

[KVM][PATCH] smp_processor_id() and sleeping functions used in invalid context

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-12/msg08152.html>

- *From:* Luca Tettamanti <kronos.it@xxxxxxxxxx>
 - *Date:* Sun, 31 Dec 2006 18:01:47 +0100
-

Hello,

I'm testing KVM on a Core2 CPU. I'm running kernel 2.6.20-git (pulled few hours ago), configured with SMP and PREEMPT.

I'm hitting 2 different warnings:

```
BUG: using smp_processor_id() in preemptible [00000001] code: kvm/7726
caller is vmx_create_vcpu+0x9/0x2f [kvm_intel]
[<b01ccbc8>] debug_smp_processor_id+0xa0/0xb4
[<f1a35b34>] vmx_create_vcpu+0x9/0x2f [kvm_intel]
[<f1b68d4f>] kvm_dev_ioctl+0x160/0xf97 [kvm]
[<b0130a7d>] autoremove_wake_function+0x0/0x35
[<b01481a9>] get_page_from_freelist+0xe9/0x353
[<b02f0c46>] _write_unlock+0x25/0x3b
[<b0148295>] get_page_from_freelist+0x1d5/0x353
[<b01364c4>] mark_held_locks+0x46/0x62
[<b0148295>] get_page_from_freelist+0x1d5/0x353
[<b0148295>] get_page_from_freelist+0x1d5/0x353
[<b01366a5>] trace_hardirqs_on+0x11e/0x141
[<b014f189>] __handle_mm_fault+0x463/0x864
[<b014f189>] __handle_mm_fault+0x463/0x864
[<b02f0c0b>] _spin_unlock+0x25/0x3b
[<b014f55f>] __handle_mm_fault+0x839/0x864
[<b0117398>] do_page_fault+0x15a/0x52a
[<f1b68bef>] kvm_dev_ioctl+0x0/0xf97 [kvm]
[<b0168ed7>] do_ioctl+0x1f/0x62
[<b016915e>] vfs_ioctl+0x244/0x256
[<b01366a5>] trace_hardirqs_on+0x11e/0x141
[<b01691a3>] sys_ioctl+0x33/0x4c
[<b0102f10>] syscall_call+0x7/0xb
```

=====

vmx_create_vcpu calls alloc_vmcs which uses smp_processor_id() in preemptible context and pass the result to alloc_vmcs_cpu(); at a later point the function may be running on a different CPU (hence the result of cpu_to_node may be meaningless).

Second one:

BUG: sleeping function called from invalid context at

[KVM][PATCH] smp_processor_id() and sleeping functions used in invalid context

```

/home/kronos/src/linux-2.6.git/mm/slab.c:3034
in_atomic():1, irqs_disabled():0
1 lock held by kvm/12706:
#0: (&vcpu->mutex){---}, at: [<f1b68d02>] kvm_dev_ioctl+0x113/0xf97
[kvm]
[<b015c32a>] kmem_cache_alloc+0x1b/0x6f
[<f1a360ab>] vmx_vcpu_setup+0x528/0x703 [kvm_intel]
[<f1a35a7d>] vmx_vcpu_load+0xdb/0xe3 [kvm_intel]
[<f1b68d8c>] kvm_dev_ioctl+0x19d/0xf97 [kvm]
[<b0130a7d>] autoremove_wake_function+0x0/0x35
[<b02d2ce3>] ip4_datagram_connect+0x293/0x2ec
[<b02f0c46>] _write_unlock+0x25/0x3b
[<b0148295>] get_page_from_freelist+0x1d5/0x353
[<b01364c4>] mark_held_locks+0x46/0x62
[<b0148295>] get_page_from_freelist+0x1d5/0x353
[<b0148295>] get_page_from_freelist+0x1d5/0x353
[<b01366a5>] trace_hardirqs_on+0x11e/0x141
[<b0102235>] setup_sigcontext+0x105/0x189
[<b014f189>] __handle_mm_fault+0x463/0x864
[<b014f189>] __handle_mm_fault+0x463/0x864
[<b02f0c0b>] _spin_unlock+0x25/0x3b
[<b014f55f>] __handle_mm_fault+0x839/0x864
[<b0117398>] do_page_fault+0x15a/0x52a
[<f1b68bef>] kvm_dev_ioctl+0x0/0xf97 [kvm]
[<b0168ed7>] do_ioctl+0x1f/0x62
[<b016915e>] vfs_ioctl+0x244/0x256
[<b01366a5>] trace_hardirqs_on+0x11e/0x141
[<b01691a3>] sys_ioctl+0x33/0x4c
[<b0102f10>] syscall_call+0x7/0xb
=====

```

I think it's caused the GFP_KERNEL allocation (vmx_vcpu_setup, vmx.c:1103) done inside get_cpu().

This first warning can be fixed by moving the cpu_to_node conversion. Not sure that it's correct though. Between vmx_create_vcpu() and vmx_vcpu_load() the CPU may change and the 'vmcs' may be actually on a different node, no?

Singed-Off-By: Luca Tettamanti <kronos.it@xxxxxxxxxx>

```

---
drivers/kvm/vmx.c | 11 ++++++----
1 file changed, 8 insertions(+), 3 deletions(-)

diff --git a/drivers/kvm/vmx.c b/drivers/kvm/vmx.c
index d0a2c2d..4d088a0 100644
--- a/drivers/kvm/vmx.c
+++ b/drivers/kvm/vmx.c
@@ -524,9 +524,8 @@ static __init void setup_vmcs_descriptor(void)

```

[KVM][PATCH] smp_processor_id() and sleeping functions used in invalid context

```
vmcs_descriptor.revision_id = vmx_msr_low;
}

-static struct vmcs *alloc_vmcs_cpu(int cpu)
+static struct vmcs *alloc_vmcs_cpu(int node)
{
- int node = cpu_to_node(cpu);
struct page *pages;
struct vmcs *vmcs;

@@ -541,7 +540,13 @@ static struct vmcs *alloc_vmcs_cpu(int cpu)

static struct vmcs *alloc_vmcs(void)
{
- return alloc_vmcs_cpu(smp_processor_id());
+ int cpu, node;
+
+ cpu = get_cpu();
+ node = cpu_to_node(cpu);
+ put_cpu();
+
+ return alloc_vmcs_cpu(node);
}

static void free_vmcs(struct vmcs *vmcs)
```

Luca

--

"L'amore consiste nell'essere cretini insieme." -- P. Valery

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>