

## Re: [RFC][PATCH 1/7] Resource counters

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-03/msg06441.html>

---

- *From:* [ebiederm@xxxxxxxxxxxxx](mailto:ebiederm@xxxxxxxxxxxxx) (Eric W. Biederman)
  - *Date:* Thu, 15 Mar 2007 10:51:46 -0600
- 

Pavel Emelianov <xemul@xxxxx> writes:

Srivatsa Vaddagiri wrote:

On Tue, Mar 13, 2007 at 06:41:05PM +0300, Pavel Emelianov wrote:

right, but atomic ops have much less impact  
on most  
architectures than locks :)

Right. But `atomic_add_unless()` is slower as it is  
essentially a loop. See my previous letter in this sub-thread.

If I am not mistaken, you shouldn't loop in normal cases, which means  
it boils down to a `atomic_read() + atomic_cmpxch()`

So does the lock – in a normal case (when it's not  
heavily contented) it will boil down to `atomic_dec_and_test()`.

Nevertheless, making charge like in this patchset  
requires two atomic ops with `atomic_xxx` and only  
one with `spin_lock()`.

To be very clear. If you care about optimization cache lines  
and lock hold times (to keep contention down) are the important  
things.

With spin locks you have to be a little more careful to put them  
on the same cache line as your data and to keep should hold times  
short. With atomic ops you get that automatically.

There is really no significant advantage in either approach.  
The number of atomic ops doesn't matter. You bring in

Re: [RFC][PATCH 1/7] Resource counters

the cache line and manipulate it. The expensive part is acquiring the cache line exclusively. This is expensive even if things are never contended but there are many users.

Sorry for the rant, but I just wanted to set the record straight. spin\_locks vs atomic ops is a largely meaningless debate.

Eric

–

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

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

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