pthreads: test, flag setting, and signaling under lock

by Alexei Finski

Need for flag setting and signaling under lock:

If a trader does not set flag and signal under lock, it can interleave with the execution of a client thread in bound-buf-condvar2 in the following way:

1) a client tests !order->fulfilled under lock as true and is preempted before calling pthread_cond_wait in cond_wait_perror,
2) without acquiring the lock, a trader sets the fulfilled flag to true and signals cond_fulfilled, which is lost because the client is not yet waiting,
3) the client calls pthread_cond_wait in cond_wait_perror and remains blocked.

Need for predicate (re)testing under lock:

A signal unblocks "at least one" thread (pthread specification). If (re)testing is not preformed under lock, client threads can interleave in the following way:

a) after a cond_not_full signal, if there are blocked client threads, at least one client thread is unblocked; however, another client thread, not previously blocked, acquires mutex before an unblocked client thread reacquires mutex; the former fills the order queue to capacity.

b) two blocked client threads are unblocked with a cond_not_full signal; one unblocked client thread reacquires mutex, fills the order queue to capacity, and releases mutex; the other unblocked client thread reacquires mutex, but the order queue is full.