Subject: Re: [PATCH 1/3] Introduce cpuid_on_cpu() and cpuid_eax_on_cpu()
Posted by Alexey Dobriyan on Tue, 03 Apr 2007 13:33:31 GMT
View Forum Message <> Reply to Message

On Mon, Apr 02, 2007 at 02:10:29PM +0200, Andi Kleen wrote:
> On Monday 02 April 2007 13:38, Alexey Dobriyan wrote:
> > They will be used by cpuid driver and powernow-k8 cpufreq driver.
> >
> > With these changes powernow-k8 driver could run correctly on OpenVZ kernels
> > with virtual cpus enabled (SCHED_VCPU).
>
> This means openvz has multiple virtual CPU levels?

Not sure what do you mean.

> One for cpuid/rdmsr and one for the rest of the kernel?

Now I think I do. No, only one VCPU level:

```
 +---+ +---+ +---+
PCPU | 0 | | 1 | | 2 |
 +---+ +---+ +---+
   | \
   |  `---,
   V       \
VCPU +---+ +---+ +---+
 | 0 | | 1 | | 2 |
 +---+ +---+ +---+
```

PCPU chooses VCPU, chooses task from it's runqueue, or becomes idle.

> Both powernow-k8 and cpuid attempt to schedule
> to the target CPU so they should already run there. But it is some other CPU,
> but when they ask your _on_cpu() functions they suddenly get a "real" CPU?
> Where is the difference between these levels of virtualness?

*_on_cpu functions do some work on given physical CPU.
set_cpus_allowed() in openvz operates on VCPU level, so process doing
set_cpus_allowed() still could be scheduled anywhere. Which horribly
breaks cpufreq drivers. To unbreak them, we need a way to do frequency
changing work on given PCPU whose number comes from /sys/*/cpu/cpu*
hierarchy.

> That sounds quite fragile and will likely break often.  I just rejected a similar
> concept -- virtual nodes and "physical nodes" for similar reasons.

Care to drop a link, so I could compare them? Don't recall something
like that posted on l-k.

> Also it has weird semantics. For example if you have multiple
> virtual CPUs mapping to a single CPU then would the powernow-k8 driver
> try to set the frequency multiple times on the same physical CPU?

If core cpufreq locking is OK, why would it?

> That might
> go wrong actually because the CPU might not be happy to be poked again
> while it is already in a frequency change. Also there is no locking
> so in theory two vcpus might try to change frequency in parallel with
> probably quite bad effects.
>
> I'm sure there are other scenarios with similar problems. e.g. what
> happens with microcode updates etc.?

apply_microcode() looks small enough to convert it to IPIs, but so far
nobody asked for microcode updates in openvz.

> Before adding any hacks like this I think your vcpu concept
> needs to be discussed properly on l-k. For me it doesn't look like it is
> something good right now though.

Andi, I think it all relies on correctness of core cpufreq locking. Core
cpufreq locking is not changed with these patches so number of bugs
should stay the same.

```
current way    new way
-----------    -------

set_cpus_allowed(cpu);
[frequency change part #1]  [IPI part #1]
   <=== nasty here ===>
[frequency change part #2]  [IPI part #2]
set_cpus_allowed(old)
```