

Re: Lockfree or not to lockfree that's the question !

Source:

<http://newsgroups.derkeiler.com/Archive/Comp/comp.programming.threads/2008-01/msg00088.html>

- *From:* "Chris Thomasson" <crisdom@xxxxxxxxxxxx>
 - *Date:* Sat, 5 Jan 2008 21:11:54 -0800
-

"Chris Thomasson" <crisdom@xxxxxxxxxxxx> wrote in message
news:KvqdnV4xELvnwx3anZ2dnUVZ_rSrNZ2d@xxxxxxxxxxxx

"David Schwartz" <davids@xxxxxxxxxxxx> wrote in message
<news:f2636424-ee26-435b-9235-76a226f3c67d@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>

On Jan 5, 1:48 pm, Dave Butenhof <david.buten...@xxxxxx> wrote:

Indeed; so then can we stick to concrete and objective advice for archetypical workloads and tasks and avoid the religious overgeneralizations entirely? ;-)

I think I've avoided religious overgeneralizations. I think it's fair to say "X is bad because Y" when Y is a serious downside to X, even if that doesn't make X objectively always bad.

If it's not those comments you are talking about, I'm not sure what is. I simply presented a list of lock-free algorithms downsides. My post didn't really contain any advice. Its sole purpose was to make sure people understood the downside of lock-free algorithms.

[...]

Explain in detail how spinning on an adaptive mutex's internal state *_fundamentally_* differs from spinning on a CAS failure wrt lock-free algorithms in general? IMVHO, its good to keep in mind that more than one mutex implementation decides to spin a little before deferring to the operating system... So be it as locks are essential to many lock-free algorithms indeed. Think writer-side of a low-overhead reader-pattern...

A little spin on the mutex is fine, as it can increase performance somewhat by turning things into a "temporary" spinlock-and-backoff semantics before the "ultimate" slow-path is hit, thus delegating to the kernel waitset that happens to be assigned to said mutex synchronization logic...

Re: Lockfree or not to lockfree that's the question !

Here is a "hint": The CAS wrt lock-free is spinning an `_actual_application_data_`, while the CAS wrt mutex is spinning on its proxy private/internal state... Sometimes you only need to atomically modify a word or two...

Basic fundamental difference; no?

.