Running MCCE Jobs with Condor on Sibyl

Yanjun Wang (yw319@cs.nyu.edu), 11/26/2003

What is Condor?

Condor is a distributed job scheduler developed by the computer science department at Univ. of Wisconsin – Madison. It utilizes the computing power of a pool of computers that communicate over a network.

When a user submits the job to Condor, Condor finds an available computer in the pool and begins running the job on it. Condor can check the available resources and assign jobs in a way that balances workload.

How to submit a job?

Submitting a job is telling Condor a description file by command:

```
condor_submit descriptionfile
```

Here is a sample description file:

```
#####
# Submit a MCCE job on sibyl
#####

universe    = vanilla
executable  = /home/mcce/mcce1.0/bin/mcce
output      = run.log
error       = condor.err
log         = condor.log

Notification = always
notify_user = yanjun@sci.ccny.cuny.edu
### If you don't want to get emails:
# Notification = never

initialdir  = RC/2prc
queue

initialdir  = RC/1aig
queue

### End of description file ###
```

`Universe:`

vanilla (serial jobs/independent of the condor library). We'll use 'vanilla' universe for mcce jobs.

`Executable:`

The absolute path and file name of the executable. Since Condor relies on the shared file system to find the executable, make sure the specified executables are in the shared file systems.

`Output:`

The output file name will capture what the program would normally write to stdout.

`Error:`

The error file name will capture any error messages the program would normally write to the stderr.

`Log:`

This file keeps condor's log of this job.

`Initialdir`:

Used to specify the initial working directory for the job in the queue. It should be a path to a preexisting directory. If not specified, the Initialdir is the directory where condor_submit is invoked. Output, error, and log files are all kept in `initialdir`.

`Notification:`

Shall condor notify you when a job completes. Possible values are "never" and "always".

`Notify_user`:

when condor finishes a job, it'll notify the user by this email address.

`Queue:`

Send a job to the condor queue. When you want to set up runs in different directories, then you may prepare multiple Initialdir/Queue pairs.

Job Scheduling - Preemption and Priority

To ensure that each user gets the fair share of using the computing resources, condor constantly calculates the priority for each user and all jobs in the queue. It means even if job 1 is submitted before job 2, it is not necessarily that job 1 executes before job 2.

Currently, preemption, or interruption of a running job, is disabled in all three clusters to make sure MCCE jobs can run through. Once a program starts, it will run to the end. The reason is that condor is unable to checkpoint jobs with external calls like MCCE jobs.

User Command Summary

Category	*Command*	*Option/Arguments*	*Description*
Submit job	`condor_submit`	`descriptionfile`	Submit a job
Monitor Progress	`condor_status`		Display the status of the condor pool
	`condor_q`		Display the jobs in the queue
		`User`	Display the jobs in the queue submitted by this user
		`-analyze`	Inquire resources for the queued jobs
Edit Queue	`condor_hold`	`-all`	Put all jobs on the local node into the hold state
		`User`	Put jobs of this user into hold state
		`Cluster.process`	Put specific job into the hold state
	`condor_release`	`-all`	Release all held jobs
		`User`	Release jobs of the user
		`Cluster.process`	Release specified job
	`condor_rm`	`-all`	Remove all queued jobs
		`User`	Remove jobs of the user
		`Cluster.process`	Remove specified job
Review Completed Jobs	`condor_history`		View log of condor jobs completed to date
Adjust job priority	`condor_prio`	`(+|-|-p) value` *`Cluster.process`*	Increase/Decrease/Set job priority for a job. The job priority ranges from –20 to +20 and the default value is 0.