

# Identifying “Store Into Instruction Stream” (SIIS) Inefficiency by Using CPU MF Counters

November 2019

John Burg  
jpburg@us.ibm.com

IBM Corporation

## Background and SIIS Description

The purpose of this TecDocs paper is to describe a new technique to identify Store Into Instruction Stream (SIIS) inefficiency by using CPU MF Counters may be occurring on your processor.

CPU MF has been available on all IBM processors starting with the z10. It provides instrumentation data on an ongoing basis for production LPARs, written to SMF. The information contains metrics including cycles, instructions, and where data and instructions are sourced within the processor cache hierarchy. Using this information, insights can be inferred into when SIIS is likely to be occurring. This TechDocs introduces a new SIIS Indicator metric and guidelines for what actions are recommended when certain thresholds are reached. For more information on CPU MF, what it is and how to enable it, see the following link: <http://www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/TC000066>

Modern processors require a codependence between their design and the code it expects to execute, including the following characteristics: Separating data and instructions, localizing storage references, and no self-modifying code. Today’s cache lines on processors are 256 bytes so it is important to ensure that data and instructions are not localized on the same cache line. Modern compilers have been written with microprocessor architecture in mind. “Old” (usually Assembler) programs with poor program practices (SIIS) continue to run. Updating these “SIIS” programs can result in significant CPU reductions.

## CPU MF Counters used for SIIS Indicator %

CPU MF can be used to help identify the potential SIIS timeframes. It is based on particular types of Instruction and Data sourcing requests (with Intervention) relative to the total misses from the L1 Cache. This is not an exact measurement, and its normal to have these types of sourcing requests, but experience has shown when the percentage of request gets to a certain threshold, it is likely to be SIIS driving the activity. Since CPU MF is at a system (LPAR) level, this only defines when it may be occurring, not who

(which job / program) is causing it. Once a time period is identified, then one must find the programs running in the time period, (e.g. with zBNA Top Programs). Once the program candidates are identified one can use a “Hot Spot” analyzer to identify the issue, or perhaps review the source code. However, the program must be remediated to eliminate the SIIS issue. The accompanying slides provide the SIIS Indicator % metric, and the formula (by processor). In all cases they formulas are derived by the CPU MF Basic and Extended Counters.

## **SIIS Indicator % Thresholds and Recommended Actions**

The accompanying slides show the SIIS Description, SIIS Indicator % Thresholds, and the Recommended Action. If the SIIS Indicator % is:

- less than 2%, then this is most likely “noise” and no action is required.
- between 2% to <5%, there may be potential MSU savings but likely minimal SIIS Impact.
- between 5% to <10%, there is likely noteworthy SIIS impact, and this should be investigated and remediated.
- 10% or higher, this should be immediately investigated and remediated, asd there is likely significant impact.

There are 2 customer examples provided in the accompanying slides. In the 1<sup>st</sup> customer example, one can see that there is significant SIIS Indicator % especially before 3 AM on multiple days. The recommendation would be to look at the jobs / programs running in the peaks and investigate via the source code or “Hot Spot” analyzer if there are programming practices that could cause SIIS activity. In Customer Example 2, there is 1 day from when the code with SIIS inefficiency was running and another day about a month later with the code was remediated from SIIS inefficiency. The customer reported that with this remediation they saved approximately 3000 seconds of CPU from the top 15 jobs.

## **Summary**

In summary, there are now recommended SIIS Indicator % formulas to identify when potential SIIS inefficiency may be occurring. The customer must determine the specific program, what needs to be changed to remediate the program, which can result in lower CPU and elapsed time. In today’s environment to save CPU for efficiency, especially with new pricing models such as Tailored Fit Pricing as all CPU time and MIPS count, using the SIIS Indicator % may be an effective way to identify potential CPU inefficiency.

As a reference there was a 2006 TechDocs “IBM System z and eserver zSeries Processor Performance Design Considerations” that provides additional background.

<http://www-03.ibm.com/support/techdocs/atmastr.nsf/WebIndex/FLASH10208>