



index : kernel/git/stable/linux.git

master

switch

Linux kernel stable tree

Stable Group

[about](#) [summary](#) [refs](#) [log](#) [tree](#) [commit](#) [diff](#) [stats](#)

log msg

search

author Breno Leitao <leitao@debian.org> 2023-03-21 04:33:38 -0700
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2023-03-30 12:51:24 +0200
commit [9d4c7b1f12e101d6d6253092588b127416ddfb6c](#) (patch)
tree [07d6df23ed8cbc043cb011b0758910fd9c6213ec](#)
parent [c90dc514c2e2be786159c8c2a08111fad82b3482](#) (diff)
download [linux-9d4c7b1f12e101d6d6253092588b127416ddfb6c.tar.gz](#)

diff options

context:
space:
mode:

perf/x86/amd/core: Always clear status for idx

[Upstream commit 263f5ecaf7080513efc248ec739b6d9e00f4129f]

The variable 'status' (which contains the unhandled overflow bits) is not being properly masked in some cases, displaying the following warning:

```
WARNING: CPU: 156 PID: 475601 at arch/x86/events/amd/core.c:972 amd_pmu_v2_handle_irq+0x216/0x270
```

This seems to be happening because the loop is being continued before the status bit being unset, in case x86_perf_event_set_period() returns 0. This is also causing an inconsistency because the "handled" counter is incremented, but the status bit is not cleaned.

Move the bit cleaning together above, together when the "handled" counter is incremented.

Fixes: 7685665c390d ("perf/x86/amd/core: Add PerfMonV2 overflow handling")
Signed-off-by: Breno Leitao <leitao@debian.org>
Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
Reviewed-by: Sandipan Das <sandipan.das@amd.com>
Link: <https://lore.kernel.org/r/20230321113338.1669660-1-leitao@debian.org>
Signed-off-by: Sasha Levin <sasha@kernel.org>

Diffstat

```
-rw-r--r-- arch/x86/events/amd/core.c 3
```

1 files changed, 1 insertions, 2 deletions

diff --git a/arch/x86/events/amd/core.c b/arch/x86/events/amd/core.c

index 4386b10682ce4f..8ca5e827f30b2c 100644

--- a/arch/x86/events/amd/core.c

+++ b/arch/x86/events/amd/core.c

@@ -923,6 +923,7 @@ static int amd_pmu_v2_handle_irq(struct pt_regs *regs)

/* Event overflow */

handled++;

+ status &= ~mask;

perf_sample_data_init(&data, 0, hwc->last_period);

if (!x86_perf_event_set_period(event))

@@ -935,8 +936,6 @@ static int amd_pmu_v2_handle_irq(struct pt_regs *regs)

if (perf_event_overflow(event, &data, regs))

```
- x86_pmu_stop(event, 0);  
-  
- status &= ~mask;  
-  
}  
  
/*
```