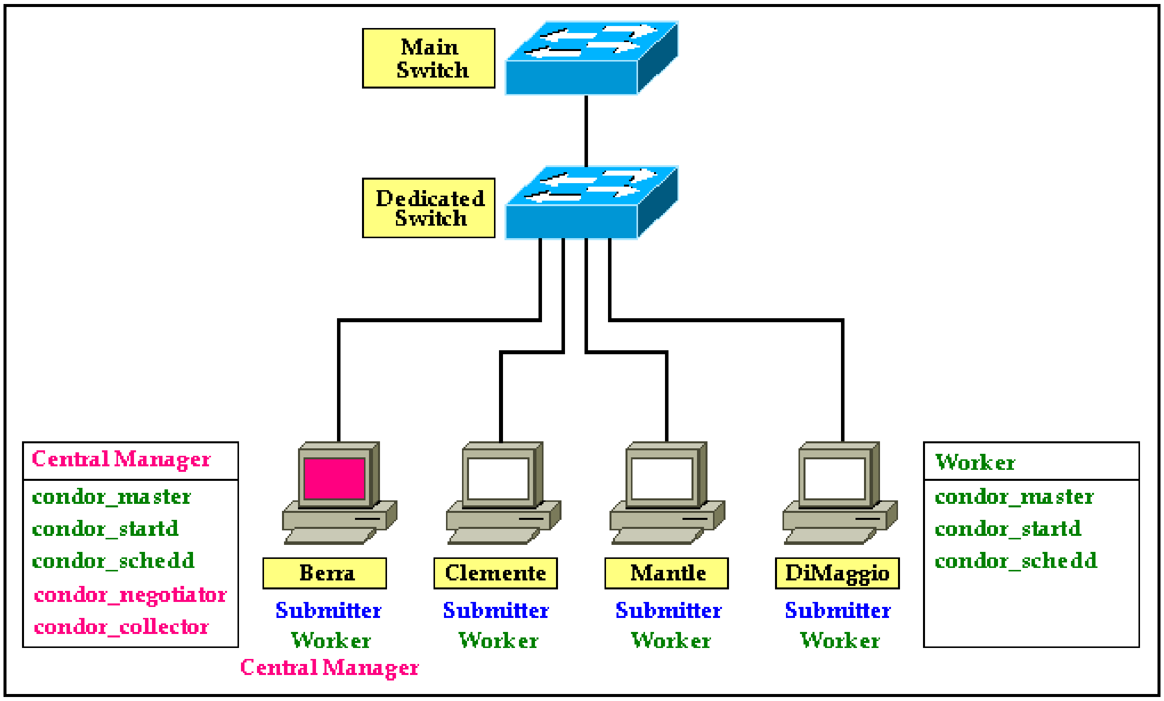
Condor Flock – User Manual

Index

1. Current Network Topology



2. Current Network Configuration

Hostname	IPv4 Address	Subnet Mask	Gateway	DNS / Secondary DNS
Berra	128.205.44.164	255.255.255.192	128.205.44.129	128.205.32.8 128.205.32.12
DiMaggio	128.205.44.165			
Clemente	128.205.44.166			
Mantle	128.205.44.167			

3. Steps to Install/Configure Condor

Version	Condor for Windows 7.0.1

1. End User License Agreement
2. Create new Condor Pool (New Pool: __)
   Or
   Join existing Pool (Central Manager Hostname: __)
3. Submit jobs to Condor Pool
   When to run jobs?
   a. Never
   b. Always, never suspend
   c. When keyboard idle for 15 minutes
   d. When keyboard idle for 15 minutes and CPU idle
   When machine is no longer idle, jobs get suspended
   After 10 minutes:
   a. Keep jobs in memory and restart it when you leave
   b. Restart on different machine
4. Accounting Domain
5. Send email (SMTP server: __, Email address of administrator: __)
6. Path to Java Virtual Machine
7. Access Permissions
   a. Hosts with read access
   b. Hosts with write access
   c. Hosts with administrative access
8. Enable VM Universe: Yes / No (Requires VMware server and Perl)

When Condor is installed, it creates configuration files which include the parameters specified in the above installation process. These files are used to customize how Condor operates at a given site. As part of its initialization process, each Condor program configures itself by calling a library routine which parses the various configuration files that might be used including pool-wide, platform-specific, machine-specific, and root-owned configuration files. Environment variables may also contribute to the configuration.

Current Configuration

Accounting Domain: <hasn’t been specified> (usually DNS domain)
SMTP Server: <hasn’t been specified>
Email address of Administrator: <hasn’t been specified>
JVM Path: C:\Program Files\Java\jre1.6.0_02\bin\JAVA.exe
Hosts with Read Access: $(FULL_HOSTNAME), berra.cse.buffalo.edu
Hosts with Write Access: $(FULL_HOSTNAME), berra.cse.buffalo.edu
Hosts with Administrator Access: $(FULL_HOSTNAME), berra.cse.buffalo.edu
Enable VM Universe: No (If enabled, requires VMware server and also path to the Perl)

4. Required services/daemons

Ensure that the following services/daemons are running on each Worker node:

1. condor_master
2. condor_startd
3. condor_schedd

Ensure that the following services/daemons are running on the Central Manager:

1. condor_master
2. condor_startd
3. condor_schedd
4. condor_negotiator
5. condor_collector

Other daemons that may be used by Condor:

1. condor_starter
2. condor_shadow
3. condor_lcbdd
4. condor_ckpt_server
5. condor_quill
6. condor_abmsd
7. condor_gridmanager
8. condor_had
9. condor_replication
10. condor_procd

5. Roadmap to Run a Job on Condor

1. Code Preparation
2. Condor Universe
3. Submit Description File
4. Submit Job

Code Preparation

A job run under Condor must be able to run unattended as a background batch job. The program cannot do interactive input and output. However, Condor can redirect console output (stdout and stderr) and keyboard input (stdin) to and from files. Hence, before submitting a job, create the required files containing the proper keystrokes needed for program input and ensure that the program will run correctly with the files.

Condor Universe

The Condor Universe specifies a Condor execution environment.

universe = standard | vanilla | scheduler | local | grid | mpi | java | vm

The **standard** Universe is the default (except where the configuration variable `DEFAULT_UNIVERSE` defines it otherwise), and tells Condor that this job has been re-linked via *condor_compile* with the Condor libraries and therefore supports checkpointing and remote system calls. The **vanilla** Universe is an execution environment for jobs which have not been linked with the Condor libraries. This may be used to submit shell scripts to Condor. There is no way to take a checkpoint or migrate a job executed under the vanilla universe. The **scheduler** is for a job that should act as a metascheduler. The **grid** universe forwards the job to an external job management system. The **mpi** universe is for running mpi jobs made with the MPICH package. The **java** universe is for programs written to the Java Virtual Machine. The **vm** universe facilitates the execution of a virtual machine.

Submit Description File

The submit description file contains information about the job such as what executable to run, the number of times to run a program, the files to use for keyboard and screen data, the platform type required to run the program, and where to send e-mail when the job completes.

Each submit description file describes one cluster of jobs to be placed in the Condor execution pool. All jobs in a cluster must share the same executable, but they may have different input and output files, and different program arguments. The submit description file is the only command-line argument to *condor_submit*. If the submit description file argument is omitted, *condor_submit* will read the submit description from standard input.

The submit description file must contain one *executable* command and at least one *queue* command. All of the other commands have default actions.

Example : trialjob.cmd

```
Executable = trial.exe
Universe = vanilla
Output = trial.out
Error = trial.err
Log = log.log

should_transfer_files = YES
when_to_transfer_output = ON_EXIT

transfer_output_files = trial.out, output.txt

Queue
```

Sample Log File

```
000 (008.000.000) 04/15 15:09:06 Job submitted from host:
<128.205.44.167:49192>
...
001 (008.000.000) 04/15 15:09:09 Job executing on host:
<128.205.44.167:49193>
...
006 (008.000.000) 04/15 15:09:18 Image size of job updated: 1796
...
005 (008.000.000) 04/15 15:11:48 Job terminated.
      (1) Normal termination (return value 0)
            Usr 0 00:02:22, Sys 0 00:00:00  -  Run Remote Usage
            Usr 0 00:00:00, Sys 0 00:00:00  -  Run Local Usage
            Usr 0 00:02:22, Sys 0 00:00:00  -  Total Remote Usage
            Usr 0 00:00:00, Sys 0 00:00:00  -  Total Local Usage
      3900  -  Run Bytes Sent By Job
      477094  -  Run Bytes Received By Job
      3900  -  Total Bytes Sent By Job
      477094  -  Total Bytes Received By Job
...
```

6. File Transfer Mechanism

If jobs are submitted without using the File Transfer mechanism in vanilla, java, parallel (or MPI) universe, Condor uses a shared file system to access input and output files. In this case, the job *must* be able to access the data files from any machine on which it could potentially run.

For jobs submitted under the standard universe, a shared file systems is not necessary, since Condor's remote system call mechanism handles access to files (input and output). The executable and checkpoint files are transferred automatically, when needed.

For the vanilla, java, MPI, and parallel universes, access to files (including the executable) through a shared file system is presumed as a default on UNIX machines. If there is no shared file system, then Condor's file transfer mechanism must be explicitly enabled. However, when submitting a job from a Windows machine, Condor presumes the opposite: no access to a shared file system. For Windows machines, the file transfer mechanism is enabled by default. The submit description file only needs to specify which files to transfer, and/or when to transfer the output files back.

In the grid universe, jobs are executed on remote machines which are unlikely to have a shared file system. In the scheduler universe, Condor only uses the machine from which the job is submitted. Therefore, the existence of a shared file system is not relevant.

7. Some frequently used commands

1. Start/Stop the condor services

```
      C:\> net start condor
      C:\> net stop condor
```

2. Setup a user account on a Windows machine

```
      C:\> condor_store_cred_add
      Account: condorsu @ MANTLE
      Enter password:
```

On a Windows machine, condor_store_cred stores the password of a user/domain pair securely in the Windows registry. This is used to run jobs with the user ID of the submitting user. Condor also uses this password to acquire the submitting user's credentials when writing output or log files. When condor_store_cred is invoked, it contacts the condor_schedd daemon to carry out the requested operations on behalf of the user. This is necessary since registry keys are accessible only by the Windows SYSTEM account, not by administrators or other users.

3. Queue jobs for execution under Condor

```
      C:\> condor_submit trialjob.cmd
```

condor_submit requires a submit description file which contains commands to direct the queuing of jobs. One submit description file may contain specifications for the queuing of many Condor jobs at once. A single invocation of *condor_submit* may cause one or more clusters. A cluster is a set of jobs specified in the submit description file between **queue** commands for which the executable is not changed.

4. Display information about jobs in Queue

```
      C:\> condor_q
```

The following information is displayed :

- ID – cluster/process id
- OWNER
- SUBMITTED – date & time of submission
- RUN_TIME – wall-clock time)

- ST – current status of the job
  - U = unexpanded (never been run)
  - H = on hold
  - R = running
  - I = idle (waiting for a machine to execute on)
  - C = completed
  - X = removed
- PRI – User specified priority of the jobs (-20 to +20)
- SIZE – Virtual image size of executable (MB)
- CMD – Name of the executable

5. Display the status of the condor pool

```
C:\> condor_status
C:\> condor_status -submitters
```

condor_status can be used to query resource information, submitter information, checkpoint server information, and daemon master information.

6. Remove jobs from condor queue

```
C:\> condor_rm <cluster-id>
```

7. View log of Condor jobs completed to date

```
C:\> condor_history
```

8. Future work

1. Setup the email mechanism.
   We have 2 options:
   a. Setup our own SMTP service.
   b. Work with cse-consult and obtain permissions/credentials to use the CSE servers.
2. Setup DHCP so that we can get other workstations on this flock and also setup the other 2 flocks.

9. References / Resources

1. Condor binaries : http://www.cs.wisc.edu/condor/downloads-v2/download.pl
2. Condor Manual : http://www.cs.wisc.edu/condor/manual/index.html
3. UB DHCP Management System : http://dhcp.cit.buffalo.edu/
4. Contact cse-consult@cse.Buffalo.EDU for IT-help.