

Re: [PATCH] i386 No-Idle-Hz aka Dynamic-Ticks 3

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-08/2596.html>

From: George Anzinger (george_at_mvista.com)

Date: 08/09/05

Date: Tue, 09 Aug 2005 13:05:12 -0700

To: Tony Lindgren <tony@atomide.com>

Tony Lindgren wrote:

> * [Srivatsa Vaddagiri](mailto:Srivatsa.Vaddagiri@in.ibm.com) <vatsa@in.ibm.com> [050805 05:37]:

>

>> On Wed, Aug 03, 2005 at 06:05:28AM +0000, Con Kolivas wrote:

>>

>>> This is the dynamic ticks patch for i386 as written by Tony Lindgen

>>> <tony@atomide.com> and [Tuukka Tikkanen](mailto:Tuukka.Tikkanen@elektrobit.com) <tuukka.tikkanen@elektrobit.com>.

>>> Patch for 2.6.13-rc5

>>>

>>> There were a couple of things that I wanted to change so here is an updated

>>> version. This code should have stabilised enough for general testing now.

>>

>> Con,

>> I have been looking at some of the requirement of tickless idle CPUs in
>> core kernel areas like scheduler and RCU. Basically, both power management and
>> virtualization benefit if idle CPUs can cut off useless timer ticks. Especially
>> from a virtualization standpoint, I think it makes sense that we enable this
>> feature on a per-CPU basis i.e let individual CPUs cut off their ticks as and
>> when they become idle. The benefit of this is more visible in platforms that
>> host lot of (SMP) VMs on the same machine. Most of the time, these VMs may be
>> partially idle (some CPUs in it are idle, some not) and it is good that we
>> quiesce the timer ticks on the partial set of idle CPUs. Both S390 and Xen ports
>> of Linux kernel have this ability today (S390 has it in mainline already and
>> Xen has it out of tree).

>

>

> Good point, and it would be nice to have it resolved for systems that support
> idling individual CPUs. The current setup was done because when I was tinkering
> with the `amd76x_pm` patch a while a back, I noticed that idling the `cpu`
> disconnects all `cpus` from the bus. (As far as I remember)

>

> So this may need to be configured depending on the system.

>

>

>> From this viewpoint, I think the current implementation of dynamic tick

>> falls short of this requirement. It cuts of the timer ticks only when

>> all CPUs go idle.

>>
>>Apart from this observation, I have some others about the current dynamic tick
>>patch:
>>
>>- All CPUs seem to cut off the same number of ticks (*dyn_tick->skip*). Isn't
>> this wrong, considering that the timer list is per-CPU? This will cause
>> some timers to be serviced much later than usual.
>
>
> Yes if it's done on per-CPU basis. In the current setup the first interrupt
> will kick the system off the *dyn-tick* state and the timers get checked again.
>
>
>>- The fact that *dyn_tick_state* is global and accessed from all CPUs
>> is probably a scalability concern, especially if we allow the ticks
>> to be cut off on per-CPU basis.
>
>
>>From idling devices point of view, we still need some global variable I
> believe. How else would you be able to tell all devices that the whole
> system does not have any timers for next 2 seconds?
>
>
>>- Again, when we allow this on a per-CPU basis, subsystems like
>> RCU need to know the partial set of idle CPUs. RCU already does
>> that thr' *nohz_cpu_mask* (which will need to replace *dyn_cpu_map*).
>
>
> Sounds like that could work for *dyn-tick* too.
>
>
>>- Looking at *dyn_tick_timer_interrupt*, would it be nice if we avoid calling
>> *do_timer_interrupt* so many times and instead update *jiffies* to
>> (*skipped_ticks - 1*) and then call *do_timer_interrupt* once? I think
>> VST does it that way.
>
>
> In the long run we would do the calculations in *usecs* and just emulate
> *jiffies* from the hw timer. But yes, optimizing updating the time would be
> great.
>
>
>>- *dyn_tick->max_skip = 0xffffffff / apic_timer_val*;
>> From my reading of Intel docs, *APIC_TMICT* is 32-bit. So why does the
>> above calculation take only 24-bits into account? What am I missing here?
>
>
> Hmm, could be a bug here, needs to be checked. Maybe 32-bit APIC timer is
> optional support, or maybe I accidentally pulled the optional 24-bit support
> from the ACPI PM timer.
>

> *But in any case on P4 systems the APIC timer is not the bottleneck as
> stopping or reprogramming PIT also kills APIC. (This does not happen on P3
> systems). So the bottleneck most likely is the length of PIT.*
>
>
>>*I can take a shot at addressing these concerns in dynamic_tick patch, but it
>>seems to me that VST has already addressed all these to a big extent. Had you
>>considered VST before? The biggest bottleneck I see in VST going mainline is
>>its dependency on HRT patch but IMO it should be possible to write a small patch
>>to support VST w/o HRT.*
>>
>>*George, what do you think?*
>
>
> *HRT + VST depend on APIC only, and does not use next_timer_interrupt().*

I convinced my self that the next_timer... code in timer.c misses timers (i.e. gives the wrong answer). I did this (after wondering due to performance) by scanning the whole timer list after I had the next_timer... answer and finding a better answer, not always, but some times. That code does not address the cascade list correctly.

--

George Anzinger george@mvista.com

HRT (High-res-timers): <http://sourceforge.net/projects/high-res-timers/>

-

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

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

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