November 2020

CALIBRATION OF TIME STAMP COUNTER VALUE IN STALL FUNCTION OF PRE-OS ENVIRONMENT

HP INC

Calibration of Time Stamp Counter Value in Stall Function of Pre-OS Environment

Abstract: An incorrect time stamp value upon return from execution of the Stall function, due to an unexpectedly long handling time of an interrupt that occurs during the function, is corrected using a wrapper for the Stall function.

This disclosure relates to the field of computers, and more specifically to a BIOS.

A technique is disclosed that ensures the time stamp counter value has a reasonable value even if the servicing of an interrupt that occurs during execution of the Stall function takes longer than the Stall time.

The BIOS of a computer obtains fundamental performance information in the early stage of the Pre-OS environment. During each UEFI Stall function, the BIOS fine tunes the time stamp counter register value if needed before returning from the stall function, based on the performance data.

A UEFI application relies on the Stall function to wait for a certain period of time. According to the UEFI Spec definition, the Stall function ensures the minimum waiting time, but does not guarantee the upper bound of waiting time of stall function execution.

Some applications, such as the OS boot manager, use the Stall function and the time stamp counter in order to calculate the CPU frequency. While in the Stall function, an interrupt may come in, and the CPU may service it for a long time, even longer than the stall time. In such a case, the time stamp counter can run to an unexpectedly large value. The abnormal increase in time stamp count prevents the boot manager from obtaining the correct CPU performance data.

According to the present disclosure, an algorithm of a wrapper for the stall function corrects the time stamp count in the event that an interrupt is received during a stall function and is serviced for longer than the stall time:

1) The performance data is read in an early stage.
2) The wrapped Stall function is invoked by a caller.
3) The time stamp counter is read immediately.
4) The expected time stamp counter range after the stall function completes is calculated.
5) The wrapper invokes the original Stall function.
6) When the original Stall function returns, the time stamp counter is again read, and compared with expected range calculated at 4).
7) If the time stamp counter is out of the expected range, the time stamp counter is overridden with the correct value. The correct value may be the maximum value of the counter range.
8) The wrapped Stall function returns to the caller.

The disclosed technique advantageously ensures that the time stamp counter value will have a reasonable value after returning from the Stall function, even if a long interrupt service occurs during execution of the stall function.

Disclosed by Daniel Wang, HPInc.