



# **C3: Cluster Power Tools**

## **Cluster Command & Control**

Presented by: Brian Luethke

Brian Luethke, John Muggler, Thomas Naughton, Stephen Scott

Commodity, High-Performance Cluster Computing Technologies and Applications  
The Sixth World Multiconference on Systemics, Cybernetics, and Informatics.

July 16, 2002 – Orlando, Florida

# Overview

- **command line based**
- **single system illusion (SSi) – single machine interface**
- **cluster configuration file**
- **ability to rapidly deploy from server – software and system images**
- **command line list option – enable subcluster management**
- **distributed file scatter and gather operations**
- **execution of non-interactive commands**
- **multiple cluster capability – from single entry point**

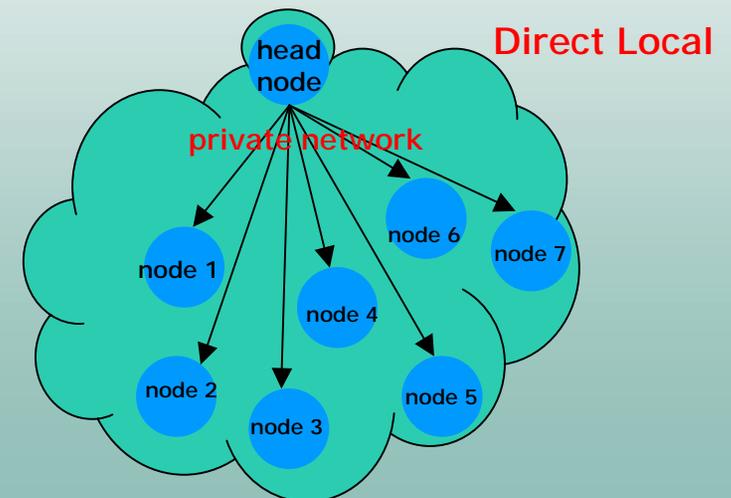
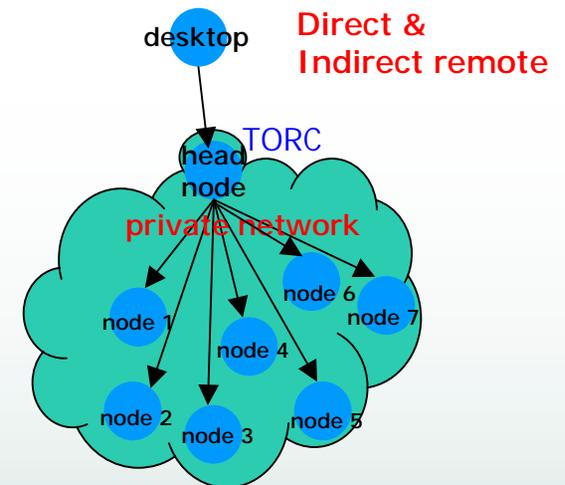
## Building Blocks

- System administration
  - **cpushimage** - “push” image across cluster
  - **cshutdown** - Remote shutdown to reboot or halt cluster
- User tools
  - **cpush** - push single file -to- directory
  - **crm** - delete single file -to- directory
  - **cget** - retrieve files from each node
  - **ckill** - kill a process on each node
  - **cexec** - execute arbitrary command on each node
    - **cexecs** – serial mode, useful for debugging
  - **clist** – list each cluster available and it’s type
  - **cname** – returns a node position from a given node name
  - **cnum** – returns a node name from a given node position

# Cluster Classification Scheme

- Direct local
  - The cluster nodes are known at run time
  - The command is run from the head node
- Direct remote
  - The cluster nodes are known at run time
  - The command is not run from the head node
- Indirect remote
  - The cluster nodes are not known at run time
  - The command is not run from the head node

- Notes:
  - Local or remote is checked by comparing the head node names to the local hostname
  - Indirect clusters will execute on the default cluster of the head node specified.



# Cluster Configuration File

- default cluster configuration file

- `/etc/c3.conf`

```
Cluster torc { #direct local cluster
    orc-00b:node0
    node[1-4]
    exclude 3
}
Cluster htorc { #indirect remote cluster
    :htorc-00
}
```

- user specified configuration file

- `/...somewhere/list_of_nodes`

```
Cluster auto-gen { #direct remote cluster
    node0.csm.ornl.gov
    node1.csm.ornl.gov
    node2.csm.ornl.gov
    node3.csm.ornl.gov
    dead node4.csm.ornl.gov
}
```

# Configuration File Information

- Offline Node Specifier
  - **Exclude** tag applies to ranges
  - **Dead** applies to single machines
  - Important for node ranges on the command line
- Cluster Definition Blocks as Meta-clusters
  - Group based on hardware
  - Groups based on software
  - Groups based on role
- User specified cluster configuration files
  - Specified at runtime
  - User can create both sub-clusters and super-clusters
  - Useful for scripting
- Can not have a indirect local cluster (info has to be somewhere...)
- Infinite loop warning: When using a indirect remote cluster, the default cluster on the remote head node is executed. This could make a call back.

# MACHINE DEFINITIONS (Ranges) on Command Line

- **[MACHINE DEFINITIONS]** as used in command line
- Position number from configuration file
  - Begin at 0
  - Does not include head node
  - **dead** and **exclude** maintain a nodes position
- Format on command line
  - First cluster name from configuration file with a colon
    - **Cluster2**: would represent all nodes on cluster2
    - **:** signifies default cluster
  - ranges and single nodes are separated by a comma
    - **Cluster2:1-5,7** executes on nodes 1, 2, 3, 4, 5, 7
    - **:4** executes node at position 4 on the default cluster
  - **cexec : torc:1-5,7 hostname**

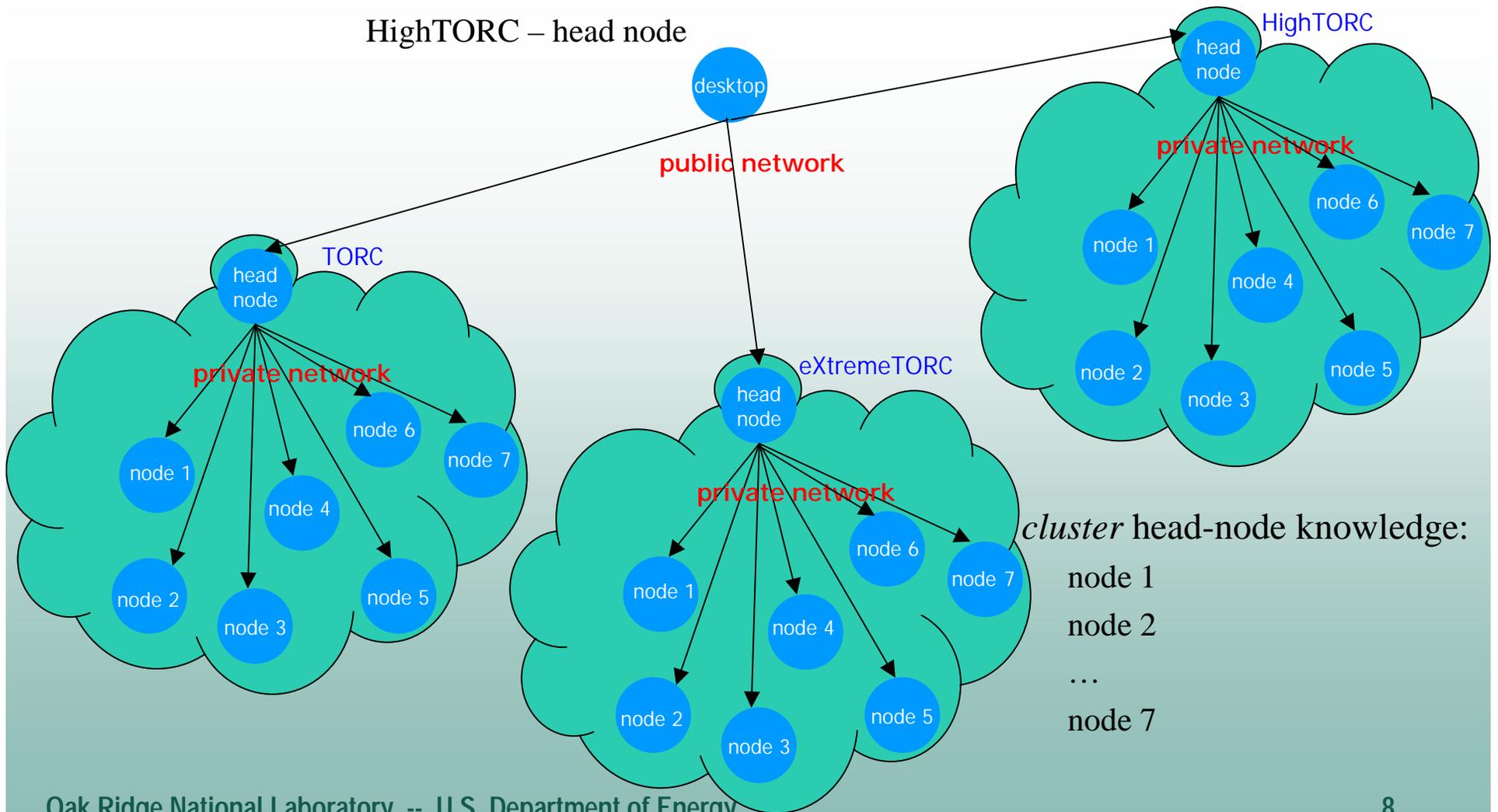
# Execution Model: External to Multi-Cluster

desktop knowledge:

TORC – head node

eXtremeTORC – head node

HighTORC – head node



# Execution Model: External to Multi-Cluster

On desktop

```
Indirect remotes (several in one file)
-----
cluster torc {
    :torc
}
cluster exterme_torc {
    :xtorc
}
cluster high_torc {
    :htorc
}
```

On eXtremeTORC

```
Direct local
-----
cluster xtorc {
    xtorc:node0
    node[1-7]
}
```



# cpush

**cpush [OPTIONS] [MACHINE DEFINITIONS] source [target]**

- h, --help** display help message
- f, --file <filename>** alternate cluster configuration file, default is /etc/c3.conf
- l, --list <filename>** list of files to push (single file per line, column1=SRC column2=DEST)
- i** interactive mode, ask once before executing
- head** execute command on head node, does not execute on compute nodes
- nolocal** the source file or directory lies on the head node of the remote cluster
- b, --blind** pushes the entire file (normally cpush uses rsync)

## **cpush**

- to move a single file

```
cpush /home/filename
```

This pushes the file "filename" to /home on each compute node

- to move a single file, renaming it on the cluster nodes

```
cpush /home/filename1 /home/filename2
```

Push the file "filename1" to each compute node in the cluster, renaming it to filename2 on the cluster nodes

- to move a set of files listed in a file

```
cpush --list=/home/filelist escaflowne:
```

This pushes each file in the filelist where it is specified to send it. Filelist format is on the next slide.

## Notes on using a file list

- One file per line
- If no destination is specified then it will push the file to the location it is on the local machine
- No comments
- Example file
  - `/home/filename`
  - `/home/filename2 /tmp`
  - `/home/filename3 /tmp/filename4`

The first line pushes the file "filename" to /home on each compute node

The second line pushes the file "filename2" to /tmp on each compute node

The third line pushes the file "filename3" to /tmp on each compute node renaming the file to "filename4"

- All options on the command line are applied to each file, In a filelist, you can not specify that file one uses the `-nolocal` option and file two goes to the machine definition clusters:3-5.

## cexec

Usage: cexec(s) [OPTIONS] [MACHINE\_DEFINITIONS] command

<b>--help -h</b>	display help message
<b>--file -f &lt;filename&gt;</b>	alternate cluster configuration file if one is not supplied then <i>/etc/c3.conf</i> will be used
<b>-i</b>	interactive mode, ask once before executing
<b>--head</b>	execute command on head node, does not execute on the cluster

Using **cexecs** executes the serial version of cexec

## **cexec**

- to simply execute a command

**cexec mkdir temp**

This executes mkdir temp on each node in the cluster. The working directory of the cexec command is always your home directory thus temp would be created in ~/

- to print the machine name and then execute the string  
( **serial version only** )

**cexecs hostname**

This executes hostname on each node in the cluster. This differs from cexec in that each node is executed before the next one. This is useful if a node is offline and you wish to see which one.

## cexec

- to execute a command with wildcards on several clusters

```
cexec cluster1: cluster2:2-5 "ls /tmp/pvmd*"
```

This will execute "ls /tmp/pvmd\*" on each compute node on cluster one and nodes 2, 3, 4, and 5 on cluster2. Notice the use of the quotes. This keeps the shell from interpreting the command until it reaches the compute nodes.

- Using pipes

```
cexec "ps -A |grep a.out"  
cexec ps -A |grep a.out
```

In the first example the | symbol is enclosed in the quotes. In this case "ps -A|grep a.out" is executed on each node. In this way you get the standard cexec output format with a.out in each nodes block if it exists. In the second example "ps -A" is executed on each node and the all the a.out lines are grep'ed out. This demonstrates that placement of ""s is very important. Example output on next slide.

# cexec quotation example

cexec "ps -A|grep xinetd"

```
***** local *****
processing node node1
***** local *****
processing node node2
***** local *****
----- node1-----
9738 ?    00:00:00 xinetd

----- node2-----
4856 ?    00:00:00 xinetd
```

cexec ps -A |grep xinetd

```
9738 ?    00:00:00 xinetd
4856 ?    00:00:00 xinetd
```

# cname

Usage: `cname [OPTIONS] [MACHINE DEFINITIONS]`

`--help -h` display help message  
`--file -f <filename>` alternate cluster configuration file if  
one is not supplied then */etc/c3.conf* will be used

## **cname**

- To search the default cluster

**cname :0-5**

This returns the node name for the nodes occupying slots 0, 1, 2, 3, 4, and 5 in the default configuration file

- To search a specific cluster

**cname cluster1: cluster2:4-8**

All of the nodes in cluster1 are returned and nodes 4, 5, 6, 7, and 8 are returned from cluster2

# cnum

Usage: cnum [OPTIONS] [MACHINE DEFINITIONS] node\_name

**--help -h**            display help message  
**--file -f <filename>**    alternate cluster configuration file if  
                         one is not supplied then */etc/c3.conf* will be used

## **cnum**

- To search the default cluster

**cnum node2**

This returns the node position (number) that node2 occupies in the default cluster configuration file

- To search several clusters in the configuration file

**cnum cluster1: cluster2: gundam eva**

This returns the node position that the nodes gundam and eva occupy in both cluster1 and cluster2. If the node does not exist in the cluster node number is returned.

# clist

Usage: `clist [OPTIONS]`

`--help -h` display help message  
`--file -f <filename>` alternate cluster configuration file if one is not supplied then */etc/c3.conf* is used

# clist

- To list all the clusters from the default configuration file

## clist

This lists each cluster in the default configuration file and its type(direct local, direct remote, or indirect remote)

- To list all the clusters from an alternate file

## clist -f cluster.conf

This lists each cluster in the specified configuration file and its type(direct local, direct remote, or indirect remote)

## Multiple cluster examples

- Command line: Same as single clusters, only specify several clusters
  - example
    - installing and rpm on two clusters
      - First push rpm out to cluster nodes  
`cpush : xtorc: example-1.0-1.rpm`
      - Use RPM to install application  
`cexec : xtorc: rpm -i example-1.0-1.rpm`
      - Check for errors in installation  
`cexec : xtorc: rpm -q example`
    - Notice the addition of `:" xtorc:"` cluster specifier – only difference between examples
      - All clusters in this list will participate in this command (the standalone `:"` represents the default cluster)

## Usage Notes

- By default C3 does not execute commands on the head node
  - Use `-head` option to execute only on the head node
- Interactive option only asks once before execution
- Commands only need to be homogeneous within itself
  - Example: binary and data on an intel and HPUX
  - Data can be pushed to both systems  
`cpush -head intel: hp: data.txt`
  - Binary for each cluster  
`cpush -head intel: app.intel app`  
`cpush -head hp: app.HPUX app`
  - Then execute app  
`cexec -head intel: hp: app`

## Usage Notes

- Notes on using multiple clusters
  - **Very powerful, but with power comes danger...**
    - malformed commands can be VERY bad
      - "homogeneous within its self" becomes very important
      - `crm -all *` could bring down MANY nodes
      - Extend nearly all unix/linux gotcha's to multiple clusters/many nodes – and very fast
  - High level administrators can easily set policies on several clusters from single access point.
    - Federated clusters – those within single domain
    - Meta-clusters – wide area joined clusters

## Contact Information

[torc@msr.csm.ornl.gov](mailto:torc@msr.csm.ornl.gov)

contact ORNL cluster team

[www.csm.ornl.gov/torc/C3](http://www.csm.ornl.gov/torc/C3)

version 3.1 (current release)

[www.csm.ornl.gov/TORC](http://www.csm.ornl.gov/TORC)

ORNL team site

[www.openclustergroup.org](http://www.openclustergroup.org)

C3 v3.1 included in OSCAR 1.3