

---

Subject: Re: bug in CoWork since C++11  
Posted by [mirek](#) on Sat, 30 Jul 2016 16:36:12 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

crydev wrote on Mon, 11 July 2016 19:36 I did some more debugging, and I found that it goes wrong in this function, in CoWork.cpp on line 196. It will wait infinitely until jobs are done, while there are still jobs to be done.

```
void CoWork::Finish() {
    if(!pool) return;
    Pool& p = *pool;
    p.lock.Enter();
    while(todo) {
        LLOG("Finish: todo: " << todo << " (CoWork " << FormatIntHex(this) << ")");
        if(todo == 0)
            break;
        if(p.scheduled)
            Pool::DoJob();
        else {
            p.lock.Leave();
            LLOG("WaitForFinish (CoWork " << FormatIntHex(this) << ")");
            waitforfinish.Wait(); // <---- Infinite wait here!
            p.lock.Enter();
        }
    }
    p.lock.Leave();
    LLOG("CoWork " << FormatIntHex(this) << " finished");
}
```

It is a very annoying bug. :(

Thanks,

crydev

Sorry I have missed this important reply... Anyway, 2 things:

1. I have recently refactored CoWork to use ConditionVariable instead of Semaphore. There is a tiny chance that this alone will fix the issue.... (if there was a bug in CoWork).
2. You might try some logging... E.g. activate LLOG in CoWork

Is not it possible that some of thread is frozen because of some deadlock?

Mirek

---