RedHawk™ High Availability NFS (HA-NFS) Version 4.2 Installation Guide

May 2007

0898018-010

READ ME BEFORE INSTALLING THIS PRODUCT

concurrent

Disclaimer

The information contained in this document is subject to change without notice. Concurrent Computer Corporation has taken efforts to remove errors from this document, however, Concurrent Computer Corporation's only liability regarding errors that may still exist is to correct said errors upon their being made known to Concurrent Computer Corporation.

License

Duplication of this manual without the written consent of Concurrent Computer Corporation is prohibited. Any copy of this manual reproduced with permission must include the Concurrent Computer Corporation copyright notice.

Trademark Acknowledgments

Concurrent Computer Corporation and its logo are registered trademarks of Concurrent Computer Corporation. All other Concurrent product names are trademarks of Concurrent while all other product names are trademarks or registered trademarks of their respective owners. Linux® is used pursuant to a sublicense from the Linux Mark Institute.

Concurrent Computer Corporation
2881 Gateway Drive
Pompano Beach, FL 33069

Note: Information subject to change without notice.

Contents

1.0. Overview

This document provides release information and discusses how to install, configure and uninstall the RedHawk™ High Availability NFS, model number WA9019-L, Version 4.2 distribution on a system running the RedHawk Linux® operating system.

1.1 Product Description

High Availability NFS (HA-NFS) combines all the components needed for a resilient file system under Concurrent’s RedHawk Linux operating system.

An HA-NFS cluster is a pair of computers acting as a single, dual-redundant file system. In normal operation, each node acts as a primary supplier for a subset of the total filesystem resources. Upon failure of either node, the resources of the failed node are acquired and made primary on the working node.

The primary benefit of an HA cluster is that the downtime exposure is limited. For example, if a power supply fails on one machine at 3 a.m., the services will be backed up and running on the partner node within seconds without any human interaction. Also, hardware and software maintenance, even those involving extended downtime, can be done with only a very limited unavailability. Depending on your data and applications, this downtime may be a few seconds or a few minutes.

Both machines must have access to the data required to run the applications, as well as network connectivity and other related resources. An external RAID array can be used to which both machines can be connected. Data is backed up between these two sets of RAID arrays using DRBD.

DRBD is a “Distributed Replicated Block Device” that utilizes local discs using a network connection for replication. DRBD can be thought of as a RAID-1 (mirrored drives) system that mirrors a local hard drive with a drive on another computer.

DRBD includes mechanisms for tracking which system has the most recent data, “change logs” to allow a fast partial re-sync, and startup scripts that reduce the likelihood that a system will come up in “split brain” operation.

Also included in HA-NFS is Heartbeat, a program that runs on both machines and elects one node as primary. Once up, the secondary machine monitors the primary and, in the event of a failure, will change the secondary node to primary. Heartbeat runs scripts on the primary node to bring the applications up and shut them down. In that way, it’s similar to a multi-machine “init” process.

The machines in the cluster communicate with each other by sending “heartbeat packets” about twice a second. Typically you would configure the heartbeats to run over multiple different paths, for example, via serial and network links. Missing a few seconds of these heartbeats could trigger a failover, so you need to make sure that rebooting a switch or unplugging a network cable will not trigger a failover.

Heartbeat provides standard “start” and “stop” scripts for doing many common tasks such as bringing up and down IP addresses or mounting partitions. It’s worth noting that the IP address scripts will send out gratuitous ARP packets to alert other devices on the network of the topology change. System start/stop scripts, typically found in **/etc/init.d**, can also be directly used with Heartbeat. For more complex tasks, custom scripts can be called to start or stop parts of the application.

Another part of Heartbeat is called STONITH, short for Shoot The Other Node In The Head. In the case of a primary node failure, the secondary node uses STONITH to ensure that the old primary is definitely down before bringing up the services on the secondary node.

Mon, another component of HA-NFS, is a general-purpose scheduler and alert management tool used for monitoring service availability and triggering alerts upon failure detection. Mon was designed to be open and extensible in the sense that it supports arbitrary monitoring facilities and alert methods via a common interface, all of which are easily implemented with programs in C, Perl, shell, etc., SNMP traps, and special mon traps. Mon views resource monitoring as two separate tasks: the testing of a condition, and triggering an action upon failure. Mon was designed to implement the testing and action-taking tasks as separate, stand-alone programs. Mon is fundamentally a scheduler which executes the monitors (each test a specific condition), and calls the appropriate alerts if the monitor fails. The decision to invoke an alert is governed by logic which offers various “squelch” features and dependencies, all of which are configurable by the user.

1.2 Documentation

1.2.1 High Availability NFS Documentation

This Installation Guide describes installing and configuring the HA-NFS Version 4.2 product on a RedHawk system.

HA-NFS includes the following open source products, each with its own online documentation:

- Heartbeat Version 2.0.8:
  - Online help files in /usr/share/doc/heartbeat-2.0.8
  - Man pages in /usr/man/man#:
    cl_status(1)
    hb_addnode(1)
    hb_standby(1)
    ha_logger(1)
    hb_delnode(1)
    hb_takeover(1)
    apphbd(8)
    ha_logd(8)
    ldirectord(8)
    stonith(8)
    cibadmin(8)
    heartbeat(8)
    meatclient(8)
    supervise-ldirectord-config(8)
- DRBD Version 0.7.23:
  - Man pages in /usr/share/man/man#:
    drbd.conf(5)
    drbd(8)
    drbdadm(8)
    drbddisk(8)
    drbdsetup(8)

- Mon Version 0.99.2:
  - Man pages in /usr/share/man/man#:
    moncmd(1)
    monshow(1)
    mon(8)

1.2.2 Additional Resources

The following web sites provide additional information.

DRBD:	**http://www.drbd.org**
Heartbeat:	**http://www.linux-ha.org/Heartbeat**
Mon:	**http://www.kernel.org/software/mon/**

1.2.3 RedHawk Linux Documentation

The following table lists RedHawk Linux documentation. Click on the red entry to display the document PDF. These documents are also available by clicking on the “Documents” icon on the desktop and from Concurrent’s web site at **www.ccur.com**.

Document Name	Document Number
RedHawk Linux Release Notes	0898003
RedHawk Linux User’s Guide	0898004
Real-Time Clock & Interrupt Module (RCIM) PCI Form Factor User’s Guide	0898007
RedHawk Linux Frequency-Based Scheduler (FBS) User’s Guide	0898005
iHawk Optimization Guide	0898011
RedHawk Linux FAQ	N/A

1.3 Product Updates

As HA-NFS updates are issued, they will be made available for downloading from Concurrent’s RedHawk Updates website, **http://redhawk.ccur.com**.

1.4 Syntax Notation

The following notation is used throughout this document:

italic Books, reference cards, and items that the user must specify appear in *italic* type. Special terms may also appear in *italic*.

`list bold` User input appears in **`list bold`** type and must be entered exactly as shown. Names of directories, files, commands, options and man page references also appear in **`list bold`** type.

`list` Operating system and program output such as prompts, messages and listings of files and programs appears in `list` type.

`[ ]` Brackets enclose command options and arguments that are optional. You do not type the brackets if you choose to specify these options or arguments.

hypertext links When viewing this document online, clicking on chapter, section, figure, table and page number references will display the corresponding text. Clicking on Internet URLs provided in ***blue*** type will launch your web browser and display the web site. Clicking on publication names and numbers in ***red*** type will display the corresponding manual PDF, if accessible.

2.0. Prerequisites

2.1 Software

- RedHawk Linux Version 4.2

  Note that HA-NFS is not supported on an upgrade from RedHawk Version 2.X to Version 4.2.

2.2 Hardware

- Two or more Concurrent iHawk or ImaGen systems linked by an Ethernet LAN

3.0. Installation

Follow the steps below to install HA-NFS on both the primary and secondary systems.

1. With RedHawk Linux Version 4.2 running, log in as root and take the system down to single-user mode:
   a. Right click on the desktop and select Open Terminal.
   b. At the system prompt, type **init 1**.
2. Insert the disc labeled “RedHawk High Availability NFS” appropriate to your system’s architecture into the CD-ROM drive.
3. To mount the cdrom device, execute the following command:

```
mount /media/cdrom
```

4. To install, execute the following commands:

```
cd /media/cdrom
./install-hanfs
```

   Follow the on-screen instructions.

5. When the installation completes, execute the following commands:

```
cd /
umount /media/cdrom
eject
```

6. Remove the disc from the CD-ROM drive and store. Exit single-user mode (Ctrl-D).

4.0. Configuration

After installing the product, perform the following configuration steps.

1. Perform the following on both nodes to activate NFS:

   a. `$ chkconfig nfs on`

   b. Increase $RPCNFSDCOUNT in **`/etc/init.d/nfs`** from a default count of 8 to a greater number depending on your NFS load. For high NFS loads, it is recommended that this value be incremented by 4 per client.

   c. `$ service nfs start`

2. DRBD devices need to be configured first on both nodes. Initial configuration of DRBD will take approximately 8 to 10 hours depending on the size of your RAID partitions.

   a. Edit **`/etc/drbd.conf`** on both nodes with values specific to your network. The contents of this configuration file should be identical on both nodes. Refer to **`/etc/ha.d/conf/examples/drbd.conf.example`**.

   b. Edit **`/etc/ha.d/conf/drbdConnect.cf`** on both nodes appropriately.

   c. On both nodes (relatively simultaneously), perform the following:

   ```
   $ service drbd start
   $ ls -l /dev/drbd*
   ```

   Both nodes will show as Secondary and Inconsistent. This is because the underlying storage is not in sync.

   d. Force one machine to be the primary for initial configuration by issuing the following on one of the nodes:

   ```
   $ drbdadm -- --do-what-I-say primary all
   ```

   The result is a full sync of the underlying devices (initial full sync). The device is usable right away, so if no file system currently exists, you should create one now.

   e. Create an ext3 filesystem for ***each*** **`/dev/drbd`***X* on the primary node:

   ```
   $ mkfs -t ext3 /dev/drbd0
   ...
   $ mkfs -t ext3 /dev/drbd15
   ```

   This will take quite some time as noted earlier.

   f. After the filesystem is created, it is very important that you shrink it by 128 MB on each of the partitions. This 128 MB is used by DRBD for its meta-data. If this step is not performed, data corruption will occur.

   To shrink the filesystem:

   i. Create **`/mnt/dir`***X* corresponding to **`/dev/drbd`***X* on each node. For example, **`/mnt/dir0`** through **`/mnt/dir15`** on both nodes. The names can be different than this example, but make these mount points the same on both nodes.

   ```
   $ mkdir -p /mnt/dirX
   ```

ii. Add the **/mnt/dir***X* entries to **/etc/fstab**. This must be the same on both nodes to facilitate a noauto mount upon failover. For example (using the name `dir`):

```
/dev/drbd0     /mnt/dir0    ext3   noauto   0 0
/dev/drbd1     /mnt/dir1    ext3   noauto   0 0
...
/dev/drbd15    /mnt/dir15   ext3   noauto   0 0
```

iii. Mount the file system in **/mnt/dir***X*

```
$ cd /mnt/dirX && df -h .
```

iv. Make note of the available space and subtract 128 MB from this.

v. Unmount the file system (the next step will complain if you don’t umount first):

```
$ umount /mnt/dirX
```

vi. Resize the filesystem, subtracting the 128 MB:

```
$ resize2fs -f /dev/drbdX NEW-SIZE
```

vii. Repeat for each DRBD device.

g. Allow the synchronizing to continue in the background. After some time, you should see that the primary and secondary devices are in sync and consistent. You can verify this by:

```
$ cat /proc/drbd
```

DRBD setup is now complete.

3. Configure Heartbeat next. Create all Heartbeat configurations the same on both nodes.

a. Edit **/etc/ha.d/ha.cf** on both nodes with values specific to your site. Refer to **/etc/ha.d/conf/examples/ha.cf.example**.

b. Edit **/etc/ha.d/haresources** on both nodes appropriately. Refer to **/etc/ha.d/conf/examples/haresources.example**.

c. Edit **/etc/ha.d/authkeys** on both nodes appropriately. Refer to **/etc/ha.d/conf/examples/authkeys.example**.

d. Change permissions to **authkeys** to enable starting Heartbeat:

```
$ chmod 600 authkeys
```

e. Create two nfs export files corresponding to the drbd partitions that will be nfs exported. Place them in **/etc/ha.d**. For example:

`exports.`*nodeA-hostname*
`exports.`*nodeB-hostname*

The name of the node must match ‘hostname’.

f. Start Heartbeat on both nodes simultaneously. Ensure that DRBD is running.

```
$ service heartbeat start
```

Heartbeat setup is now complete.

4. Next, configure Mon.

   a. Edit **/etc/ha.d/conf/mon.cf** on both nodes appropriately. Refer to **/etc/ha.d/conf/examples/mon.cf**

   b. Set up **mon.cf** with the appropriate eth*X* and ping node name. Mon will trigger an appropriate customizable alert if it stops receiving ping acknowledgement from the ping node.

      Example: (as provided in **examples/mon.cf**)

      ```
      Watch ping_heartbeat_gateway
          service ping1
              description ping hbt_gway
              interval 1s
              monitor ping.monitor eth0
              period wd (Sun-Sat)
              alert ipmi_suicide.alert
              alertevery 1h
      ```

      In this example, hbt_gway is the ping node between the two HA-NFS servers and eth0 is the direct interface to hbt_gway.

      hbGWCheck is a script that blocks the init until the ping node (hbt_gway) is available. This sychronizes the boot, not starting heartbeat until the ping node (hbt_gway) is up. This script uses the hbt_gway parameter specified in **mon.cf**.

   c. $ **service mon start**

5. After all the components of HA-NFS (DRBD, Heartbeat and Mon) have been manually tested on both nodes, perform the following to enable the services to be activated at boot time:

   ```
   $ chkconfig drbd on
   $ chkconfig heartbeat on
   $ chkconfig mon on
   ```

5.0. Uninstall

Perform the following steps if you wish to uninstall the entire HA-NFS Version 4.2 distribution from your system:

1. Log in as root and take the system down to single-user mode:

   ```
   init 1
   ```

2. Insert the disc labeled “RedHawk High Availability NFS” appropriate to your system’s architecture into the CD-ROM drive.

3. To mount the cdrom device, execute the following command:

   ```
   mount /media/cdrom
   ```

4. To uninstall, execute the following commands:

   ```
   cd /media/cdrom
   ./uninstall-hanfs
   ```

5. When complete, execute the following commands:

   ```
   cd /
   umount /media/cdrom
   eject
   ```

6. Remove the disc from the CD-ROM drive and store. Exit single-user mode (Ctrl-D).