ClusterShell

Overview

Contents

- `nodeset` tool
- Node Groups
- `clush` tool
- ClusterShell Python API
- Next steps

nodeset tool

- **Perform nodeset expansion/folding**
  - Fold host names to nodeset
    - `$ nodeset -f foo3 foo7 foo9 foo8`
      `foo[3,7-9]`
  - Expand nodeset(s) to host names
    - `$ for n in $(nodeset -e foo[3,7-9]) ; do`
      `./do_something_with $n`
      `done`
- **And much more advanced operations**
  - `$ nodeset -f foo[3,7-10]^foo[2-10/2]`
    `foo[2-4,6-7,9]`
  - Also supporting ranges only, splitting, slicing nodeset, ...

ClusterShell node groups

- **A node group is a nodeset**
  - Special notation starting with @
    - `@groupname`
  - Several group namespaces can be setup: **Group Sources**
    - Notation : `@source:groupname`
    - Simple setup : Group Source callbacks as shell commands
- **Nodeset arithmetic applies to node groups**
  - `$ nodeset -f @gpu^@slurm:bigmem!@rack[1-9/2]`
- **List available groups**
  - `$ nodeset -l (or -ll)`
- **Support reverse resolution (nodes to groups)**
  - `$ nodeset -r foo[1-101]`
    `@rack[0-7],foo101`

clush tool

- **Pdsh-like tool based on ClusterShell library**
- **Valued features**
  - Integrated output gathering (options -b/-B)
  - Buffered write support on stdin
  - Support nodeset arithmetic notations and node groups
- ***Usage example:***
  - ```
    $ clush -br -w callisto[32-157] uname -r
    ---------------
    @rack[1-7] (126)
    ---------------
    2.6.18-164.11.1.el5
    ```
    ```

ClusterShell Python API

- **NodeSet class: set of indexed node names**
- **Task: main class of ClusterShell commands execution support**
- **Two coding/usage models are available**
  - sequential mode
  - event-based mode: the application derives the `EventHandler` class to listen for events
- ***Simple example of use:***

```
from ClusterShell.Task import task_self, NodeSet

task = task_self()

task.run("/bin/uname -r", nodes="callisto[36-39,133]")

for output, nodes in task.iter_buffers():
    print NodeSet.fromlist(nodes), output
```

Next steps

- **ClusterShell v2**
  - Tree-based remote command propagation engine
  - Beta version is working
- **Node Groups**
  - GroupSource mapping upcalls available as Python plugin
- **New API in ClusterShell v3**
  - Even more easy to use, object-based API on top of current API
  - High level design set up

Questions ?

http://clustershell.sourceforge.net/

clustershell-devel@lists.sourceforge.net