author	Sudeep Holla <sudeep.holla@arm.com>	2023-03-08 11:26:32 +0000
committer	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2023-03-22 13:34:04 +0100
commit	1318a07706bb2f8c65f88f39a16c2b5260bcdcd4 (patch)	
tree	5507e0b30d49f3c87ab9543282e58242c234f3a6	
parent	ac1d15d58d8ac2a934d6244f4fedb6f5590c2be2 (diff)	
download	linux-1318a07706bb2f8c65f88f39a16c2b5260bcdcd4.tar.gz	

diff options

context: 3

space: include

mode: unified

ACPI: PPTT: Fix to avoid sleep in the atomic context when PPTT is absent

commit 91d7b60a65d9f71230ea09b86d2058a884a3c2af upstream.

Commit 0c80f9e165f8 ("ACPI: PPTT: Leave the table mapped for the runtime usage")
enabled to map PPTT once on the first invocation of acpi_get_pptt() and
never unmapped the same allowing it to be used at runtime with out the
hassle of mapping and unmapping the table. This was needed to fetch LLC
information from the PPTT in the cpuhotplug path which is executed in
the atomic context as the acpi_get_table() might sleep waiting for a
mutex.

However it missed to handle the case when there is no PPTT on the system
which results in acpi_get_pptt() being called from all the secondary
CPUs attempting to fetch the LLC information in the atomic context
without knowing the absence of PPTT resulting in the splat like below:

```
| BUG: sleeping function called from invalid context at kernel/locking/semaphore.c:164
| in_atomic(): 1, irqs_disabled(): 1, non_block: 0, pid: 0, name: swapper/1
| preempt_count: 1, expected: 0
| RCU nest depth: 0, expected: 0
| no locks held by swapper/1/0.
| irq event stamp: 0
| hardirqs last  enabled at (0): 0x0
| hardirqs last disabled at (0): copy_process+0x61c/0x1b40
| softirqs last  enabled at (0): copy_process+0x61c/0x1b40
| softirqs last disabled at (0): 0x0
| CPU: 1 PID: 0 Comm: swapper/1 Not tainted 6.3.0-rc1 #1
| Call trace:
|  dump_backtrace+0xac/0x138
|  show_stack+0x30/0x48
|  dump_stack_lvl+0x60/0xb0
|  dump_stack+0x18/0x28
|  __might_resched+0x160/0x270
|  __might_sleep+0x58/0xb0
|  down_timeout+0x34/0x98
|  acpi_os_wait_semaphore+0x7c/0xc0
|  acpi_ut_acquire_mutex+0x58/0x108
|  acpi_get_table+0x40/0xe8
|  acpi_get_pptt+0x48/0xa0
|  acpi_get_cache_info+0x38/0x140
|  init_cache_level+0xf4/0x118
|  detect_cache_attributes+0x2e4/0x640
```

```
 |  update_siblings_masks+0x3c/0x330
 |  store_cpu_topology+0x88/0xf0
 |  secondary_start_kernel+0xd0/0x168
 |  __secondary_switched+0xb8/0xc0
```

Update acpi_get_pptt() to consider the fact that PPTT is once checked and
is not available on the system and return NULL avoiding any attempts to
fetch PPTT and thereby avoiding any possible sleep waiting for a mutex
in the atomic context.

Fixes: 0c80f9e165f8 ("ACPI: PPTT: Leave the table mapped for the runtime usage")
Reported-by: Aishwarya TCV <aishwarya.tcv@arm.com>
Signed-off-by: Sudeep Holla <sudeep.holla@arm.com>
Tested-by: Pierre Gondois <pierre.gondois@arm.com>
Cc: 6.0+ <stable@vger.kernel.org> # 6.0+
Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

Diffstat

-rw-r--r-- drivers/acpi/pptt.c 5

1 files changed, 4 insertions, 1 deletions

```
diff --git a/drivers/acpi/pptt.c b/drivers/acpi/pptt.c
index c91342dcbcd636..ced3eb15bd8b73 100644
--- a/drivers/acpi/pptt.c
+++ b/drivers/acpi/pptt.c
@@ -537,16 +537,19 @@ static int topology_get_acpi_cpu_tag(struct acpi_table_header *table,
 static struct acpi_table_header *acpi_get_pptt(void)
 {
        static struct acpi_table_header *pptt;
+       static bool is_pptt_checked;
        acpi_status status;

        /*
         * PPTT will be used at runtime on every CPU hotplug in path, so we
         * don't need to call acpi_put_table() to release the table mapping.
         */
-       if (!pptt) {
+       if (!pptt && !is_pptt_checked) {
                status = acpi_get_table(ACPI_SIG_PPTT, 0, &pptt);
                if (ACPI_FAILURE(status))
                        acpi_pptt_warn_missing();
+
+               is_pptt_checked = true;
        }

        return pptt;
```