

FUNCTIONS of DAGMan and Problems related to ERROR RECOVERY

Written By : Anand Balasubramanian.

Date : 5/9/2002.

DAGMan

DAGMan is a meta-scheduler to schedule jobs that have inter- dependencies. A Directed Acyclic Graph(DAG) can represent a set of jobs which depend on each other. The output of one program maybe the input for the other program. In that case the second program needs to wait for the first one to finish its execution and then it gets the output of that and uses it as its input. Such ordering of jobs can be represented by means of a DAG. In a DAG the programs or the jobs are represented by means of nodes in the graph and their dependencies are represented by means of edges of the graphs. DAGMan is used as a meta-scheduler for Condor jobs. Jobs are submitted in the order in which they are specified in the DAG and the results are processed. The DAG is specified by means of a file, which has description about the nodes and their dependencies. Each of the node has its own Condor submit description file. In the file describing the DAG, we can specify a PRE script and a POST script for each of the job. A PRE script is the one that is executed on the local machine to which the DAG is submitted and it is used to make sure that the executables are present and other processing required before the actual execution of the jobs. The POST script is run once the job gets completed.

ERROR RECOVERY:

Job-resubmission on failure is not supported in DAGMan. The DAG is aborted once a node fails. But there is feature provided by DAGMan for us to see where the error occurred and also to make the re-submission easier. When a job fails, DAGMan will create what is called a **Rescue DAG**. The Rescue DAG is a input file, just like the file that describes the DAG, but it marks nodes that completed successfully as DONE. So now when we re run the DAG using the Rescue DAG, nodes that have been completed will not be executed again. So this way it saves lots of processing time. Rescue DAG will be created based on the following failures.

1. When the PRE script returns a non-zero value.
2. When the job fails.
3. When the POST script returns a non-zero value.

ISSUES IN ERROR RECOVERY:

1. 1.When a job is submitted to the globus universe, and if the job fails, the Rescue DAG is not getting created. When a job fails it has to return a non-zero exit code for a Rescue DAG to be created. The reason given by the people who wrote Condor is, jobs submitted to globus universe are unable to return their exit code

- and so they are always assumed to be completed successfully and also globus has limitations in some of the job managers it supports.
2. When the executable of the job is not present , i.e if the file is not available, in that case DAGMan just hangs.

SOLUTIONS :

1. To get around the first problem, the job should be wrapped around a script, and this script captures the exit code from the actual executable and writes it to a file. This file is then read by the POST script and it does the appropriate thing based on the exit code.
2. For the second problem, the jobs which don't run are signaled as HELD, so oneway to get around this is, fix the problem and use condor_release to continue the processing. The other way is to remove that job using condor_rm and DAGMan will notice this and will assume that the job failed and will generate the Rescue DAG.