More Aggressive Flocking

# Overview

Flocking is when the schedd sends jobs outside the local pool, because there are insufficient resources to fulfill the requirements of the submitters using the schedd. Flocking is implemented by using the FLOCK\_TO HTCondor configuration variable. FLOCK\_TO is merely a list of collectors; on the CHTC submit node *submit-1.chtc.wisc.edu,* the FLOCK\_TO configuration variable is set to

FLOCK\_TO = glidein.chtc.wisc.edu condor.cae.wisc.edu glow.cs.wisc.edu wid-cm.discovery.wisc.edu condor.cs.wisc.edu

As implemented, this means that the CHTC submit host first tries to fulfill all requests from the CHTC pool. Then it tries the glidein pool, then CAE, then GLOW, and so on until it reaches the end of the list. We say that glidein corresponds to flock level 1, CAE to flock level 2, and so on. The schedd only increments a flock level by an increment of 1 when trying another pool to fulfill job requests. The flock level varies according to the submitter or owner of the job.

# Architecture of Proposed Solution

We propose to add a FLOCK\_INC configuration knob that defaults to 1. When the schedd decides to change the flocklevel of a submitter, it will increase the flock level of the submitter by the FLOCK\_INC value instead of 1. In the CHTC, setting the flock increment to 2 will cause the schedd to try to flock jobs to glidein and to CAE, rather than to glidein only (which corresponds to FLOCK\_INC = 1). In this way, we speed up the flocking.

# Development Plan

To implement the code change takes one day. To insert it into the test suite will take another day. We only need to implement capturing FLOCK\_INC from the config file and then inspect the SchedLog to verify that the flock level has increased by an increment bigger than 1.