Subject: Re: CoWork buggy!?
Posted by mirek on Sun, 23 Mar 2008 13:11:41 GMT
View Forum Message <> Reply to Message

Werner wrote on Sun, 23 March 2008 06:38luzr wrote on Sun, 23 March 2008 08:18Well, not sure if I fully undestand the complaint, but if the issue is that Finish does not terminate threads, it is exactly what we wanted. Once created, threads are terminated only at app exit.

The reason is obvious - creating / terminating threads is somewhat expensive. So we have a pool of worker threads ready when next CoWork hits. And they are shared across all CoWorks as well, even nested.

Mirek
Ok, maybe I misunderstood the idea of "CoWork" . If so, sorry for that . After all there isn't any documentation and the code is written in a style which - well, let's say - I'm not familiar with .

Anyway, this raises bags of questions of which the following 2 are my favorites:

1.

Under which conditions should "CoWork::Finish" be called?

Finish waits until all the work required by calling "Do" is finished.

Quote:
What is "CoWork::waitforfinish" for?

It is used for synchronization. If there are any unfinished jobs, Finish has to wait until they are finished.

See:

```
bool CoWork::Pool::DoJob()
{
 Pool& p = pool();
 if(p.jobs.Top().work == NULL) {
  LLOG("Quit thread");
  return true;
 }
 MJob job = p.jobs.Pop();
 p.lock.Leave();
 job.cb();
 p.lock.Enter();
 if(--job.work->todo == 0) {
  LLOG("Releasing waitforfinish of (CoWork " << FormatIntHex(job.work) << ")");
```

```
  job.work->waitforfinish.Release();
 }
 LLOG("Finished, remaining todo " << job.work->todo << " (CoWork " << FormatIntHex(job.work)
<< ")");
 return false;
}
```

In this method, thread picks a new job to do. Each job has associated pointer to CoWork, where todo is number of jobs that yet has to be finished (you order a job to be finished by calling "Do"). If this goes to zero, all the work is done and waitforfinish semaphore can be released, so that Finish will get past waitforfinish.Wait.

a) no waiting job, i. e. "todo == 0":

Nothing is done. "CoWork::waitforfinish" remains unchanged.

It is not waiting job, it is *unfinished* job! (Includes waiting jobs and jobs that are currently being processed).

The conclusion is: "CoWork::waitforfinish" is never decremented and might raise without limit 8o .

Well, it is true that the last "Release" can have no matching Wait, but that is hardly a problem, as in that situation (todo == 0), everything is finised and we do not need to wait...

Mirek
