System z: SPKA Instruction Execution

System z processor development has identified an aspect of the z196 processor design where a specific instruction generates a higher Cycles Per Instruction (CPI) than the equivalent instruction on a z9 or z10 processor. When an SPKA (Set PSW Key from Address) instruction is executed in problem state, the new out-of-order design of the z196 processor requires more pipeline stalls (i.e. serialization) to give functionally correct results than in prior generations of processors. As a result of this difference in design, code paths using this instruction will likely show up with higher relative utilization compared to other portions of the applications when comparing z196 performance to z9/z10 performance.

One of the more common uses of problem state SPKA instruction is found in a CICS environment due to the implementation of the CICS storage protection facility to improve availability. This change in SPKA behavior is not large enough to offset the overall benefits the z196 provides for the CICS environment. IBM has run performance comparisons between the z10 and z196 for a CICS web service workload and overall the resulting benefits are still in the overall published performance range. The net of this is there is no problem since System z processor performance has always been stated in terms of workloads and not specific instructions, or code segments. This phenomenon reflects the design choices of the z196 processor and gives insight into the mechanisms by which the z196 designers were able to improve instruction execution beyond the points achievable by an increase in processor frequency alone.

The overall effect in programs with a large concentration of SPKAs instructions in problem state is a code offset analysis done as part of a hot spot analysis can show the z196 spends more time executing the SPKA instruction. Some vendor performance tools or single instruction benchmarks may also be able to detect additional time spent either on the SPKA instruction or instructions immediately following the SPKA instruction.

The apparent speed increase for any given instruction will reflect the degree to which its execution can benefit from the various design choices by which the z196 has improved instruction processing. The SPKA (Set PSW Key from Address) and STOSM (Store Then Or System Mask) are examples of instructions which are inherently not capable of having their instruction execution overlapped relative to other instructions. The performance impact on the SPKA instruction follows from the requirement for serialization relative to subsequent instructions. Subsequent instruction operations must not be executed until the storage protect key for their operation has been established.

Execution of any instruction with a similar requirement for serialized processing would have the same effect, these instructions cut down on the benefit of multiple, concurrent and overlapped instruction execution. Each time one of these instructions is encountered, its execution must be delayed until the execution of all in-flight instructions has completed. Consequently, the relative execution speed of instructions requiring serialization is reduced.

It is an unavoidable consequence of the execution of instructions requiring enhanced serialization for the following instructions (after the SPKA, for example) to also be singled out by performance monitoring tools more frequently as being in process. Following instructions are highlighted because it takes some time for all of the parallelism in instruction handling/execution to effectively come on-line and benefit the execution of the instructions which follow the serialized instructions. For example, neither subsequent instructions nor their operands can be fetched until the applicable storage protect key has been established. The fetching and execution of several instructions is required to fully realize the benefits of multiple instruction decoding and operation units and overlapped instruction execution so the maximum effective parallel instruction processing can be achieved.

For example, running one of the System z performance benchmarks, it was possible to reproduce similar behavior to what was observed in hot spot analysis data from customers.

Comparing CICS data from z10 vs z196:

DFHAIP DFHEIP - there is a growth with SPKA on z196 at different offsets, similar to what was seen at several customers

z10 offset	z10 Instruction	% of Time	z196 offset	z196 Instruction	% of Time	Comments
336	SPKA	2.49	336	SPKA	4.97	
33A	ST	1.18	33A	ST	9.61	This is related to the SPKA
1820	SPKA	1.82	1820	SPKA	4.77	
1824	LM	3.92	1824	LM	10.28	This is related to the SPKA

Summary

To summarize, this change in SPKA behavior is not large enough to offset the overall benefits the z196 provides for the CICS environment. IBM has run performance comparisons between the z10 and z196 for a CICS web service workload, and overall the net benefits are still in the overall published performance range. The impact of the z196 design changes will impact the reporting by hot spot analysis tools but overall there is no problem. This phenomenon reflects the design choices of the z196 processor, and highlights one of the mechanisms by which the z196 designers were able to improve instruction execution beyond the points achievable by an increase in processor frequency alone.

IBM Corporation
New Orchard Rd.
Armonk, NY 10504
U.S.A

Produced in the United States of America
08/12

Information concerning non-IBM products was obtained from the suppliers of their products or their published announcements. Questions on the capabilities of the non-IBM products should be addressed with the suppliers.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM’s future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.