

freebsd-current: Re: TSC instead of ACPI: powerd doesn't work anymore (to be expected?)

## Re: TSC instead of ACPI: powerd doesn't work anymore (to be expected?)

*Source:* <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/current/2005-10/1228.html>

---

**From:** Robert Watson ([rwatson\\_at\\_FreeBSD.org](mailto:rwatson_at_FreeBSD.org))

**Date:** 10/31/05

Date: Mon, 31 Oct 2005 10:53:45 +0000 (GMT)

To: David Xu <[davidxu@freebsd.org](mailto:davidxu@freebsd.org)>

On Mon, 31 Oct 2005, David Xu wrote:

> *In thread program, we have to use clock\_gettime, for example a thread*  
> *wants to wait for condition variable for two seconds, it has to:*  
>  
> *struct timespec ts;*  
>  
> *clock\_gettime(CLOCK\_REALTIME, &ts);*  
> *ts.tv\_sec += 2;*  
> *pthread\_cond\_wait(&cond, &mtx);*  
>  
> *problem is who really cares time precise? how many people are really*  
> *handling realtime critical tasks? this is really an unpleasant side*  
> *effect that a simple syscall can stall cpu.*

The questions are: how good is good enough? And for what?

The problem here is that while `pthread_cond_wait()` can afford to accept low resolution, the same clock retrieval interface is also used for things like scientific measurement. So it sounds like the real mistake was the specification for the thread API: while `timespec` is the New Way for time-related interfaces, this approach has coalesced precision time measurement and "how long to sleep for" time measurement. Moving to HZ incrementing of the clock probably exceeds the quality reduction that makes sense for `clock_gettime()`, but moving to a roughly sync'd TSC clock (such as the Linux approach) probably doesn't.

Robert N M Watson

---

`freebsd-current@freebsd.org` mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-current>

To unsubscribe, send any mail to "`freebsd-current-unsubscribe@freebsd.org`"