

author Tiezhu Yang <yangtiezhu@loongson.cn> 2023-02-25 15:52:56 +0800
committer Huacai Chen <chenhuacai@loongson.cn> 2023-02-25 22:12:16 +0800
commit bb7a78e343468873bf00b2b181fcfd3c02d8cb56 (patch)
tree 76350dd5fda6a12ee77d014d79d6b0bc5ad3b1e5
parent fd200632d09dcea9fd67714241155b7635505cd6 (diff)
download linux-bb7a78e343468873bf00b2b181fcfd3c02d8cb56.tar.gz

diff options

context: 3

space: include

mode: unified

LoongArch: Only call get_timer_irq() once in constant_clockevent_init()

Under CONFIG_DEBUG_ATOMIC_SLEEP=y and CONFIG_DEBUG_PREEMPT=y, we can see the following messages on LoongArch, this is because using might_sleep() in preemption disable context.

```
[ 0.001127] smp: Bringing up secondary CPUs ...
[ 0.001222] Booting CPU#1...
[ 0.001244] 64-bit Loongson Processor probed (LA464 Core)
[ 0.001247] CPU1 revision is: 0014c012 (Loongson-64bit)
[ 0.001250] FPU1 revision is: 00000000
[ 0.001252] BUG: sleeping function called from invalid context at kernel/locking/mutex.c:283
[ 0.001255] in_atomic(): 1, irqs_disabled(): 1, non_block: 0, pid: 0, name: swapper/1
[ 0.001257] preempt_count: 1, expected: 0
[ 0.001258] RCU nest depth: 0, expected: 0
[ 0.001259] Preemption disabled at:
[ 0.001261] [<9000000000223800>] arch_dup_task_struct+0x20/0x110
[ 0.001272] CPU: 1 PID: 0 Comm: swapper/1 Not tainted 6.2.0-rc7+ #43
[ 0.001275] Hardware name: Loongson Loongson-3A5000-7A1000-1w-A2101/Loongson-LS3A5000-7A1000-1w-A2101, BIOS vUDK2018-LoongArch-V4.0.05132-beta10
[ 0.001277] Stack : 0072617764726148 0000000000000000 9000000000222f1c 90000001001e0000
[ 0.001286]          90000001001e3be0 90000001001e3be8 0000000000000000 0000000000000000
[ 0.001292]          90000001001e3be8 0000000000000040 90000001001e3cb8 90000001001e3a50
[ 0.001297]          9000000001642000 90000001001e3be8 be694d10ce4139dd 9000000100174500
[ 0.001303]          0000000000000001 0000000000000001 00000000ffffe0a2 0000000000000020
[ 0.001309]          000000000000002f 9000000001354116 000000000056b000 ffffffff
[ 0.001314]          0000000000000000 0000000000000000 90000000014f6e90 9000000001642000
[ 0.001320]          900000000022b69c 0000000000000001 0000000000000000 9000000001736a90
[ 0.001325]          9000000100038000 0000000000000000 9000000000222f34 0000000000000000
[ 0.001331]          00000000000000b0 0000000000000004 0000000000000000 0000000000070000
[ 0.001337]          ...
[ 0.001339] Call Trace:
[ 0.001342] [<9000000000222f34>] show_stack+0x5c/0x180
[ 0.001346] [<90000000010bdd80>] dump_stack_lvl+0x60/0x88
[ 0.001352] [<9000000000266418>] __might_resched+0x180/0x1cc
[ 0.001356] [<90000000010c742c>] mutex_lock+0x20/0x64
[ 0.001359] [<90000000002a8ccc>] irq_find_matching_fwspec+0x48/0x124
[ 0.001364] [<90000000002259c4>] constant_clockevent_init+0x68/0x204
[ 0.001368] [<900000000022acf4>] start_secondary+0x40/0xa8
[ 0.001371] [<90000000010c0124>] smpboot_entry+0x60/0x64
```

Here are the complete call chains:

```
smpboot_entry()
  start_secondary()
    constant_clockevent_init()
      get_timer_irq()
        irq_find_matching_fwnode()
          irq_find_matching_fwspec()
            mutex_lock()
              might_sleep()
                __might_sleep()
                  __might_resched()
```

In order to avoid the above issue, we should break the call chains, using timer_irq_installed variable as check condition to only call get_timer_irq() once in constant_clockevent_init() is a simple and proper way.

Signed-off-by: Tiezhu Yang <yangtiezhu@loongson.cn>
Signed-off-by: Huacai Chen <chenhuacai@loongson.cn>

Diffstat

-rw-r--r-- arch/loongarch/kernel/time.c 11

1 files changed, 6 insertions, 5 deletions

```
diff --git a/arch/loongarch/kernel/time.c b/arch/loongarch/kernel/time.c
index a6576dea590c0b..4351f69d995015 100644
--- a/arch/loongarch/kernel/time.c
+++ b/arch/loongarch/kernel/time.c
@@ -140,16 +140,17 @@ static int get_timer_irq(void)

int constant_clockevent_init(void)
{
-    int irq;
```

```
unsigned int cpu = smp_processor_id();
unsigned long min_delta = 0x600;
unsigned long max_delta = (1UL << 48) - 1;
struct clock_event_device *cd;
- static int timer_irq_installed = 0;
+ static int irq = 0, timer_irq_installed = 0;

- irq = get_timer_irq();
- if (irq < 0)
-     pr_err("Failed to map irq %d (timer)\n", irq);
+ if (!timer_irq_installed) {
+     irq = get_timer_irq();
+     if (irq < 0)
+         pr_err("Failed to map irq %d (timer)\n", irq);
+ }

cd = &per_cpu(constant_clokevent_device, cpu);
```