

## ProCKSI - Enhancement #34

### Scheduling: queue vs. cron job

08/02/2007 09:16 AM - Anonymous

<b>Status:</b>	New	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Anonymous	<b>% Done:</b>	0%
<b>Category:</b>	ProCKSI/scheduling	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	9.0		
<b>Resolution:</b>			
<b>Description</b>			
<p>- <b>Current State:</b> The complete scheduling is done by cron jobs including the entire post-processing, e.g. uncompressing the TGZ output, registering results, finishing tasks/requests.</p> <p>- <b>Potential Problem:</b> The cron jobs start at fixes times, e.g. every 5 minutes, check the status and react accordingly. If the post-processing takes longer than the interval between two cron job starts, two or more cron jobs might run at the same time concurrently.</p> <p>- <b>Possible Solutions:</b></p> <ol style="list-style-type: none"><li>1. Lock the program that is being started by a cron job, e.g. <i>check_tasks.lock</i>, and don't start a new cron job before this lock file has been deleted again at the end of the first cron job's run.</li><li>2. Do not perform any "real tasks", e.g. post-processing, from within the cron job, but rather submit a new job to a special <i>administration</i> queue. Introduce new task states for post-processing as to exclude the job from being submitted multiple times into the queue within multiple cron job runs.</li><li>3. Implement a daemon that controls all activities instead of running several cron jobs potentially in parallel. Possible candidate: <i>Schedule::Cron</i></li></ol>			

#### History

#1 - 08/02/2007 09:16 AM - Anonymous

- **Temporary Solution:** Solution 1 is being implemented.
- **Final Solution:** A combination of Solution 2 and 3 will be favourable