# Condor Roll: Users Guide

**Condor Roll: Users Guide :**

Version @VERSION@ Edition

Published Jul 2008

Copyright © 2008 UC Regents

# Table of Contents

# Preface

The Rocks Condor Roll uses the latest stable Condor Release to provide High Throughput Computing environment for Rocks clusters. The Rocks Condor Roll builds on the very good work by the Condor team, to seamlessly install the *de facto* standard grid middleware on Rocks Clusters.

Please visit the Condor Project Homepage[1] to learn more about Condor and the software releases.

# Notes

1.   http://www.cs.wisc.edu/condor/

# Chapter 1. Requirements

## 1.1. Rocks Version

The Condor Roll is for use with Rocks version 7.0.1 ().

## 1.2. Other Rolls

The Condor Roll is does not require any other Rolls to be installed on the Frontend. Compatiblity has been verified with the following Rolls.

- HPC
- Patch

# Chapter 2. Installing the Condor Roll

## 2.1. Adding the Roll

The Condor Roll must be installed during the Frontend installation step of your cluster (refer to section 1.2 of the Rocks usersguide). Future releases will allow the installation of the Condor Roll onto a running system.

The Condor Roll is added to a Frontend installation in exactly the same manner as the required HPC Roll. Simply select the Condor roll as you would any network- or CD-based roll.
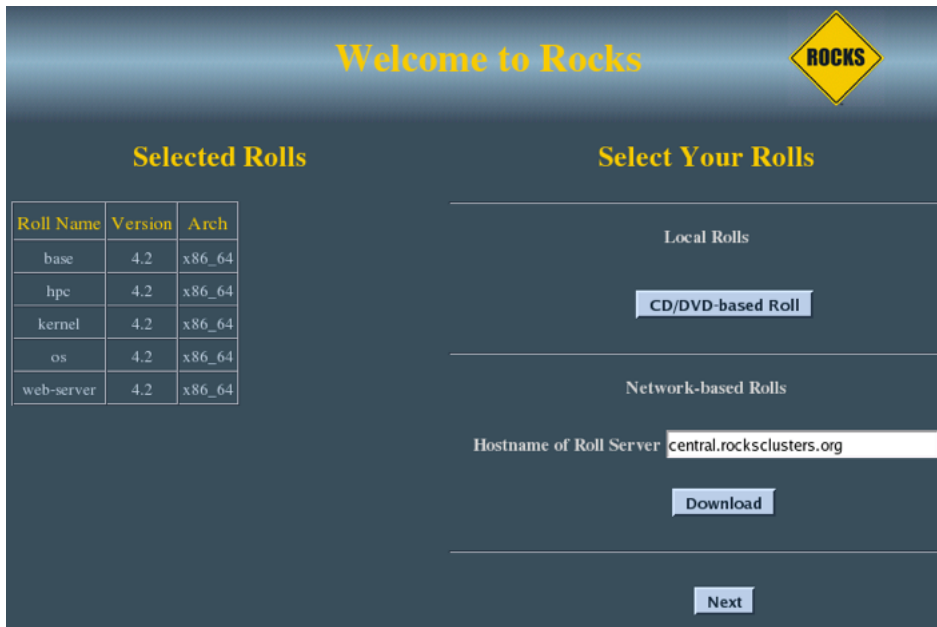


Once the Condor Roll is loaded the installer will continue installation automatically with no further user input. The Frontend will be configured as a condor pool manager and a submit node. The rest of the machines in the cluster will be configured as submit/execute nodes.

## 2.2. Frontend Boot

Near the end of the first startup of the newly installed Frontend node the Condor Roll will complete the installation and configuration of the Condor packages.

# Chapter 3. Using the Condor Roll

## 3.1. Using Condor

This section explains the Condor pool configuration on Rocks clusters, and provides link to further documentation.

Machines in the Condor pool can serve a variety of roles:

- *Central manager* machine is the collector of information, and the negotiator between resources and resource requests. In any Condor pool one machine must be the Central Manager.
- *Submit machine* allows Condor jobs submission.
- *Execute machine* provides resources for Condor jobs execution.

The Frontend is configured as the pool's Central Manager, and the Submit machine. The rest of the nodes are configured as Submit/Execute machines.

The configuration of condor is done during the install, the resulting configuration files are located in /opt/condor/etc. To reconfigure condor pool use /opt/condor/sbin/CondorConf command, and then restart condor daemons so the new configuration takes an effect. CondorConf takes *-h* flag that gives the usage explanation on stdout. Usually, the reconfiguration is only needed when you have multiple network interfaces on your nodes, and the default configuration is not using your desired interface.

By default, on the frontend condor daemons are running on *eth1* interface, and on the compute nodes on *eth0* interface.

Here is en axample command for configuring condor as the pool manager and the submit machine on the frontend, assuming the forntend has IP 198.202.89.155:

```
/opt/condor/sbin/CondorConf -n frontend -t sm -m 198.202.89.155
```

The following command will configure compute nodes as Submit/Execute machines:

```
/opt/condor/sbin/CondorConf -n compute -t se -m 198.202.89.155
```

To find information about administrering and using Condor Pools please see the original Condor manual at Condor manuals[1]

## 3.2. Testing the Condor Roll

1. First, make sure condor daemons are running by executing:

   ```
   # ps -ef | grep condor
   ```

   On the frontend, the output should be similar to following:

   ```
   condor    2623     1  0 Apr19 ?        00:04:26 /opt/condor/sbin/condor_master
   condor    2646  2623  0 Apr19 ?        00:20:25 condor_collector -f
   ```

```
condor     2647  2623  0 Apr19 ?        00:04:56 condor_negotiator -f
condor     2649  2623  0 Apr19 ?        00:00:02 condor_schedd -f
```

And on the compute nodes, the output should be similar to following:

```
condor    17007     1  0 Apr19 ?        00:01:09 /opt/condor/sbin/condor_master
condor    17009 17007  0 Apr19 ?        00:00:02 condor_schedd -f
condor    17010 17007  0 Apr19 ?        00:09:09 condor_startd -f
```

2. Try a test job submission.

```
# su - condor
$ cp /opt/condor/tests/hello.* .
$ condor_submit hello.sub
```

3. Check if jobs are submitted by executing:

```
$ condor_q
```

The output should be similar to:

```
-- Submitter: viz-build.rocksclusters.org : <137.110.119.116:55854> : viz-build.rocksclusters.org
 ID      OWNER          SUBMITTED    RUN_TIME ST PRI SIZE CMD
   7.0   condor         7/30 16:43  0+00:00:02 R  0    0.0  hello.sh

1 jobs; 0 idle, 1 running, 0 held
```

*R* in status column(ST) means running. *I* means idling.

4. Once the queue is empty (above command shows no jobs) can see the history of jobs execution with:

```
$ condor_history
```

To see all the nodes in the condor pool do:

```
$ condor_status
```

The output should be similar to:

```
Name         OpSys      Arch   State      Activity   LoadAv Mem   ActvtyTime

vm1@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:40:04
vm2@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:45:05
vm3@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:45:06
vm4@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:45:07
vm1@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:35:04
vm2@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:40:05
vm3@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:40:06
vm4@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:40:07
vm1@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:25:04
vm2@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:30:05
vm3@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:30:06
vm4@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:30:07
vm1@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:15:05
vm2@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:20:06
vm3@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:20:07
vm4@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:20:08
vm1@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:10:04
vm2@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:15:05
vm3@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:15:06
vm4@compute-0 LINUX      INTEL  Unclaimed  Idle       0.000  506  0+00:15:07
```

```
vm1@compute-0 LINUX       INTEL  Unclaimed  Idle        0.000   506  0+00:00:04
vm2@compute-0 LINUX       INTEL  Unclaimed  Idle        0.000   506  0+00:05:05
vm3@compute-0 LINUX       INTEL  Unclaimed  Idle        0.000   506  0+00:05:06
vm4@compute-0 LINUX       INTEL  Unclaimed  Idle        0.000   506  0+00:05:07
vm1@compute-0 LINUX       INTEL  Owner      Idle        0.860   506  0+00:00:09
vm2@compute-0 LINUX       INTEL  Unclaimed  Idle        0.000   506  0+00:00:05
vm3@compute-0 LINUX       INTEL  Unclaimed  Idle        0.000   506  0+00:00:06
vm4@compute-0 LINUX       INTEL  Unclaimed  Idle        0.000   506  0+00:00:07

              Machines  Owner  Claimed  Unclaimed  Matched  Preempting

  INTEL/LINUX       28      1        0         27        0           0

        Total       28      1        0         27        0           0
```

# Notes

1.  http://www.cs.wisc.edu/condor/manual