Re: [PATCH/RFC] KVM: track pid for VCPU only on KVM_RUN ioctl

[Date Prev][Date Next][Thread Prev][Thread Next][Date Index][Thread Index]

- *Subject*: Re: [PATCH/RFC] KVM: track pid for VCPU only on KVM_RUN ioctl
- *From*: Paolo Bonzini <pbonzini@xxxxxxxxxx>
- *Date*: Wed, 03 Dec 2014 14:20:34 +0100
- *Cc*: KVM <kvm@xxxxxxxxxxxxxxx>, Gleb Natapov <gleb@xxxxxxxxxx>, Rik van Riel <riel@xxxxxxxxxx>, Raghavendra K T <raghavendra.kt@xxxxxxxxxxxxxxxxxx>, Michael Mueller <mimu@xxxxxxxxxxxxxxxxxx>, David Hildenbrand <dahi@xxxxxxxxxxxxxxxxxx>
- *In-reply-to*: <1407249854-2953-1-git-send-email-borntraeger@de.ibm.com>
- *References*: <1407249854-2953-1-git-send-email-borntraeger@de.ibm.com>
- *User-agent*: Mozilla/5.0 (X11; Linux x86_64; rv:31.0) Gecko/20100101 Thunderbird/31.2.0

3

On 05/08/2014 16:44, Christian Borntraeger wrote:
> We currently track the pid of the task that runs the VCPU in
> vcpu_load. Since we call vcpu_load for all kind of ioctls on a
> CPU, this causes hickups due to synchronize_rcu if one CPU is
> modified by another CPU or the main thread (e.g. initialization,
> reset). We track the pid only for the purpose of yielding, so
> let's update the pid only in the KVM_RUN ioctl.
>
> In addition, don't do a synchronize_rcu on startup (pid == 0).
>
> This speeds up guest boot time on s390 noticably for some configs, e.g.
> HZ=100, no full state tracking, 64 guest cpus 32 host cpus.
>
> Signed-off-by: Christian Borntraeger <borntraeger@xxxxxxxxxx>
> CC: Rik van Riel <riel@xxxxxxxxxx>
> CC: Raghavendra K T <raghavendra.kt@xxxxxxxxxxxxxxxxxx>
> CC: Michael Mueller <mimu@xxxxxxxxxxxxxxxxxx>
> ---
> virt/kvm/kvm_main.c | 17 +++++++++--------
> 1 file changed, 9 insertions(+), 8 deletions(-)
>
> diff --git a/virt/kvm/kvm_main.c b/virt/kvm/kvm_main.c
> index 9ae9135..ebc8f54 100644
> --- a/virt/kvm/kvm_main.c

```
> +++ b/virt/kvm/kvm_main.c
> @@ -124,14 +124,6 @@ int vcpu_load(struct kvm_vcpu *vcpu)
>
>       if (mutex_lock_killable(&vcpu->mutex))
>               return -EINTR;
> -     if (unlikely(vcpu->pid != current->pids[PIDTYPE_PID].pid)) {
> -             /* The thread running this VCPU changed. */
> -             struct pid *oldpid = vcpu->pid;
> -             struct pid *newpid = get_task_pid(current, PIDTYPE_PID);
> -             rcu_assign_pointer(vcpu->pid, newpid);
> -             synchronize_rcu();
> -             put_pid(oldpid);
> -     }
>       cpu = get_cpu();
>       preempt_notifier_register(&vcpu->preempt_notifier);
>       kvm_arch_vcpu_load(vcpu, cpu);
> @@ -1991,6 +1983,15 @@ static long kvm_vcpu_ioctl(struct file *filp,
>               r = -EINVAL;
>               if (arg)
>                       goto out;
> +             if (unlikely(vcpu->pid != current->pids[PIDTYPE_PID].pid)) {
> +                     /* The thread running this VCPU changed. */
> +                     struct pid *oldpid = vcpu->pid;
> +                     struct pid *newpid = get_task_pid(current, PIDTYPE_PID);
> +                     rcu_assign_pointer(vcpu->pid, newpid);
> +                     if (oldpid)
> +                             synchronize_rcu();
> +                     put_pid(oldpid);
> +             }
>               r = kvm_arch_vcpu_ioctl_run(vcpu, vcpu->run);
>               trace_kvm_userspace_exit(vcpu->run->exit_reason, r);
>               break;
>
```

Applied with rewritten commit message:

```
    KVM: track pid for VCPU only on KVM_RUN ioctl

    We currently track the pid of the task that runs the VCPU in vcpu_load.
    If a yield to that VCPU is triggered while the PID of the wrong thread
    is active, the wrong thread might receive a yield, but this will most
    likely not help the executing thread at all.  Instead, if we only track
    the pid on the KVM_RUN ioctl, there are two possibilities:

    1) the thread that did a non-KVM_RUN ioctl is holding a mutex that
    the VCPU thread is waiting for.  In this case, the VCPU thread is not
    runnable, but we also do not do a wrong yield.

    2) the thread that did a non-KVM_RUN ioctl is sleeping, or doing
    something that does not block the VCPU thread.  In this case, the
    VCPU thread can receive the directed yield correctly.

    Signed-off-by: Christian Borntraeger <borntraeger@xxxxxxxxxx>
    CC: Rik van Riel <riel@xxxxxxxxxx>
    CC: Raghavendra K T <raghavendra.kt@xxxxxxxxxxxxxxxxxx>
    CC: Michael Mueller <mimu@xxxxxxxxxxxxxxxxx>
    Signed-off-by: Paolo Bonzini <pbonzini@xxxxxxxxxx>

Thanks,

Paolo
--
To unsubscribe from this list: send the line "unsubscribe kvm" in
the body of a message to majordomo@xxxxxxxxxxxxxxx
More majordomo info at  http://vger.kernel.org/majordomo-info.html
```

- **References**:
  - **[PATCH/RFC] KVM: track pid for VCPU only on KVM_RUN ioctl**
    - *From:* Christian Borntraeger

- Prev by Date: **Re: [PATCH RFC 0/2] assign each vcpu an owning thread and improve yielding**
- Next by Date: **[PATCH 2/2] KVM: cpuid: set CPUID(EAX=0xd,ECX=1).EBX correctly**
- Previous by thread: **Re: [PATCH/RFC] KVM: track pid for VCPU only on KVM_RUN ioctl**
- Next by thread: **[PATCH] virtio-rng: complete have_data completion in removing device**
- Index(es):
  - **Date**
  - **Thread**