17665 Processor reporting:

RMF and Hardware Instrumentation Services (HIS)

John Ticic
www.IntelliMagic.com
Agenda

This is a two part presentation.

Part 1 will examine what detailed processor reporting is available in RMF.

Part 2 will introduce and discuss the value of continuous HIS data collection and reporting.
Agenda

This is a two part presentation.

Part 1 will examine what detailed processor reporting is available in RMF.

Part 2 will introduce and discuss the value of continuous HIS data collection and reporting.

Part 3 will look at tying these together.
Who is IntelliMagic

• A leader in Availability Intelligence
  – New visibility of threats to continuous availability by automatic interpretation of RMF/SMF/Config data using built-in expert knowledge

• Over 20 years developing storage performance solutions

• Privately held, financially independent

• Customer centric and highly responsive

• Products used daily at some of the largest sites in the world
John Ticic


Joined IntelliMagic in 2008 as a Senior Consultant

Specialties include:

- Disk/Tape performance
- z/OS Performance
- z/OS, zSeries implementation
- Presenting (I/O classes, SHARE, GSE,..)
RMF Processor Reporting
Measurement Sources

RMF Monitor 1 (Long term collection) settings:

CPU
WKLD

RMF Post Processor Reports

REPORTS(CPU)
SYSRPTS(WLMGL(..))
Measurement Sources

Decide on an interval time (synchronize with SMF).

In ERBRMF.. Specify:

SYNC(SMF)

SMF Records created are:

SMF 70 CPU, PRSM, ICF, Cryptographic
SMF 72 WLM
# RMF Report Samples

## CPU Activity

<table>
<thead>
<tr>
<th>NUM</th>
<th>TYPE</th>
<th>ONLINE</th>
<th>LPAR BUSY</th>
<th>MVS BUSY</th>
<th>PARKED</th>
<th>LOG PROC</th>
<th>I/O INTERRUPTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CP</td>
<td>100.00</td>
<td>99.96</td>
<td>100.00</td>
<td>0.00</td>
<td>100.00</td>
<td>95.31</td>
</tr>
<tr>
<td>1</td>
<td>CP</td>
<td>100.00</td>
<td>99.60</td>
<td>100.00</td>
<td>0.00</td>
<td>100.00</td>
<td>90.00</td>
</tr>
<tr>
<td>2</td>
<td>CP</td>
<td>100.00</td>
<td>99.58</td>
<td>99.97</td>
<td>0.00</td>
<td>100.00</td>
<td>89.00</td>
</tr>
<tr>
<td>3</td>
<td>CP</td>
<td>100.00</td>
<td>99.58</td>
<td>99.97</td>
<td>0.00</td>
<td>100.00</td>
<td>89.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PARTITION DATA</th>
<th>-- MSU --</th>
<th>-- CAPPING --</th>
<th>-- DISPATCH TIME DATA --</th>
<th>-- LOGICAL PROCESSORS --</th>
<th>-- PHYSICAL PROCESSORS --</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>S</td>
<td>WGI</td>
<td>DEF</td>
<td>ACT</td>
<td>DEF</td>
</tr>
<tr>
<td>S59</td>
<td>A</td>
<td>801</td>
<td>0</td>
<td>502</td>
<td>0</td>
</tr>
<tr>
<td>S60</td>
<td>A</td>
<td>800</td>
<td>0</td>
<td>500</td>
<td>0</td>
</tr>
<tr>
<td>S51</td>
<td>A</td>
<td>100</td>
<td>0</td>
<td>53</td>
<td>0</td>
</tr>
<tr>
<td>S55</td>
<td>A</td>
<td>100</td>
<td>0</td>
<td>53</td>
<td>0</td>
</tr>
<tr>
<td>S58</td>
<td>A</td>
<td>999</td>
<td>0</td>
<td>493</td>
<td>0</td>
</tr>
</tbody>
</table>

## LPAR Cluster Report

<table>
<thead>
<tr>
<th>CLUSTER</th>
<th>PARTITION</th>
<th>SYSTEM</th>
<th>INIT</th>
<th>MIN</th>
<th>MAX</th>
<th>AVG</th>
<th>MIN</th>
<th>MAX</th>
<th>Defined</th>
<th>Actual</th>
<th>Actual</th>
<th>Total</th>
<th>Central</th>
<th>Expanded</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVPLEX5</td>
<td>S51</td>
<td>S51</td>
<td>500</td>
<td>100</td>
<td>999</td>
<td>100</td>
<td>87.5</td>
<td>0.0</td>
<td>28</td>
<td>3.0</td>
<td>31.11</td>
<td>4.67</td>
<td>10240</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>S55</td>
<td>S55</td>
<td>500</td>
<td>100</td>
<td>999</td>
<td>101</td>
<td>83.3</td>
<td>0.0</td>
<td>20</td>
<td>20.0</td>
<td>6.01</td>
<td>6.01</td>
<td>10240</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>S58</td>
<td>S58</td>
<td>500</td>
<td>100</td>
<td>999</td>
<td>100</td>
<td>83.3</td>
<td>0.0</td>
<td>20</td>
<td>20.0</td>
<td>43.70</td>
<td>43.70</td>
<td>10240</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>S59</td>
<td>S59</td>
<td>500</td>
<td>100</td>
<td>999</td>
<td>801</td>
<td>0.0</td>
<td>0.0</td>
<td>20</td>
<td>20.0</td>
<td>44.53</td>
<td>44.53</td>
<td>10240</td>
<td>N/A</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td>2801</td>
<td>68</td>
<td>125.4</td>
<td>98.90</td>
<td>40960</td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Another Way of Showing the Data

There are many ways of visualizing the data, this presentation is using IntelliMagic Vision
What can we do with this Data?

How well am I servicing Tasks?

Who is waiting and why?
Breakdown Usage by LPAR/Workload

Which workloads are using the resources (e.g. zIIP.)
Do we have Latent Demand

Which workloads could use more resources (e.g. zIIP.)
Hardware Instrumentation Services (HIS) Data Collection and Reporting
What are we looking at?
(e.g. zEC12)
Possible Data

Count of the number of instructions executed by the CPU

Count of the number of Cache Level 1 misses.
   Misses need to be resolved and cost extra cycles.

Count of staging a miss from Level 2 Cache

Depending on the hardware architecture, there are different levels of Cache, and different books. They all have a different penalty for staging misses.

Many, many more ....
About the Data Collection

Collection can be short-term sampling (e.g. to investigate a specific problem) or long term data gathering. *Data is written per LPAR.*

SMF data is currently available as:

- 113 subtype 1       Interval (i.e. Deltas)
- 113 subtype 2       Continuous

*z/OS 2.1 or later is required for subtype 1 creation.*

All charts show here are based on subtype 1 data.
About the Data

Different counter sets are available (see appendix for documentation):

- Basic
- Problem State
- Crypto

- Extended

The Load-Program-Parameter and the CPU-Measurement Facilities

The CPU-Measurement Facility Extended Counters Definition for z10, z196/z114, zEC12/zBC12, and z13
Basic and Extended Counters

Faster

Slower

Extended Information

Basic Information
Setting it up - LPAR

Each LPAR needs to allow the creation of these records. These settings are located in the Image Profile on the HMC.

E.g. On the EC12 hardware, these settings are in the category Counter Facility Security Options (multiple check marks are possible.)
Setting it up – Z/OS

Details in MVS Systems Commands (SA38-0666)
Search for Hardware Event Data Collection

Define HIS Started Task to RACF
OMVS segment required for UNIX file sampling data

Enable SMF 113 recording (SMFPRM.. in PARMLIB)

Ensure that WLM assigns the HIS Started Task sufficient importance
Setting it up – Z/OS (con’t)

Start the HIS Started Task early (e.g. during the IPL procedure)

After the task is initialized, modify it to specify the desired sampling:

S HIS
HIS002I HIS INITIALIZATION COMPLETE
....wait 10 seconds ...
F HIS,BEGIN,DDNAME=PARMLIB
HIS033I HIS SERVICE PARAMETER(S) ACCEPTED
HIS011I HIS DATA COLLECTION STARTED

Activates the commands pointed to be the PARMLIB DD
Setting it up – Z/OS (con’t)

That’s it!

Now comes the tough part.

How to look at the data and what you can see!
Sample HIS Parameters Used

In SYS1.PARMLIB(HISPARM)

SMFINTVAL=SYNC,
CTRSET=(B,P,C,E),
CTRONLY

SMFINTVAL=SYNC
Synchronize with SMF (e.g. every 15 mins. on the hour)

CTRSET=(B,P,C,E)
Basic + Problem + Crypto + Extended
(or code CTRSET = HARDWARE)

CTRONLY
Limit collection to only event counter set (i.e. no
instruction sampling)
What does it look like

Based on the Basic counters.
Relative Nest Intensity

"RNI reflects the distribution and latency of sourcing from shared caches and memory"
Higher values means more cache misses to slower memory.
Cache Lifetime

Knowing the architecture allows you to look at utilization.
Possible Uses of this Data
or “How to Tie them together”
Possible Uses

Some possible uses for HIS and Processor/Workload data will be explored.

Investigating problems is the obvious one, but let’s look at some other possibilities.

- Optimizing CPU usage
- Using new architectural changes
- Using new hardware features
Optimizing CPU usage
Optimizing CPs - HiperDispatch

HiperDispatch optimizes Task distribution on CPUs by favoring CPUs that can provide a better/faster access to cache for that task.

Logical CPUs may be “parked” to improve CPU responsiveness.

The benefit is cache miss reduction.

Note: Benefit is noticed more when a large number of logical CPUs are defined to an LPAR that has a workload with a favorable memory access pattern.
Implementation

Ensure LPAR weights are set appropriately.

Set HIPERDISPATCH=YES in IEAOPTxx (it may already be a default).

Review WLM goals and importance definitions.
### RMF View

#### z/OS V2R1
- **System ID:** IBM1
- **Date:** 05/01/2015
- **Interval:** 14.59.985
- **Report Version:** V2R1 RMF
- **Time:** 10.00.00
- **Cycle:** 0.500

#### System Details
- **Seconds:**
- **CPU:** 2964
- **CPC Capacity:** 3313
- **Model:** 725
- **Change Reason:** NONE
- **HiperDispatch:** YES
- **HW Model:** N96

#### CPU Usage

<table>
<thead>
<tr>
<th>Num</th>
<th>Type</th>
<th>Online</th>
<th>LPAR Busy</th>
<th>MVS Busy</th>
<th>Parked</th>
<th>Prod</th>
<th>Util</th>
<th>Share %</th>
<th>Rate</th>
<th>% Via TPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CP</td>
<td>100.00</td>
<td>67.61</td>
<td>67.47</td>
<td>0.00</td>
<td>100.00</td>
<td>67.61</td>
<td>100.0</td>
<td>5713</td>
<td>15.17</td>
</tr>
<tr>
<td>1</td>
<td>CP</td>
<td>100.00</td>
<td>45.41</td>
<td>45.37</td>
<td>0.00</td>
<td>100.00</td>
<td>45.41</td>
<td>100.0</td>
<td>4443</td>
<td>16.20</td>
</tr>
<tr>
<td>2</td>
<td>CP</td>
<td>100.00</td>
<td>40.81</td>
<td>41.87</td>
<td>0.00</td>
<td>100.00</td>
<td>40.81</td>
<td>69.5</td>
<td>316.3</td>
<td>51.44</td>
</tr>
<tr>
<td>3</td>
<td>CP</td>
<td>100.00</td>
<td>7.00</td>
<td>44.48</td>
<td>82.42</td>
<td>100.00</td>
<td>2.98</td>
<td>69.5</td>
<td>0.0</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>CP</td>
<td>100.00</td>
<td>2.98</td>
<td>45.50</td>
<td>91.38</td>
<td>100.00</td>
<td>1.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>CP</td>
<td>100.00</td>
<td>1.50</td>
<td>52.00</td>
<td>95.32</td>
<td>100.00</td>
<td>1.09</td>
<td>0.0</td>
<td>0.0</td>
<td>0.00</td>
</tr>
<tr>
<td>6</td>
<td>CP</td>
<td>100.00</td>
<td>1.09</td>
<td>46.35</td>
<td>96.21</td>
<td>100.00</td>
<td>0.00</td>
<td>0.0</td>
<td>0.0</td>
<td>0.00</td>
</tr>
<tr>
<td>7</td>
<td>CP</td>
<td>100.00</td>
<td>3.61</td>
<td>3.98</td>
<td>0.00</td>
<td>100.00</td>
<td>3.61</td>
<td>0.0</td>
<td>0.0</td>
<td>0.00</td>
</tr>
<tr>
<td>8</td>
<td>CP</td>
<td>100.00</td>
<td>0.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>0.00</td>
<td>0.0</td>
<td>0.0</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**Total/Average:** 18.49 50.99 100.00 18.49 269.5 10472 16.70

#### I/O Interrupts

<table>
<thead>
<tr>
<th>Num</th>
<th>Type</th>
<th>Online</th>
<th>LPAR Busy</th>
<th>MVS Busy</th>
<th>Parked</th>
<th>Prod</th>
<th>Util</th>
<th>Share %</th>
<th>Rate</th>
<th>% Via TPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>IIP</td>
<td>100.00</td>
<td>14.94</td>
<td>15.33</td>
<td>0.00</td>
<td>100.00</td>
<td>14.94</td>
<td>60.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>IIP</td>
<td>100.00</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>IIP</td>
<td>100.00</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>IIP</td>
<td>100.00</td>
<td>3.61</td>
<td>3.98</td>
<td>0.00</td>
<td>100.00</td>
<td>3.61</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Total/Average:** 4.64 9.66 100.00 4.64 60.0
Lots of parked CPs in the LPAR. Let’s look at the SMF 113 data.
Looking at the big picture, LVL1 cache miss doesn’t show us much. Let’s look at the individual processors.
Relative Nest Intensity (RNI)

Two very distinct groups of processors. Let’s look at Proc. 00 (Vertical High) and Proc. 06 (Vertical Low).

Note: Logical Processor 3 is Physical Processor 06 for LPAR IBM1
Relative Nest Intensity (RNI)

Can we explain what is happening after 5:00 PM!
Relative Nest Intensity (RNI)

This LPAR (IBM2) is also running on this CEC and CP#6 is Vertical High. But, there are other LPARs active and using CP#6 at times.
CPU Utilization

LPAR IBM3 un-parks processors at various intervals.
Relative Nest Intensity (RNI)

LPAR IBM3 is using CP#6 (Vertical Low), and it looks like this is reflected in the RNI value for LPAR IBM1, and slightly affects IBM-2.
Conclusions

HiperDispatch is doing a good job to minimize cache disruption.

i.e. What would the cache miss picture look like if CP 06 were always available on LPAR IBM1 and LPAR IBM3 to run work.

Better would be to observe RNI before/after HiperDispatch implementation.

The RMF and HIS records can help!
Using New Architectural Changes
Using New Architectural Changes
Larger Pages

Reducing Address Translation misses.

Implementing larger real storage pages should reduce the number of TLB (Translation Lookaside Buffer) misses since a single TLB entry will be able to cover a larger span of address translations.

The TLB is related to Dynamic Address Translation (DAT) and is used to translate a virtual address to a real address.

TLB is divided into an instruction and data portion
Implementation

SYS1.PARMLIB(IEASYS..)

LFAREA=50%, 50% OF ALLOWABLE STORAGE 1 MB FIXED

Results in:

IAR040I REAL STORAGE AMOUNTS:
  TOTAL AVAILABLE ONLINE: 24064M
  LFAREA LIMIT FOR xM, xG, OR xT : 17203M
  LFAREA LIMIT FOR SUM OF 1M= AND 2G= : 15974M
  LFAREA LIMIT FOR 2GB PAGES FOR 2G= : 7
IAR048I LFAREA=50% WAS PROCESSED WHICH RESULTED IN 9984 1MB PAGES AND 0 2GB PAGES.
Implementation (con’t)

For example: Allow DB2 buffer pools to use 1MB frames (e.g. BP0)

```
-DBBG ALTER BUFFERPOOL(BP0) VPSIZE(20000) PGFIX(YES) FRAMESIZE(1M)
```

Should result in:

```
DSNB546I -DBBG PREFERRED FRAME SIZE 1M
20000 BUFFERS USING 1M FRAME SIZE ALLOCATED
```
Expectations

Larger real storage frames will result in fewer virtual to real translation misses.

This should result in an improvement in processor usage and elapsed time.

Expect the most benefit for long running work with high memory access frequency.
What are we look for?

Reducing the TLB Miss will result in fewer CPU cycles.
(z13 Sample)
What does the TLB miss look like.

Estimated TLB1 CPU Miss % of Total CPU (%)
for all Processor Hardware Counters by Processor Complex serial

4K Pages

1M Pages

4 x DB2 database Batch Jobs starting at 2:00 AM
What does the miss rate look like.

Reduced LVL1 miss rate (small difference). E.g. 2.05 vs. 2.38 Million/sec
How much can we save.

Estimated Impact Cache and TLB Misses (Cycles/Inst)
for all Processor Hardware Counters

A slight reduction in the estimated impact of cache and TLB misses
Conclusions

A very small system and a very small workload sample.

But ... results were positive.

Shows how 4KB to 1MB (or 2GB) exploitation can easily be measured and observed.

Expect more of a CPU saving for favorable workloads.
Using New Hardware Features
Offloading CPU Cycles

What can I expect from a zEDC compression board?

Sample Jobs were run on a zBC12 (2882-N02) H06
2 CPs + 1 zIIP

14 x DFDSS DUMP of 3390-27 (DS8870)

4 different groups of Jobs were run.
Results (4 Groups)

**DFDSS DUMP Parameters:**
- COMPRESS
- HWCOMPRESS
- ZCOMPRESS(PREFERRED)
- ZCOMPRESS(NONE)

- OPTIMIZE(1) defaulted
- CMPSC Instruction
- zEDC hardware
- no compression
ZCOMPRESS using the zEDC card is certainly the best option to reduce CPU usage, but no compression used fewer cycles.

The Jobs using the zEDC card finished much quicker than the Jobs with no compression (elapsed time.)

What does RMF say?
Delays

Using and Delay per service class (samples)
For Workload 'BATCH'

Sample values for different time periods and Compression methods:
- Compress
- HWCompress
- No Compression
- zEDC
Delays – More Detail

Using and Delay components per workload (samples)

For Workload 'BATCH', for System 'RIJN'

Compress

HWCompress

No Compression

zEDC
# How did they run?

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
</table>
| COMPRESS           | Not too bad. *Second highest CPU cost*  
Significant CPU delays. Few DASD delays  
**Velocity 65%**    |
| HWCOMPARE          | Not good! *By far highest CPU cost*  
Significant CPU delays. Fewest DASD delays. Very long execution time.  
**Velocity 42%**    |
| zEDC card          | Quite good. *Low CPU cost.*  
**Velocity 65%**    |
| No compression     | Quite good. *Lowest CPU cost* (but highest DASD allocation size!)*  
Fewest CPU delays by far. Moderate DASD delays.  
**Velocity 75%**    |
Processor Usage (%)

Fraction of time that CPUs, zIIPs and zAAPs are dispatched to do work (%)

For Processor Complex serial 'IBM-93767'

- Compress
- HWCompress
- No Compression
- zEDC
Observations

Clearly, if compression is required (i.e. to reduce disk space), the zEDC card offers the best solution.

Note: Compression also reduces the amount of disk data that needs to be replicated - but that’s another story.

So, can we see anything in the SMF 113 data to tie into the RMF data?
Instruction Complexity

Processor Cycles Per Instruction (Cycles/Inst)
For System 'RIJN' by Processor ID and Processor Type

- Compress
- HWCompress
- No Compression
- zEDC

DB2 DDF
TLB Miss %CPU

Estimated TLB1 CPU Miss % of Total CPU (%)
For System 'RIJN' by Processor ID and Processor Type

- Compress
- HWCompress
- No Compression
- zEDC

IntelliMagic
Availability Intelligence
LVL 1 Instr. Cache Miss %

Level 1 Instruction Cache Miss (Cache Directory Write) (%) for all Processor Hardware Counters by Processor Complex serial

Compress

HWCompress

No Compression

zEDC
LVL 1 Data Cache Miss %

Level 1 Data Cache Miss (Cache Directory Write) (%) for all Processor Hardware Counters by Processor Complex serial

- Compress
- HWCompress
- No Compression
- zEDC
Estimated Source Cycles L1 Miss

Estimated Sourcing Cycles per Level 1 Miss (Cycles) for all Processor Hardware Counters by System

Compress
HWCompress
No Compression
zEDC
LVL 1 Data Misses Sourced from L2

L1 Data Miss Sourced from Local L2 Data Cache (Million miss/sec)
for all Processor Hardware Counters by Processor Complex serial

Compress
HWCompress
No Compression
zEDC
Estimated Impact (Breakdown)

Estimated Impact Cache and TLB Misses (Cycles/Inst)
for all Processor Hardware Counters

- Processor Cycles Per Instruction (Cycles/Inst)
- Estimated Impact of Cache and TLB Misses (Cycles/Inst)
- L2 Cache Miss Cycle Estimate (Cycles/Instruction)
- L3 On-Chip Cycle Estimate (Cycles/Instruction)
- L3 Off-Chip Cycle Estimate (Cycles/Instruction)
- L3 Off-Book Cycle Estimate (Cycles/Instruction)
- L4 On-Chip Cycle Estimate (Cycles/Instruction)
- L4 Off-Chip Cycle Estimate (Cycles/Instruction)
- Mem On-Book Cycle Estimate (Cycles/Instruction)
- Mem Off-Book Cycle Estimate (Cycles/Instruction)
- TLB Data Miss Cycle Estimate (Cycles/Instruction)

Compress
HWCompress
zEDC
No Compression
<table>
<thead>
<tr>
<th>SMF 113 Data Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COMPRESS</strong></td>
</tr>
<tr>
<td>Few LVL 1 cache misses but high sourcing cycles for misses</td>
</tr>
<tr>
<td><strong>HWCOMPARE</strong></td>
</tr>
<tr>
<td>Complex instructions (large number of cycles per instr.)</td>
</tr>
<tr>
<td>High LVL1 Instruction + Data cache misses</td>
</tr>
<tr>
<td><strong>zEDC card</strong></td>
</tr>
<tr>
<td>High LVL 1 Instruction cache misses</td>
</tr>
<tr>
<td><strong>No compression</strong></td>
</tr>
<tr>
<td>High LVL 1 Data cache misses</td>
</tr>
<tr>
<td>High sourcing cycles for LVL 1 misses</td>
</tr>
</tbody>
</table>
Conclusion

If you’re using HWCOMPRESS – think about a zEDC card.

If you want to compress disk data (for whatever reason), think about a zEDC card.

Note: More SMF data is available to observe zEDC card usage (SMF 30, 74.9)

Possible presentation topic for SHARE 2016?
Summary/Conclusions
Conclusions

Implementing large frames (1MB, 2GB) can reduce TLB misses. This can result in CPU savings.

HiperDispatch will reduce the number of cache misses and optimize CP usage.

zEDC card looks like a nice way to reduce the batch window and possibly reduce CPU cache misses.
Summary

RMF CPU and HIS data can be used together to observe changes and optimize processor usage.

Improving cache misses can relate to saving processor usage = more CPs available for other work.

Note: Use smaller RMF/SMF intervals (e.g. 5 mins.) when researching.
Another way of reducing CPU usage is to improve I/O.

Quicker I/O can result in reducing the CPU cache hit count. Work can be dispatched quicker back onto a CPU that has a favorable memory access.

How can we make I/O quicker?
- Faster (newer) Disk and FICON technology
- Newer technologies (e.g. Flash enclosures)
- Application changes to use Coupling Facility
- Larger I/O buffers
Appendix
# Documentation

<table>
<thead>
<tr>
<th></th>
<th>IBM</th>
<th>RMF Report Analysis</th>
<th>SC34-2665</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>IBM</td>
<td>MVS System Management Facility (SMF)</td>
<td>SA38-0667</td>
</tr>
<tr>
<td>3</td>
<td>IBM</td>
<td>The Load-Program-Parameter and the CPU-Measurement Facilities</td>
<td>SA23-2260</td>
</tr>
<tr>
<td>4</td>
<td>IBM</td>
<td>The CPU-Measurement Facility Extended Counters Definition for z10, z196/z114, zEC12/zBC12, and z13</td>
<td>SA23-2261</td>
</tr>
<tr>
<td>5</td>
<td>IBM</td>
<td>Hardware Instrumentation Services, SMF record type 113 and related CPU Measurement Facility issues</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>IBM</td>
<td>2015 CPU MF Update</td>
<td>SHARE Seattle 2015</td>
</tr>
<tr>
<td>7</td>
<td>IntelliMagic</td>
<td>Reporting on the processor hardware with SMF 113 records</td>
<td></td>
</tr>
</tbody>
</table>
Thank You

www.intellimagic.com