

# NHC TUTORIAL

Presented by  
Jacqueline Scoggins  
Lawrence Berkeley National Lab  
Berkeley, California

SchedMD 2014

September 23, 2014



National Laboratory Bringing Science  
Solutions  
to the world

# HISTORY OF NHC

- Written and developed by Michael Jennings at Lawrence Berkeley National Lab starting in 2010.
- NHC is a subproject of the Warewulf system management suite. But warewulf core packages are not required for it to work.
- Originally written for Torque's node health check infrastructure but has expanded to Slurm and other resource managers.
- It is written in Native Bash and has over 40 built-in functions to choose from that can check various components of a nodes health

# BENEFITS OF USING NHC

- Prevent queued jobs from running on unhealthy node(s)
- Works with the resource manager to control the state of a node outside of scheduler related issues – hardware problems, mount issues, misconfiguration of nodes, etc.
- A simple and modular solution that can do multiple checks and report the reason why the node is unhealthy through multiple mechanisms.

# BENEFITS OF USING NHC

(Cont'd)

- ▮ Keeps administrators aware of nodes that are unhealthy
- ▮ Ease of Use
- ▮ Shell scripts – just straight BASH
- ▮ Built-in scripts for immediate use

# CONFIGURING NHC ON A NODE

- ▣ NHC should be installed on every node
  - ▣ Configuration file - `/etc/nhc/nhc.conf` (default)  
(all nodes should have the same copy - no need to create a separate one on each node)
  - ▣ Script files - `/etc/nhc/scripts/*.nhc`
  - ▣ Scheduler/helper scripts - `/usr/libexec/nhc`
  - ▣ Shell script - `/usr/sbin/nhc`
- ▣ Test it locally on a node and then integrate it into the scheduler
  - ▣ Simple test - just run `/usr/sbin/nhc` with no parameters
  - ▣ Review test - view the log file - `/var/log/nhc.log`

# CONFIGURATION FILE - /etc/nhc/nhc.conf

- Field 1: Contains a target specifier
  - Three separate methods for specifying target (nodes)
    - Glob - \*
    - Regexp - `/^n00[0-5][0-9]\.jbs2$/`
    - Node range expressions - `{n00[00-10].clustername}`
- Separator string - `||`
- Field II: Health check directive (or other command)
  - 9 Filesystem check functions
  - 2 Command check functions
  - 3 Files check functions
  - 10 Process check functions
  - Numerous different Hardware check functions

# WHAT ARE SOME OF THE BUILT-IN CHECKS?

- ▢ Command Execution – run any command and check the results

```
# check within 1 second and return the status from service command
```

```
check_cmd_output -t 1 -r 0 -m '/is running/' /sbin/service  
rpcbind status
```

- ▢ Filesystems – checks various things like is it mounted, are the permissions correct, inode count, space consumed or free

```
# checks that root is mounted read-write
```

```
check_fs_mount_rw -f /
```

# WHAT ARE SOME OF THE BUILT-IN CHECKS?

- Hardware – cpu, ethernet, infiniband, nvidia, myrinet, mcelogs, memory, swap, load average

```
# check that eth0 is available  
check_hw_eth eth0
```

- Processes – blacklist, cpu consumed, daemon, memory usage, existence (status), time used, unauthorized users (user not running job on the node\*\*), rogue processes

```
# check that sshd is running as the root user and start it if it is not (-S)  
check_ps_service -S -u root sshd
```

\*\* only works for single node jobs.

# EXAMPLE NHC.CONF

```
* || export PATH="$PATH:/opt/nv/bin"
* || check_fs_mount_rw -t '/(ext4|tmpfs)/' -f / -f /tmp -f /local
* || check_fs_mount_rw -s '/^nas:\/(home|soft)/' -f /home -f /soft
* || check_fs_mount_ro -o '/(^|,)vers=4(,|$)/' -f /var/chroots
* || check_ps_service -u root -S sshd
* || check_ps_service -u root -r crond
* || check_ps_service -u daemon -d munged -r munge
* || check_file_test -r -w -x -d -k /tmp /var/tmp
* || check_file_test -c -r -w /dev/null /dev/zero
* || check_file_contents /etc/passwd "/^root:x:0:0:/" "sshd:*"
node* || check_ps_service -u root -d slurmd -r slurm
node* || check_ps_userproc_lineage log syslog
node* || check_hw_cpufreq 2 20 20
node* || check_hw_physmem 64GB 64GB 3%
node* || [ -x "$NVIDIA_HEALTHMON" ] && check_nv_healthmon || continue
master* || check_fs_free / 10%
master* || check_fs_used /home 97%
master* || check_fs_ifree /biodata 20k
master* || check_ps_service -u slurm -d slurmd -r slurm
master* || check_ps_service -u slurm -r slurmdbd
master* || check_hw_cpufreq 2 12 12
master* || check_hw_physmem 32GB 32GB 3%
master* || check_hw_swap 18GB 18GB 3%
master* || check_cmd_output -m '/10\.0\.0\.1/' -m '/eth1.*,UP/' ip addr
master* || check_dmi_data_match -t "BIOS Information" "/Revision: 4\[3-7]/"
```

# HOW IS IT INTEGRATED IN THE SCHEDULER?

- ▢ The scheduler (SLURM) is set up to use NHC by setting the parameter in `/etc/slurm/slurm.conf`
  - ▢ `HealthCheckProgram` = location of the script to execute the checks (default: `/usr/sbin/nhc`)
  - ▢ `HealthCheckInterval` = seconds (the number of seconds between executions of the health check program)
  - ▢ `HealthCheckNodeState` = state of the node to execute the script on

# SCHEDULER INTERACTION

- ▮ NHC is configured to know which Resource Manager the system is running and which commands to run to communicate with the scheduler
- ▮ Parameters settings and location
  - ▮ `nhc.conf` - `export NHC_RM=slurm` (should not be needed)
  - ▮ `node-mark-[off|on]line` - 2 variables
    - ▮ `SLURM_SINFO` (`sinfo`)
    - ▮ `SLURM_SCONTROL` (`scontrol`)

# NHC INTERACTIONS VIA THE SCHEDULER

- ▣ sinfo is used to check the existing state of the node before performing any Actions
- ▣ Current node message is read to know if NHC already took the node offline or if the admin had the node offline (sinfo -o '%t %E' -hn nodename)
- ▣ Parameters defined in (/etc/nhc/nhc.conf)  
\$IGNORE\_EMPTY\_NOTE
  - 1 - do not ignore** (we choose this because all offline nodes need to have a message as to why they were offlined)
  - 0 - ignore (default)

# NHC INTERACTIONS VIA THE SCHEDULER

- ▣ By default, NHC will automatically run node-mark-offline if the node fails a check, or node-mark-online if the node passes all the checks.
- ▣ Each helper script determines what if any commands to run.
- ▣ The variable `$STATUS` contains the current state of a node, and based on its value certain actions can occur to a node via the scheduler.
- ▣ The following are the node states which are valid in slurm - down, idle, drain, alloc, resume, undrain, offline, unknown, comp, failed, mixed

# NHC INTERACTIONS VIA THE SCHEDULER

## SLURM ACTIONS

***scontrol update State=DRAIN NodeName=hostname Reason="NHC: failure error message"***

if node is already down by NHC or any reason do nothing  
if node is already offline by Admin preserve the message  
if node is online but unhealthy mark it offline and report the reason

***scontrol update State=Resume NodeName=hostname***

if node is down by NHC and the error is fixed bring it back online and remove the Reason  
if node is down by admin leave it down and do nothing

# HINTS FOR WRITING CHECKS

- Whenever possible, use native bash constructs

- Read Name < /some/file instead of NAME=`cat file`

```
read HOSTNAME < /proc/sys/kernel/hostname
```

- `${VAR/[/]pattern/replacement}` instead of sed

```
HOSTNAME_S="${HOSTNAME/%.*}"
```

```
CLUSTERNAME="${HOSTNAME/#$HOSTNAME_S.}"
```

```
i.e. - Hostname=n0000.jar.gov Hostname_S = n0000 Clustername=jar.gov
```

- Read/split data using IFS and arrays instead of awk

- Redirection to loops instead of pipelines

```
IFS=':'
```

```
while read -a LINE; do
```

```
  PW_USER="${LINE[0]}"
```

```
  PW_UID="${LINE[1]}"
```

```
done < /etc/passwd
```

# COMING SOON

- ▣ `nhc-wrapper`
  - ▣ Run `nhc` in a loop with a specified interval; and/or,
  - ▣ Notify via e-mail on failure or when failure is cleared; and/or,
  - ▣ Cache results from previous NHC run and expire cache after specified time period
- ▣ `nhc-genconf`
  - ▣ Uses built-in NHC information-gathering functions to scan system
  - ▣ Generates configuration file based on hardware, filesystems, etc. found on system
  - ▣ Customizable hostmask for generated configuration file for use with, e.g., `dshbak`
- ▣ More checks: SLURM diags, sockets/open files, “rolling” reboots/scripts

# WAREWULF NHC - QUICK START

DOWNLOAD and INSTALL NHC RPM

```
http://warewulf.lbl.gov/downloads/releases/warewulf-nhc/
```

```
rpm -ivh warewulf-nhc-1.4.el6.noarch.rpm  
(prepackaged for RHEL5 and 6)  
test NHC execution and command
```

```
# nhc  
# echo $?  
# cat /var/log/nhc.log
```

Configure SLURM to run NHC

```
# vi /etc/slurm/slurm.conf  
HealthCheckProgram=/usr/sbin/nhc  
HealthCheckInterval=300  
HealthCheckNodeState=ANY  
# /etc/init.d/slurm restart
```

# CONTACT INFORMATION

WareWulf NHC

[warewulf-devel@lbl.gov](mailto:warewulf-devel@lbl.gov)

[mej@lbl.gov](mailto:mej@lbl.gov) (Developer)

<http://go.lbl.gov/nhc> (documentation)

Jacqueline Scoggins

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

(510) 486-8651

QUESTIONS?



# ANSWERS TO NHC QUESTIONS YESTERDAY

1. How to prevent an offline node from coming back online if you wish to leave it down

\* `|| ONLINE_NODE==: check_hw_ib 56 ib0`

2. Statistics

No statistics are gathered but it's in the thousands of times, if not tens of thousands

3. Epilogue scripts

If a node goes down it requires manual intervention to restore the node to service. This mechanism is also incompatible with detach mode (which is what nhc is about) It can be used in this manner but we choose not to do it this way.