

Monitord Management via Condor

Table Of Contents:

- 1 Motivation
- 2 Solutions
 - 2.1 Monitord is added as an independent condor job in the executable workflow created by Pegasus.
 - 2.1.1 Open Question: How does monitord know workflow has completed?
 - 2.1.2 Open Question: Discount monitor job in workflow.
 - 2.2 Monitord is a condor job separate from the executable workflow Pegasus creates
 - 2.2.1 Open Questions
 - 2.3 Wrapper around condor dagman
 - 2.3.1 Open Question
 - 2.4 Keep monitord separate Unix process as-is
 - 2.5 Not use monitord for notifications

Motivation

As part of adding notification support in Pegasus and monitord, we would want monitord to be managed by Condor. Currently, monitord is launched via pegasus-run as a separate process.

In case of system crashes, condor comes up automatically but monitord does not. This is a problem, as we will lose notifications in that case.

Solutions

There are several ways to get monitord to launch via Condor

Monitord is added as an independent condor job in the executable workflow created by Pegasus.

- Using Condor DAGMan priority we can ensure that DAGMan runs the job first.
- In case of restarts DAGMan will restart the monitord process also.

Open Question: How does monitord know workflow has completed?

Currently monitord looks at the dagman out file to determine whether a workflow has finished. Since now monitord is a part of the workflow itself, it cannot rely on that.

Monitord either

- detects from dagman log that it is the only job remaining
- or relies on its internal state to determine no other jobs are left to execute.

Open Question: Discount monitor job in workflow.

There is also the open issue about confusion caused by the extra job in the workflow, which will need to be (un)accounted for by pegasus-statistics, and any form of statistics.

Monitord is a condor job separate from the executable workflow Pegasus creates

In this case either

1. Pegasus creates the separate condor job outside the workflow
2. Or pegasus-run creates the condor job and submits it in addition to submitting the dag file to condor dagman

Open Questions

- How do we ensure that monitord runs first, before dagman starts running.
- issue of starvation, as there is limit on local universe jobs. It could be that all local universe jobs are monitord instances and there is no dagman running because of it. Or vice versa

Wrapper around condor dagman

- We have a wrapper around condor dagman, that first launches monitord and then execs condor dagman
- pegasus run will submit a condor job that refers to the wrapper instead of condor dagman

Open Question

- Users will get confused as they see the wrapper script instead of condor dagman.
- ~~condor_dag won't work~~ It does work, just the dagman name is different.

Keep monitor separate Unix process as-is

As shown in the motivation, a user ceases to get notification, if something untoward happens to monitor, or the entire system.

Not use monitor for notifications

Certain fine-grained notification will not be possible. However, we don't have to travel the whole journey in one step. Push Condor to support multiple PRE and POST scripts, and do some notifications from within DAGMan. Not all use-cases can be handled, but it is a start.

Another **big disadvantage** is no workflow level notifications. This is because of lack of postscripts at the dag level. In case of hierarchical workflows, we will get notifications for sub workflows as they are jobs in the parent workflow.