

# Warewulf:

## The Cluster Node Management Solution

*SuperComputing 2003*



Greg M. Kurtzer

[GMKurtzer@lbl.gov](mailto:GMKurtzer@lbl.gov)

Lawrence Berkeley National Laboratory

<http://warewulf-cluster.org/>

- **A brief background and intro**
- **Warewulf design**
  - Summary of design goals
- **Warewulf implementation**
  - Some implementation specifics
- **Warewulf walkthrough**
  - Journey through a cluster installation
- **Summary**

- **Historically (and sometimes still) Linux is installed by hand on each node of the cluster**
  - Hand built clusters may not be efficient to manage but they lend themselves to flexibility
- **Other cluster distributions do already exist! but...**
  - Typically they have traded ease of use with flexibility
  - The quest for ease of use has created overly complex solutions!
- **Problem is not how to create yet another cluster distribution, rather how to manage cluster nodes**

- **Berkeley Lab is a US department of Energy National laboratory that conducts unclassified scientific research**
- **Built from necessity as part of the Scientific Cluster Support (SCS) program at Berkeley Lab intended to promote wide use of cluster computing**
- **The SCS project needed a tool that would allow us to deploy and manage a large number of clusters**

- **Implements a centralized management paradigm for all slave nodes**
- **It differs from other clustering solutions**
  - **Flexibility**
  - **Ease of use**
  - **Lightweight**
  - **RAM-disk based file system support**
- **Warewulf facilitates both customization and scalability**
- **Nothing else like this presently in the community**

- **Supports standard Beowulf architecture**
  - **Master/slave relationship**
    - **Master node(s):**
      - Supports interactive logins and job dispatching to slaves
      - Gateway between outside world and cluster network
      - Central management for all nodes
    - **Slave nodes:**
      - Slave nodes are optimized primarily for computation
      - Only available on private cluster network(s)
  - **Supports multiple physical cluster networks**
    - **Fast Ethernet administrative network**
    - **High speed data networks (Myricom, IB, GigE, bonded, etc...)**

- **Simple and intuitive to use**
  - **Easy to install and maintain for not only moderate level admins but also gifted scientists!**
    - Once familiar with Warewulf, cluster installs should take 1-2 hours for a basic 50 node system
  - **Solution is not over engineered, rather a simplistic approach to a complex problem**
  - **Based on known standards and methods**
- **Modular design that facilitates customization**
  - **Your choice of Kernel, Linux distribution, cluster apps (ie. MPI implementation, etc...)**

- **Designed to be Lightweight**
  - Not over complex thus easy to modify
  - Does not interfere with underlying OS
  - Distribution neutral
- **Does not interfere with standard system administration practices**
  - It is not a Distribution of Linux!
  - Leverages support and tools from the underlying distribution
    - Obtain security patches and updates via distribution vendor
    - Utilizes existing bundled administration tools
    - Standard account management

- **Solution scales both administratively and physically**
  - **Administrative scaling**
    - Gives the admin and users more control of implementation
    - Less wasted time
    - Bad for hourly consultants
  - **Physical scaling**
    - One master node can boot and handle a large number of nodes
    - There are no major anticipated scaling limitations

- **Network booting**
  - Implements RAM-disk file systems (not nfs:/)
  - Boot image is built from the Virtual Node File System (VNFS)
  - Thus no node installations, boots right into OS!
  - Nodes boot utilizing Etherboot
    - OSS project that facilitates network booting
    - Uses DHCP and TFTP to obtain boot image
  - This method of booting is fast!
    - Approx 50 nodes can be booted ready to run jobs in 2 minutes on fast Ethernet

- **RAM Disk based file systems**
  - All nodes are capable of running diskless
  - NFS root is not ideal because of the potential scaling issues and kernel requirements
  - Non-persistent disk images means a clean file system on every boot
  - Node replacement is simplified
  - Need I mention fast?

- **Implements centralized management**
  - **VNFS: The Virtual Node File System**
    - A small chroot'able Linux distribution residing on the master node
    - Allows administrator to modify easily
    - The network boot image is created using the VNFS as a template
    - Destined to live in RAM on the nodes
      - It should be small and tuned
      - Ours is about 50Mb and based on RH73
    - Changes the administration paradigm

- **Implements centralized management (cont)**
  - **Simple user account management**
    - Standard authentication schemes (files, NIS, LDAP)
    - Node access is granted based on standard Unix user and group definitions
    - Warewulf builds a passwd file for all nodes
    - Rsync is used to push files to nodes
- **Shared file system image facilitates troubleshooting**
  - **If a node is having problems, must be hardware related**

- **Installing Warewulf**
  - **Install Linux on the master node**
  - **Download the Warewulf RPMS from <http://warewulf-cluster.org/>**
  - **Install using RPM 'rpm -ivh warewulf-\***
  - **Run some of the Warewulf configuration tools**
    - **Masterconf: Configures the master node config and sets up networking on all devices if needed.**
    - **Nodeconf: Configures the VNFS (thus this should be run after the VNFS has been installed)**
    - **Dhcp-build: Builds the /etc/dhcpd.conf file and restarts dhcpd**
  - **(Re)Start the warewulfd process**
    - **`/etc/rc.d/init.d/warewulfd restart`**

- **Building the VNFS(s)**
  - Defined in `/etc/warewulf/config`
  - Format (`%vnfs`): `name => [path]`
  - Install or create a VNFS at `[path]`
    - Install from rpm, tarball or script at specified path
    - Basically a standard distro with minor boot tweaks
  - Install software into the VNFS
    - ``rpm -ivh --root [path] warewulf-node-*.rpm``
    - `'nodechroot'` brings you into the VNFS
- **Specifying what kernel modules are installed into the VNFS**
  - Edit `/etc/warewulf/config` and specify kernel modules to include in each build

```
root@gmk-clust:/etc/warewulf
File Edit View Terminal Go Help
@modules = qw(
  kernel/fs/jbd/jbd.o
  kernel/fs/ext3/ext3.o
  kernel/drivers/net/3c59x.o
  kernel/drivers/net/3c509.o
  kernel/drivers/net/mii.o
  kernel/drivers/net/eeepro100.o
  kernel/drivers/net/hp100.o
  kernel/drivers/net/pcnet32.o
  kernel/drivers/net/sk98lin/sk98lin.o
  kernel/drivers/net/e100/e100.o
  kernel/drivers/net/e1000/e1000.o
  kernel/net/sunrpc/sunrpc.o
  kernel/fs/lockd/lockd.o
  kernel/fs/nfs/nfs.o
);

%vnfs = (
  warewulf => '/vnfs/warewulf',
);
"config" 86L, 1991C written          61,2          62%
```

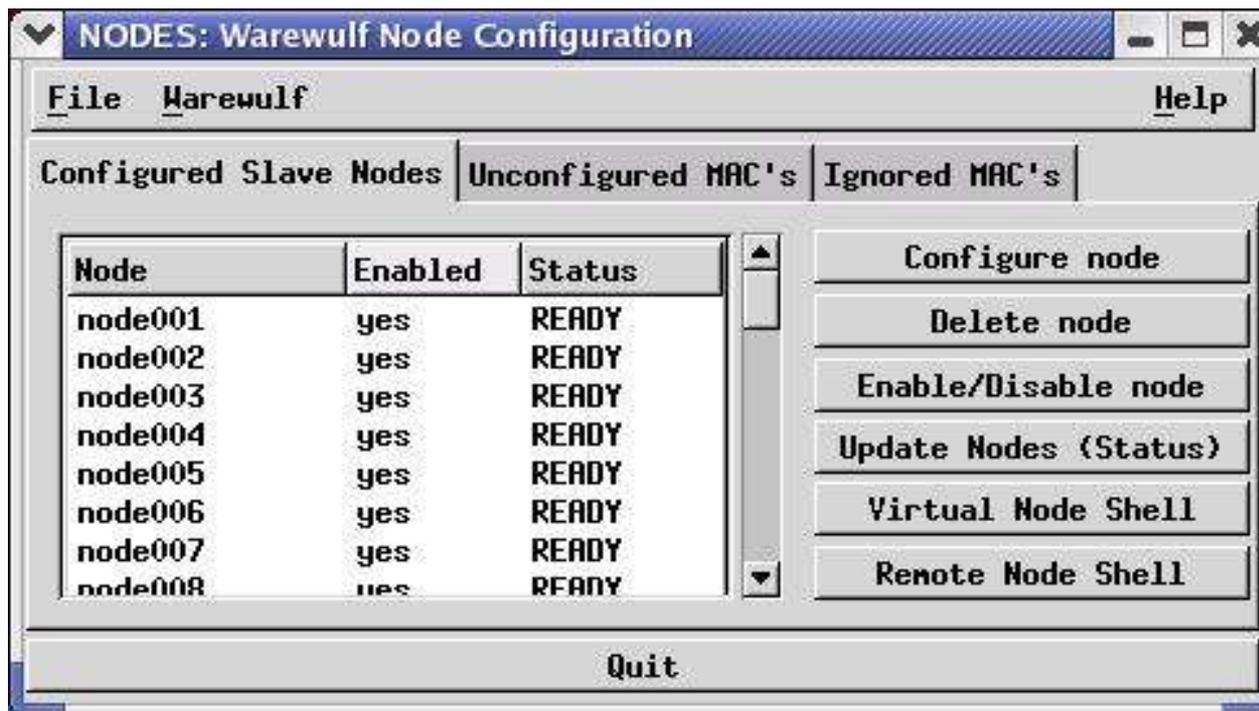
/etc/warewulf/config

Showing the kernel  
module include list  
and the VNFS  
definitions.

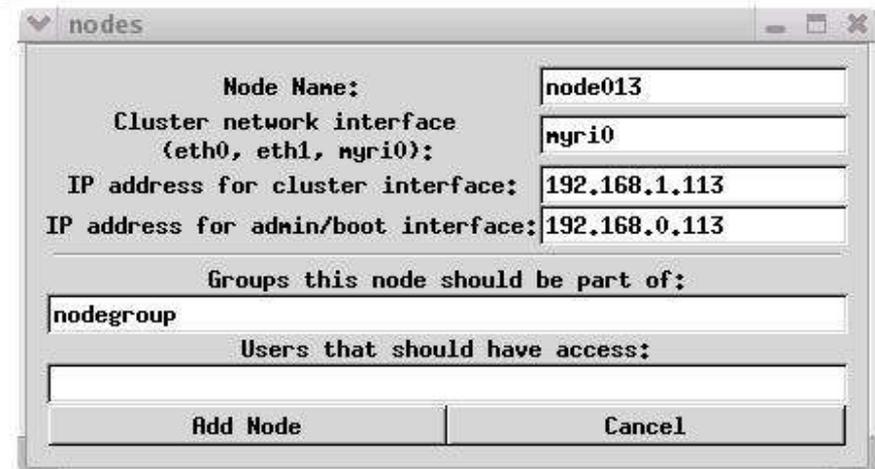
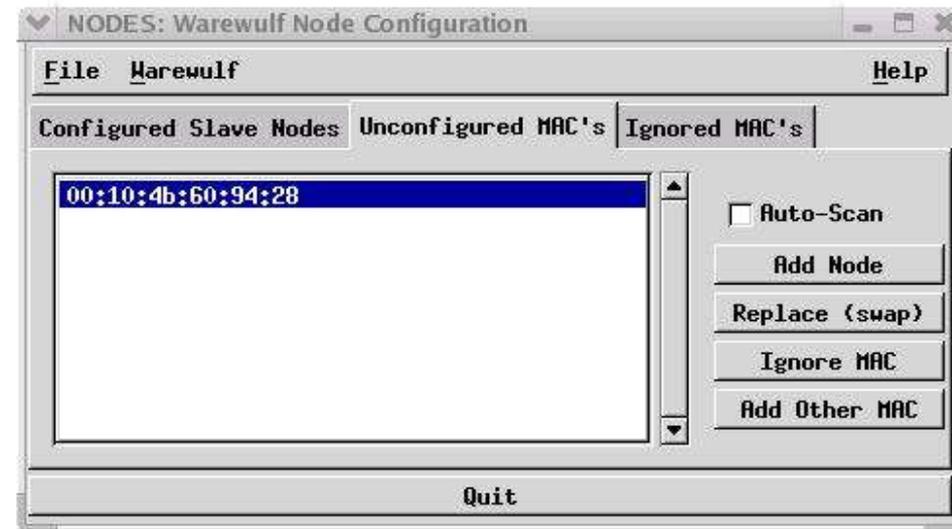
```
root@gmk-clust:~  
File Edit View Terminal Go Help  
[root@gmk-clust root]# nodebuild --net `uname -r`  
-Determining some prelim sizes...  
-Creating image (67612Kb)  
-Formatting image with ext2  
-Migrating /vnfs/warewulf into image  
-Adding kernel modules:  
  /lib/modules/2.4.20-13.7/kernel/fs/jbd/jbd.o  
  /lib/modules/2.4.20-13.7/kernel/fs/ext3/ext3.o  
  /lib/modules/2.4.20-13.7/kernel/drivers/net/3c59x.o  
  /lib/modules/2.4.20-13.7/kernel/drivers/net/3c509.o  
  /lib/modules/2.4.20-13.7/kernel/drivers/net/mii.o  
  /lib/modules/2.4.20-13.7/kernel/drivers/net/eepro100.o  
  /lib/modules/2.4.20-13.7/kernel/drivers/net/hp100.o  
  /lib/modules/2.4.20-13.7/kernel/drivers/net/pcnet32.o  
  /lib/modules/2.4.20-13.7/kernel/drivers/net/sk98lin/sk98lin.o  
  /lib/modules/2.4.20-13.7/kernel/drivers/net/e100/e100.o  
  /lib/modules/2.4.20-13.7/kernel/drivers/net/e1000/e1000.o  
  /lib/modules/2.4.20-13.7/kernel/net/sunrpc/sunrpc.o  
  /lib/modules/2.4.20-13.7/kernel/fs/lockd/lockd.o  
  /lib/modules/2.4.20-13.7/kernel/fs/nfs/nfs.o  
-Building Network boot image  
-Compressing image  
-Merging /boot/vmlinuz-2.4.20-13.7 into /tftpboot/warewulf  
Warning: elf format and --first32pm require Etherboot 5.0 or later  
-Done  
[root@gmk-clust root]#
```

Building the bootable  
node image using  
'nodebuild'.

- Introduction to the node config tool
  - GUI based tool
  - Modifies ASCII config: `'/etc/warewulf/node.conf'`

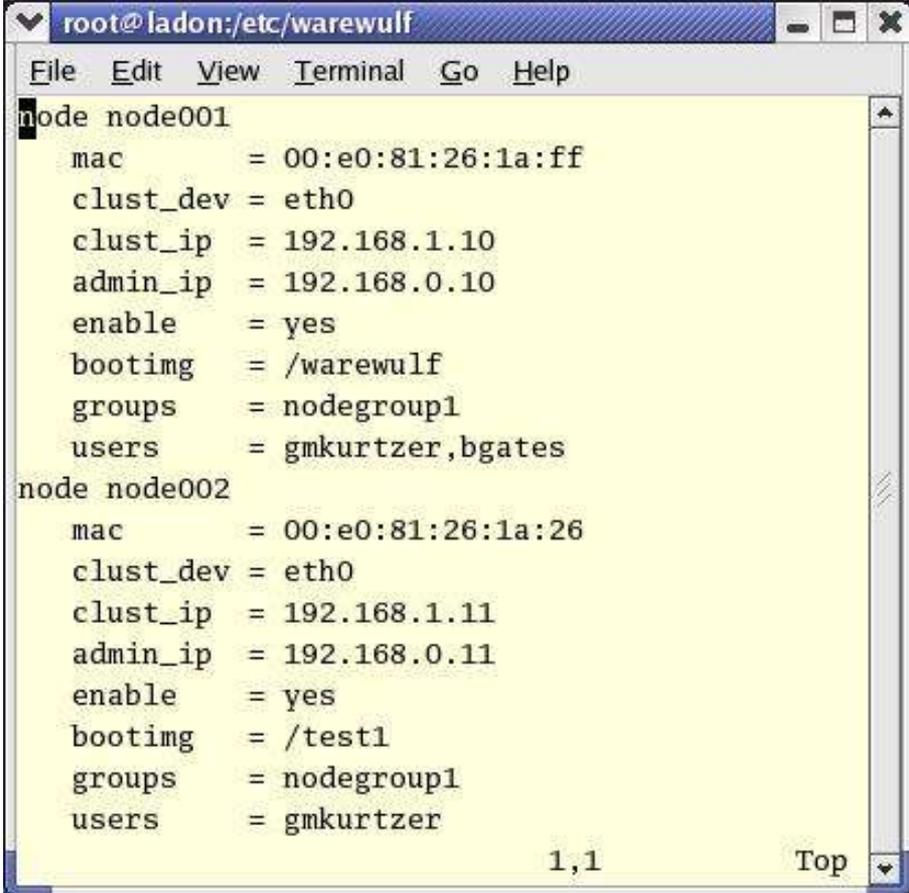


- **Node setup**
  - Boot node with Etherboot
  - Master will recognize unknown DHCP requests and prompt for configuration in tool
  - Configures the node by using the MAC address as the unique identifier



- **Making the nodes usable after booting**
  - There are no persistent user accounts on VNFS except root
  - **Nodeupdate: tool used to make nodes usable**
    - Pushes user entries to node:/etc/passwd
    - Brings the nodes cluster network up
      - Runs myrinet mapper, ifup devX, checks connectivity
    - Runs any misc commands in '/etc/warewulf/node.commands'
      - Starts scheduler exec daemon
    - Puts nodes in 'READY' state

- **User account management**
  - Add user to master node via standard tools (adduser)
  - Add user(s) to specific nodes
  - OR -
  - Add group(s) to specific nodes, then add users to those groups
- **Run 'nodeupdate' to push changes to nodes**



```
root@ladon:/etc/warewulf
File Edit View Terminal Go Help
node node001
  mac      = 00:e0:81:26:1a:ff
  clust_dev = eth0
  clust_ip  = 192.168.1.10
  admin_ip  = 192.168.0.10
  enable    = yes
  booting   = /warewulf
  groups    = nodegroup1
  users     = gmkurtzer,bgates
node node002
  mac      = 00:e0:81:26:1a:26
  clust_dev = eth0
  clust_ip  = 192.168.1.11
  admin_ip  = 192.168.0.11
  enable    = yes
  booting   = /test1
  groups    = nodegroup1
  users     = gmkurtzer
1,1 Top
```

/etc/warewulf/node.conf

- **Installing Warewulf add-ons**
  - Add-ons are standard packages that have been packaged in RPM form in a manner that works well with Warewulf
    - LAM-MPI, SGE, Ganglia, PVM, etc...
    - Can be found from <http://warewulf-cluster.org/>
  - Install the master node component, then the slave node (VNFS) component:
    - 'rpm -ivh package.rpm'
    - 'rpm -ivh --root [vnfs path] package-node.rpm'
  - Rebuild the node image
    - 'nodebuild --net `uname -r`'
  - Reboot all nodes
    - 'wwcommand /sbin/reboot'

- **Warewulf facilitates both customization and scalability**
- **Warewulf is an ideal solution for managing an arbitrary number of similar nodes**
- **Used already on many production clusters throughout the community**
- **Released and maintained by the Berkeley Lab's SCS project under the GPL**
  - **Release intended to encourage contributions and community participation with the project**

**Warewulf - <http://warewulf-cluster.org/>**

**Scientific Cluster Support - <http://scs.lbl.gov/>**

**Berkeley Lab - <http://www.lbl.gov/>**

