# Design Document: Statistics Monitoring

## Motivation

Each HTCondor daemon keeps important statistics of its internal state and activities. These statistics have been present for several years (greatly maturing in 2012) but they do not appear to be widely used by sites.

We believe one reason is that the task of integrating the HTCondor statistics into the site monitoring system is left to the sysadmins. This task is not trivial and requires some amount of expertise.

This design document proposes a framework for pushing statistics from HTCondor into Ganglia. This same framework could be adapted to other monitoring systems.

## Architecture and Implementation

A new daemon condor\_gangliad will be added to HTCondor. This daemon will periodically query the collector to retrieve desired statistics. It may also query individual daemons to retrieve statistics that are not published to the collector. It will then publish statistics to ganglia.

The condor\_gangliad daemon *could* also receive ads pushed to it by the collector via the classad forwarding mechanism in the collector. However, we prefer to first implement a solution that does not rely on this. The advantage of a pure query-based approach is that condor\_gangliad can monitor a pool to which it has read access without any configuration changes being made to that pool. Also, the query projection mechanism can be used to limit how much data is fetched, whereas no such mechanism currently exists for ads that are forwarded.

Simply pushing all numeric attributes into ganglia does not seem very user-friendly. We propose instead to make the publication code handle specific statistics on a case-by-case basis, using patterns where possible to simplify. This is what the current CHTC scripts do.

The publication to ganglia will use libgmond. The libgmond library gives us a simple API to access the worker node’s Ganglia configuration and send Ganglia UDP packets. We do not want the ganglia rpm to be a prerequisite of the UW’s condor rpm. For the UW condor rpm, we plan to do as we do for other externals: ship the ganglia library with condor but preferentially use the system version of the library if it is installed. For the OSG condor rpm, if support for ganglia is desired, we would use the “proper” approach: condor\_gangliad would be put in its own rpm, which depends explicitly on both condor and ganglia.

## Development Plan

We have code for ganglia library interaction from Brian Bockelman’s prototype project. We also have Becky Gietzel’s perl code that selects specific attributes for publication in CHTC. The first version of condor\_gangliad will borrow from both of these.

Version 1:

* [ 1 day ] Create condor\_gangliad framework to gather a list of attributes from the collector. Where possible, structure the code so it could be used in other daemons that interact with other monitoring systems.
* [ 1 day ] Get publication to ganglia working.
* [ 2 days ] Refine the basic set of statistics to publish to ganglia. For now, it will just be a hard-coded list of attributes with hard-coded publication properties, such as type and units.
* [ 0.5 day ] Add some basic aggregate statistics: size and state of pool.

After version 1 is done, we could consider additional developments, such as higher levels of verbosity, customizability, aggregation and manipulation of statistics, special ganglia graph definitions, etc.