad (where\ .84\ is\ Gary\ King\ z10\ Constant)$$
 - Think of this as Penalty Cycles per instruction, but since there is an 'overlap' of sourcing cycles from the different levels, we need scale value downward to exclude these 'overlap' cycles
 - Thus the multiplication by King constant .84 for the z10 and .63 for z196
 - Note a lower value for z196 to show improved overlapping
- **Estimated Instruction Complexity CPI (Est Instr Cmplx CPI)**

$$EIC_CPI = ((CPI) - (EF_CPI))$$
 - Think of this as CPI if there was an infinitely large L1 cache (i.e. no penalty cycles)

Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 63

z196 New CPI Formulas (for Contention Index)

- **Total Machine Cycles per Instruction (Actual CPI)**

$$CPI = (Total\ Cycles / Total\ Instructions)$$

$$= (B0/B1)$$
- **Estimate Finite CPI (Est Finite CPI)**

$$EF_CPI = (Penalty\ Cycles / Total\ Instructions)$$

$$= ((B3+B5)/B1) * .63 \quad (where\ .63\ is\ Gary\ King\ z196\ Constant)$$
 - Think of this as Penalty Cycles per instruction, but since there is an 'overlap' of sourcing cycles from the different levels, we need scale value downward to exclude these 'overlap' cycles
 - Thus the multiplication by King constant .84 for the z10 and .63 for z196
 - Note a lower value for z196 to show improved overlapping
- **Estimated Instruction Complexity CPI (Est Instr Cmplx CPI)**

$$EIC_CPI = ((CPI) - (EF_CPI))$$
 - Think of this as CPI if there was an infinitely large L1 cache (i.e. no penalty cycles)

Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 64

Estimated Sourcing Cycles per L1 Miss

Estimated Sourcing Cycles per L1 Miss (from the Nest)

$$Est_SCPL1M = (Penalty\ Cycles / Penalty\ Writes)$$

$$= ((B3+B5)/(B2+B4)) * .84 \quad (where\ .84\ is\ Gary\ King\ z10\ Constant)$$

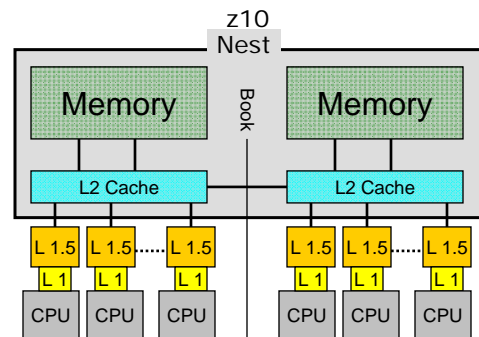
- Since penalty cycles and writes include sourcing from L1.5, and since we are only interested in the Nest, we need to multiple by .84 to compensate 'Sourcing Cycles per L1 Miss' downward

Scaling factors for sourcing overlap:

- .84 for z10
- .63 for z196

My random wondering...

- 4.4GHz / 5.2 GHz = ~.846
- Does this have any relationship to .84 scaling?



Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

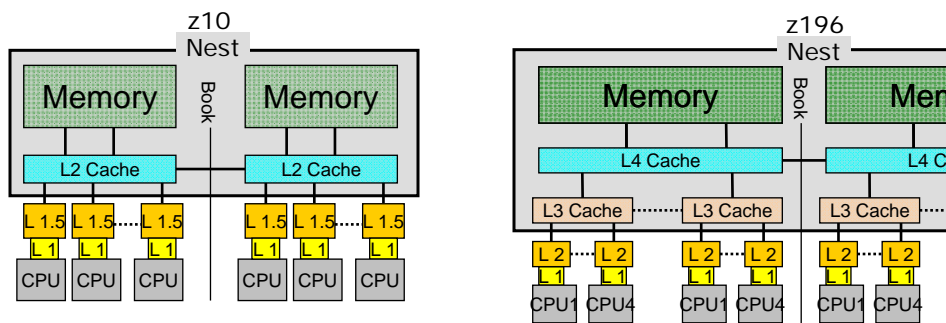
Exploring the SMF 113 Record - 65

Relative Nest Intensity

- "Relative Nest Intensity reflects the distribution and latency of sourcing from shared caches and memory" (J Burg, IBM)

New LSPR workloads will be based on RNI

- Low RNI : <0.5
- Med RNI: 0.5 – 1.0
- High RNI > 1.0



Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 66

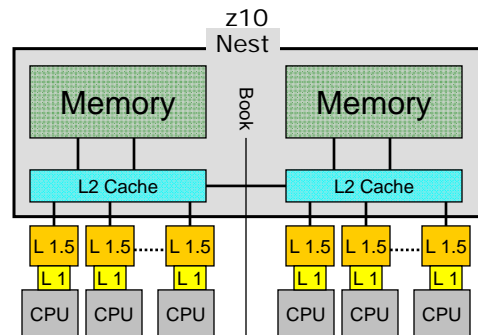
Relative Nest Intensity (for z10)

- “Relative Nest Intensity reflects the distribution and latency of sourcing from shared caches and memory” (J Burg, IBM)

$$RNI = ((1.0 * L2LP) + (2.4 * L2RP) + (7.5 * MEMP)) / 100$$

(where weights are Gary King Constants)

- L2 Local Sourcing %
 - Weighted by 1.0
- L2 Remote Sourcing %
 - Weighted by 2.4
- Memory Sourcing % (Local + Remote)
 - Weighted by 7.5
- Note: L1.5 not considered since not part of Nest



Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 67

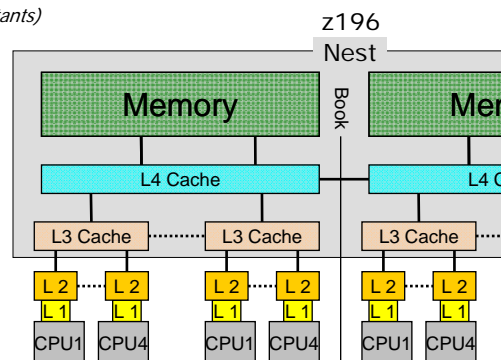
Relative Nest Intensity (for z196)

- “Relative Nest Intensity reflects the distribution and latency of sourcing from shared caches and memory” (J Burg, IBM)

$$RNI = 1.6 * ((0.4 * L3P) + (1.0 * L4LP) + (2.4 * L4RP) + (7.5 * MEMP)) / 100$$

(where weights are Gary King Constants)

- Note: L2 not part of nest so not factored in
- Note benefit L3P relative to other caches

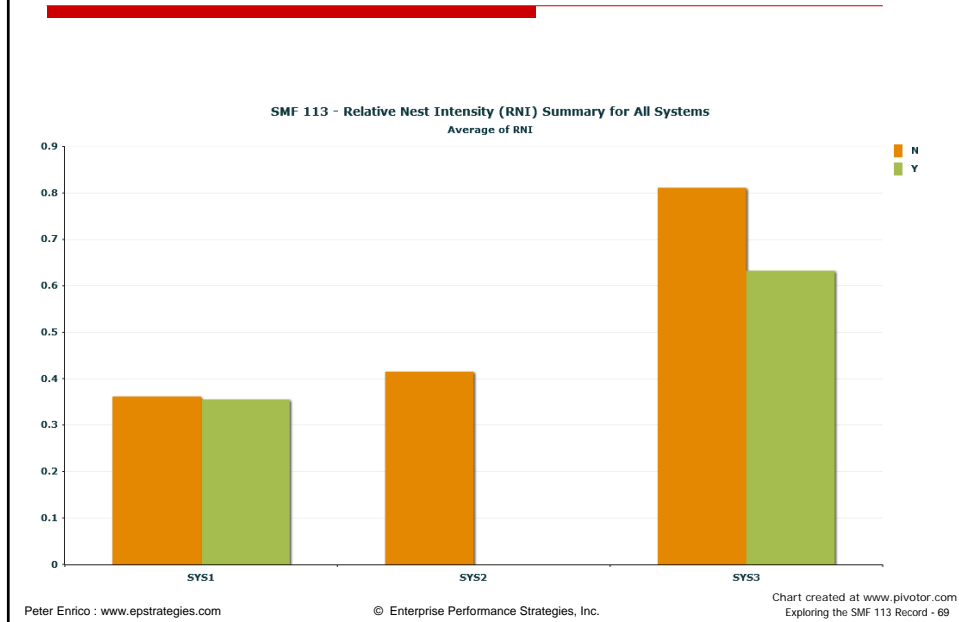


Peter Enrico : www.epstrategies.com

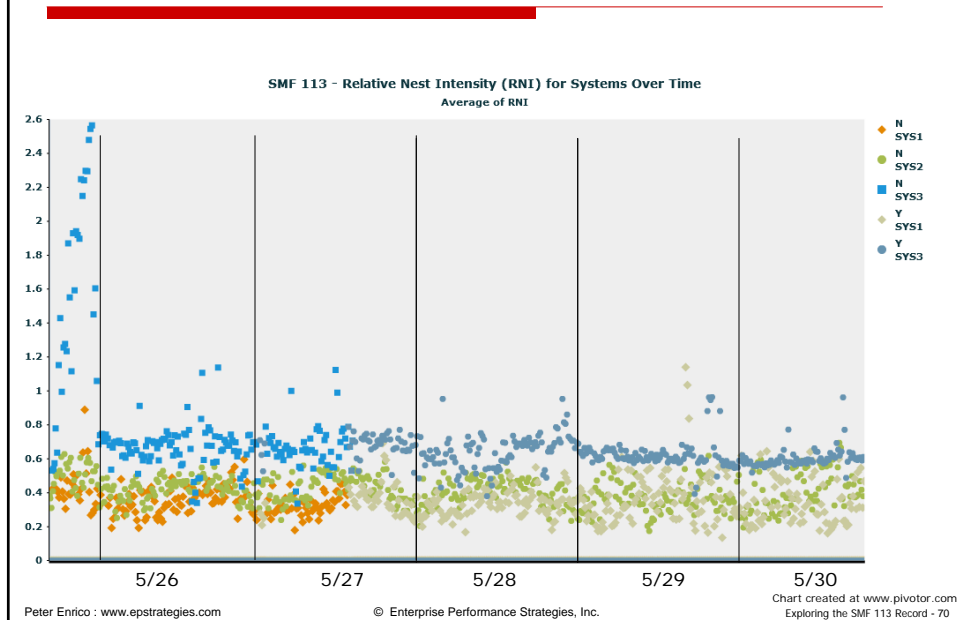
© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 68

HD (Y/N) Before/After Example – RNI Small MP z10



HD Before/After Example – RNI Small MP z10



LSPRs and SMF 113s and RNI Hint

- SMF 113 measurements are now used to provide guidelines / hints for LSPR and zPCR processor sizing
- This RNI Hint table was documented in the Large System Performance Reference (LSPR)
 - Document Number SC28-1187-14
- The next slide shows an example of an LSPR chart used for processor sizing
- Using the SMF 113 records you now need to calculate
 - L1MP - L1 Miss Percentage
 - RNI - Relative Nest Intensity
- Note: This table and these guidelines are expected to change as more is learned from the SMF 113 records

L1MP	RNI	Workload Hint
<3%	>= 0.75	AVERAGE
	< 0.75	LOW
3% to 6%	>1.0	HIGH
	0.6 to 1.0	AVERAGE
	< 0.6	LOW
>6%	>=0.75	HIGH
	< 0.75	AVERAGE

LSPR Table Example

IBM System z9 EC
(System z9 2094-701 = 1.00)

Processor	#CP	PCI**	MSU***	Low*	Average*	High*
2094-601	1	454	65	0.81	0.81	0.81
2094-602	2	880	127	1.6	1.57	1.53
2094-603	3	1303	184	2.38	2.33	2.23
2094-604	4	1720	240	3.13	3.07	2.92
2094-605	5	2109	292	3.87	3.77	3.58
2094-606	6	2482	339	4.59	4.43	4.21
2094-607	7	2842	385	5.3	5.08	4.81
2094-608	8	3188	428	5.99	5.69	5.37
2094-701	1	560	81	1	1	1
2094-702	2	1086	158	1.98	1.94	1.89
2094-703	3	1607	229	2.93	2.87	2.75
2094-704	4	2122	298	3.86	3.79	3.6
2094-705	5	2601	363	4.78	4.65	4.42
2094-706	6	3062	422	5.67	5.47	5.19
2094-707	7	3505	479	6.54	6.26	5.93
2094-708	8	3932	532	7.38	7.02	6.62



Reports / SMF 113 Processing Offer !!!

- Special Reports Offer!
 - See your SMF 113 records in chart and table format
 - Please contact me, Peter Enrico for instructions for sending raw SMF data
 - Send an email to peter.enrico@epstrategies.com
 - Deliverable:
 - Dozens of SMF 113 based reports (charts and tables)
 - Summary by system
 - Summary by CPU
 - Before / After comparison reports
 - Raw counter reports
 - Much more...
 - One-on-one phone call to explain your SMF 113 measurements

Current 2011 Class Schedule

- WLM Performance and Re-evaluating of Goals
 - Instructor: Peter Enrico
 - June 6 - 10, 2011 Columbus, Ohio USA
 - September 12 - 16, 2011 Baltimore, Maryland, USA
- Essential z/OS Performance Tuning
 - Instructor: Peter Enrico and Tom Beretvas
 - May 9 - 13, 2011 St. Louis, Missouri, USA
- Parallel Sysplex and z/OS Performance Tuning
 - Instructor: Peter Enrico
 - May 16 - 20, 2011 Omaha, Nebraska USA
 - September 19 - 23, 2011 Dallas, Texas, USA
- z/OS Capacity Planning and Performance Analysis
 - Instructor: Ray Wicks
 - August 15 - 17, 2011 Columbus, Ohio, USA



Crypto Counters

(If time permits)

COUNTER SET = CRYPTO-ACTIVITY

- zArchitecture include a something called message-security assist that supports cryptographic operations
- On z10 machines, 2 CPs can share a single crypto processor
 - So there is the concept of interference
 - When 2 sharing CPs attempt to use same coprocessor, one will be blocked until a predetermined time slice has passed
- Counter Set = Crypto-Activity
 - Contains counters of the crypto processor activity
 - Interference activity
- A cryptographic coprocessor group
 - Contains of a cipher coprocessor for DEA, AES
 - Contains a hash coprocessor for SHA
- Crypto processors with 2 CPs so blocking possible



COUNTER SET= CRYPTO-ACTIVITY

64: PRNG FUNCTION COUNT
65: PRNG CYCLE COUNT
66: PRNG BLOCKED FUNCTION COUNT
67: PRNG BLOCKED CYCLE COUNT

68: SHA FUNCTION COUNT
69: SHA CYCLE COUNT
70: SHA BLOCKED FUNCTION COUNT
71: SHA BLOCKED CYCLE COUNT

72: DEA FUNCTION COUNT
73: DEA CYCLE COUNT
74: DEA BLOCKED FUNCTION COUNT
75: DEA BLOCKED CYCLE COUNT

76: AES FUNCTION COUNT
77: AES CYCLE COUNT
78: AES BLOCKED FUNCTION COUNT
79: AES BLOCKED CYCLE COUNT

COUNTER SET= CRYPTO-ACTIVITY

- **PRNG - pseudorandom number generator**
 - Algorithm for generating a sequence of numbers that approximates the properties of random numbers. A PRNG is normally just an algorithm where the same initial starting values will yield the same sequence of outputs
- **SHA - Secure Hash Algorithm**
 - A set of five cryptographic hash functions designed by the National Security Agency (NSA) and published by the NIST as a U.S. Federal Information Processing Standard,
- **DEA - Data Encryption Algorithm**
 - A block cipher (a form of shared secret encryption) that was selected by the National Bureau of Standards as an official Federal Information Processing Standard (FIPS) for the United States in 1976 and which has subsequently enjoyed widespread use internationally.
 - It is based on a symmetric-key algorithm that uses a 56-bit key
- **AES - Advanced Encryption Standard (a.k.a. Rijndael)**
 - Iterated block cipher algorithm recently chosen by the National Institute of Science and Technology (NIST) as the Advanced Encryption Standard (AES)
 - It super cedes the Data Encryption Standard (DES).
 - NIST selected Rijndael as the standard symmetric key encryption algorithm to be used to encrypt sensitive (unclassified) American federal information



COUNTER SET= CRYPTO *.CNT file excerpt example

```
COUNTER SET= CRYPTO-ACTIVITY
COUNTER IDENTIFIERS:
  64: PRNG FUNCTION COUNT
  65: PRNG CYCLE COUNT
  66: PRNG BLOCKED FUNCTION COUNT
  67: PRNG BLOCKED CYCLE COUNT
  68: SHA FUNCTION COUNT
  69: SHA CYCLE COUNT
  70: SHA BLOCKED FUNCTION COUNT
  71: SHA BLOCKED CYCLE COUNT
  72: DEA FUNCTION COUNT
  73: DEA CYCLE COUNT
  74: DEA BLOCKED FUNCTION COUNT
  75: DEA BLOCKED CYCLE COUNT
  76: AES FUNCTION COUNT
  77: AES CYCLE COUNT
  78: AES BLOCKED FUNCTION COUNT
  79: AES BLOCKED CYCLE COUNT

START TIME: 2010/03/16 11:25:21  START TOD: C5AFCE3D7E54909C
END TIME:   2010/03/16 14:44:15  END TOD:   C5AFFAB2674B2F8C
COUNTER VALUES (HEXADECEMAL) FOR CPU 00 (CPU SPEED = 4404 CYCLES/MIC):
64- 67 0000000000000000 0000000000000000 0000000000000000 0000000000000000
68- 71 000000000000009D 0000000000043B46 0000000000000000 0000000000000000
72- 75 0000000000000000 0000000000000000 0000000000000000 0000000000000000
76- 79 0000000000000000 0000000000000000 0000000000000000 0000000000000000
```

CPU Measurement Facility Sampling

(If time permits)
(Note: Real quickly for completeness before discussing SMF 113)



Questions to ask of the Sampling Measurements

- What was the virtual storage layout
 - General system mapping
 - Primary area mapping
- Where in the addressing range of the different areas of virtual storage did each logical processor spend its time processing
 - Module, CSECT, Entry point, address range
 - How often
- In what memory area did the CPU spend time processing?
 - Nucleus, MLPA, PLPA, FLPA, Private, Common
- In whose private area did the CPU spend time processing
 - By ASID or by Jobname
- From what VOLSER and library (or USS file) was the load module loaded?

Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 81

Introduction to HIS Sampling Data

- Hardware event data collection can optionally sample processor activity
 - Useful instrumentation of programming environments
- Types of sampling
 - Basic – instruction addresses, programming states, parameter sets
 - Diagnostic – provides details of internal hardware design
 - Sampling frequency : default 800000 per minute (i.e. 8M samples in 10 minutes)
- Results recorded in the USS file system
 - Map file
 - Named SYSHISyyyymmdd.hhmmss.MAP
 - Contains load module mapping information
 - Sample data files
 - Named SYSHISyyyymmdd.hhmmss.SMP.cpu#
 - Large / voluminous files written for each z/OS logical processor on which data collection has been run
 - Contains sample data of the addresses on the instructions found executing during the sample, as well as some state information about the logical processor

Peter Enrico : www.epstrategies.com

© Enterprise Performance Strategies, Inc.

Exploring the SMF 113 Record - 82

Introduction to HIS Sampling Data

- Contents of load module mapping output file
 - MAPONLY option
 - Used to collect only load module mapping information
 - MAPASID
 - Used to map a specific set of ASIDs (address spaces) to produce private area load module map
 - Example MAPASID=ALL, or MAPASID=(A,1E,30)
 - MAPJOB
 - Used to map a specific set of job names to produce private area load module maps
 - Example: MAPJOB=(WLMSPAS1,MDD*,TSOA*)

Example of Portion of .MAP File

```
I SYS SY1
I SMFIIBM2
I OS z/OS
I FMIDHBB7750
I DATE08091
I TIME16311902
I MAP V1R1
I LPID00000000
I MACH00002097
B BDY PRIVATE 0000000008FFFFF
B BDY CSA 0090000000BB2FFF
. . .
CNNUC IECVPRNT00FD700000FD74F7
ENNUC PRDSE 00FD7006
ENNUC PRTSIO 00FD700C
. . .
MPPLPAIGG019T800BDE46000BDE53F
MPPLPAIGG019TX00BDE54000BDE5DF
. . .
MMMLPAIEFACTRT06663CD006663EBF VOLSER=CTTPAKDSN=ARTMVS.EXIT.S.LOAD
CMMLPAIEFACTRT06663CD006663EBF
MMMLPAEZBREARR24CA000024D16FFF
. . .
MX0002IEAVXMAS25B0000025B01FC7PCAUTH VOLSER=ZD110 DSN=SYS1.NUCLEUS
CX0002IEAVXMAS25B0000025B01167
. . .
MX0003IAXDINIT25B0000025B00B77RASP VOLSER=ZD110 DSN=SYS1.NUCLEUS
CX0003IAXDI 25B0000025B00B77
```



Example of Portion of .MAP File

- The map file explains the virtual storage layout
 - By itself it is sort of interesting
 - But when exploited by the sampling file we can discover where processor cycles are spending their time in the code

Field name	Offset	Length	Format	Description
Record type	0	1	Text	I=Information, B=Boundary, M=Module, C=CSECT, E=Entry Point
Memory area	1	1	Text	N=Nucleus, M=MLPA, P=PLPA, F=FLPA, X=Private area, C=Common
ASID	2	4	Printable Hex	ASID (for Private area) or Record Type (for other records)
Name	6	8	Text	Short name (may be blank, may not be unique)
Start address	14	8	PrintableHex	Start address (Record types B, M, C and E only)
End address	22	8	Printable Hex	End address (Record types B, M, and C only)
Job name	30	8	Text	Job name of the address space (Module records only)
Long name	38	To end of record	Text	VOLSER=xxxxxx, DSN=xxx for modules loaded from MVS datasets, pathname for HFS modules, longname for any names longer than 8 bytes

Example of Portion of .SMP Contents

- Basic Sampling
 - Number of unique completed instructions executed simultaneously during sampling cycle
 - State bits
 - DAT mode, Wait sate, Problem state, Address space control ,Primary ASN
 - Instruction address
 - Of an instruction on the logical processor that was executing during the sampling cycle
 - Guest program parameter
 - Program parameter set by most recent SET PROGRAM PARAMETER instruction (by the processor running at the virtual machine level)
 - Host program parameter
 - Program parameter set by most recent SET PROGRAM PARAMETER instruction executed by the processor running under VM
 - TCB address
 - SRB mode indicator
 - Home ASID
 - Task ID token
 - WEB address of SRB
 - Etc.

Using the Sampling Data

- This is a software vendor opportunity
- A tool must be developed to apply the MAP data to the sampling data
- Examples of reports that need to be developed
 - Where (i.e. module / CSECT / offset) where instruction cycles are being spent
 - Heaviest hit modules and/or CSECTs and/or instruction ranges
 - Module flow