Subject: Re: [PATCH v4] posix timers: allocate timer id per process
Posted by Eric Dumazet on Fri, 19 Oct 2012 07:56:40 GMT
View Forum Message <> Reply to Message

On Fri, 2012-10-19 at 11:50 +0400, Stanislav Kinsbursky wrote:
> v4:
> 1) a couple of coding style fixes (lines over 80 characters)
>
> v3:
> 1) hash calculation simlified to improve perfomance.
>
> v2:
> 1) Hash table become RCU-friendly. Hash table search now done under RCU lock
> protection.

This should not be in the changelog, only after the --- separator.

> I've tested scalability on KVM with 4 CPU. The testing environment was build
> of 10 processes, each had 512 posix timers running (SIGSEV_NONE) and was
> calling timer_gettime() in loop. With all this stuff being running, I was
> measuring time of calling of syscall timer_gettime() 10000 times.
>
> Without this patch: ~7ms
> With this patch : ~7ms
>
> This patch is required CRIU project (www.criu.org).
> To migrate processes with posix timers we have to make sure, that we can
> restore posix timer with proper id.
> Currently, this is not true, because timer ids are allocated globally.
> So, this is precursor patch and it's purpose is make posix timer id to be
> allocated per process.
>
> Patch replaces global idr with global hash table for posix timers and
> makes timer ids unique not globally, but per task. Next free timer id is type
> of integer and stored on signal struct (posix_timer_id). If free timer id
> reaches negative value on timer creation, it will be dropped to zero and
> -EAGAIN will be returned to user.

I wonder if some applications relied on our idr, assuming they would get
low values for their timer id.
(We could imagine some applications use a table indexed by the timer id)

> Hash table is size of page (4KB).

Only on x86_64. Why not instead saying hashtable has 512 slots ?

> Key is constructed as follows:
> key = hash_32(current->signal) ^ hash_32(posix_timer_id);

This is outdated.

>
> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
> ---

Thanks