Subject: 2 basic problems
Posted by kohait00 on Mon, 08 Feb 2010 21:29:17 GMT
View Forum Message <> Reply to Message

hi mirek,

i am still thinking about the WorkQueue.. there is 1 basic problem to face:

* how to ensure that the common thread pool does not execute more than X jobs from a specific CoWork instace in parallel. since the pool does not care about sequencing relative to a CoWork instance. (restrict the global pool itself is no option, this would affect other CoWorks with proper different constraints)

the threads serve themselves from a global job queue. if we i.e say, only X=1 task from all the tasks commited to a certain CoWork may execute at once, other threads would still dequeue and execute jobs, since they dont care.

any idea? to make a per CoWork queue, which then really submits to the global queue?

i mean the following:
task A1 is executing in a thread from pool. task A2 (related to A1, may not intersect it) is also posted there, but may not execute, before A1 is done, meanwhile any other thread from pool finishes, dequeues A2 task, checks and realizes that A1 is still running, and it may not execute it. what to do with the job? the thread may not wait, it would block work of other CoWorks, post the job back to queue is not reliable, other related tasks might have been posted meanwhile. results in desorder.

seems as here we come close to a scheduler