Re: NTP problem − Clock too fast for NTP to keep up?

Source: http://linux.derkeiler.com/Mailing−Lists/Fedora/2005−02/2793.html

---

From: John DeDourek (*dedourek_at_unb.ca*)
Date: 02/09/05

```
Date: Wed, 09 Feb 2005 09:44:37 −0400
To: For users of Fedora Core releases <fedora−list@redhat.com>
```

James Wilkinson wrote:
> jdow wrote:
>
>>Also try booting with "noapic" and "nolapic" options. (I'd LOVE to
>>know why the APIC screws up the ntp operations.)
>
>
> I'm sure the kernel developers would, too.
>
> But it's not that improbable, if you think about it.
>
> The kernel measures the passage of time by counting timer ticks. A timer
> works by sending interrupts to the CPU (which are received by the
> kernel) every so often. And a Programmable Interrupt Controller (such as
> the Advanced PIC, or APIC) is responsible for marshalling those
> interrupts and sending them on.
>
> James.
>
I was browsing some of the 2.6 kernel sources recently. It seems
that there have been major changes in the timekeeping since the
last time that I looked at the sources (not sure if that was 2.2
or 2.4). I am NOT a kernel hacker, therefore read the following
with some reservation. It appears that:
−− the kernel actually runs with a much higher HZ (more clock
ticks per second, smaller tick value which is in microseconds)
than indicated by tickadj
−− the kernel "lies" to user space and says that HZ is 100 (tick
is 10000) for backward compatibility
−− the kernel does miss clock interrupts; however it "compensates"
by using another counter to detect missing ticks; there are
several to choose from, depending on the architecture of the
machine and the processor installed. Later Pentiums have
a "TSC", a cycle counter that runs at the cpu clock rate;
it is not available on early pentiums; it is also affected

by CPU power management which lowers the CPU frequency to save power when lightly loaded (on some systems). APIC apparently specifies another clock counter which is not affected by the CPU speed throtteling. (I suspect that if you have a very old system, you WILL see clock interrupt loss because you don't have one of the newer features to compensate)

−− When returning time to a user program (presumably including ntpd) the system does account for lost ticks using the timers, so lost ticks shouldn't be a problem (if the timers are present)

−− From a previous experience with Red Hat's attempt to set HZ to a high value (in the 7.x series) I know that the fixed point calculations in the time routines were sensitive to the setting of HZ. The ACTUAL tick value computed (in those kernels) was incorrect due to truncation when converting from HZ to the actual clock constants. This was compensated by ntpd essentially recognizing this as a "frequency error" which it tuned out over time. It also saved the frequency correction in the "drift" file and used that when restarting. HOWEVER, the correction was different for various values of HZ, because the computation roundoff error would be different for each HZ. This meant that the value in "drift" deduced by ntpd for one kernel HZ was wrong when starting another kernel. Then ntpd would start with the wrong frequency correction and adjust over time. Not sure how this affects the new 2.6 kernel because I haven't looked at the new caluculations.