EMSL

Pacific Northwest NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY

The Road to DevOps HPC Cluster Management

Ken Schmidt, Evan Felix

Nov 22, 2019

SCIENTIFIC INNOVATION THROUGH INTEGRATION

Short EMSL Computing History

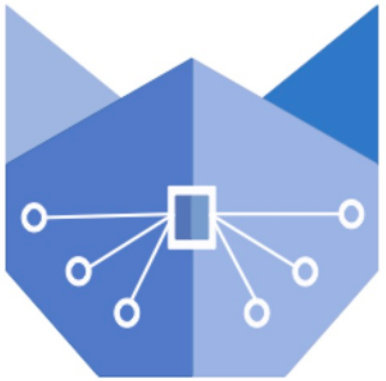
- MPP2 – First Generation Cluster
  - One of the first Linux Clusters in the top 5
  - One of the earliest deployments Lustre
  - ~1000 nodes – How do you install it?
  - More than a year for deployment and acceptance
- Chinook – Second Generation
  - Let's do it again – only bigger!
  - Deployment took months
  - Time to strike out on our own
    - Drop vendor product
    - Use open source tools – xCAT

Next Cluster

- Cascade – Third Generation
  - Hardware only procurement
  - Same configuration repository
  - Deployed with Scientific Linux 6
  - From delivery to acceptance to full production in weeks
  - Upgraded to CentOS 7 without user problems

Open Source advantages

- xCAT
  - Source is available to modify and contribute back changes
  - Decouple hardware, operating system, and management software
- Git
  - Track all changes made to xCAT database
  - Store all configurations and scripts
  - Facilitate collaboration between administrators
- Gitlab
  - Facilitate collaboration between administrators
  - Track work being done
  - Store Documentation for new/unfamiliar administrators



Monthly Outages

- Every month take an outage
  - Makes the system more up-to-date for the users
  - Reduces the number of unplanned outages
  - Gives administrators full unfettered access to make changes
  - Drives development cycles
  - ~1.5% yearly downtime

Weekly Meetings

- Week 1
  - What work was completed during the last outage? – outage minutes
  - What went wrong that needs to be changed?
  - What can be done better?
- Week 2
  - Plan the tasks/priorities for up upcoming outage
- Week 3
  - Evaluate what will be complete by outage
  - Schedule outage (plan resources & check for interdependencies)
- Week 4
  - Burn the cluster to the ground and build it up again

Gitlab

- Issue Tracking
  - All work is tracked in Gitlab using issues
  - Progress is tracked to determine what will and what won’t go into the upcoming outage
  - Admin/Developer only use
- Documentation
  - All procedures are documented
  - They are never complete and need almost constant updating
  - Gitlab provides easy web and markdown file wikis
  - Procedures are occasionally tested by outside driver outages

Disadvantages of Traditional Management

- “Organically” Grown – Install a system and make changes as needed
  - Upgrades are near impossible
  - “How did we get here?”
  - “How do we upgrade?”
  - “How do we prevent user interruption?”

Working with (around) xCAT

- Automated table backups checked into git
- Difficult to test changes on test cluster and merge into production
- Move configurations out of xCAT's databases into revision control
- Use glue script to jump from xCAT config into scripts stored in revision control

```
#!/bin/sh

set -x # Log everything the script is doing
scriptdir=$1
shift

for i in $scriptdir/* # For every file in the directory passed on the command line.
do
        if [ -x $i ] # Make sure the script is executable
        then
                if ! $i $* # Execute postscript and check the results
                then
                        set +x                                  # If the script fails,
                        echo "Error executing $i postscript" # log the error
                        exit 1                                  # and bail
                        set -x
                fi
        fi
done
```

```
.
..
00_savenodeinfo -> /install/postscripts/msc/custom/savenodeinfo
01_configics -> /install/postscripts/configics
05_mkresolvconf -> /install/postscripts/mkresolvconf
10_syncfiles -> /install/postscripts/syncfiles
15_configxcathost -> /install/postscripts/configxcathost
20_setup_sssd -> /install/postscripts/msc/custom/setup_sssd
30_hardeths -> /install/postscripts/hardeths
40_rsyslog -> /install/postscripts/msc/custom/rsyslog
50_remoteshell -> /install/postscripts/msc/custom/remoteshell
60_mountscratch -> /install/postscripts/msc/custom/mountscratch
70_rc-local -> /install/postscripts/msc/custom/rc-local
80_hugepages -> /install/postscripts/hugepages
```

MSC's Implementation of Cluster as Code

- All configurations are checked into repositories and are checked out minutely on the clusters' management nodes
  - Any change attempted outside git are stashed and removed
- Any change gets committed to the development branch and is tested on test cluster
  - Prior to outages, development is merged into production
- Easily track changes to configurations and can track down when/why the change was made
- Example on github - https://github.com/EMSL-MSC/MscClusterMgmt

Next Cluster

- New Cluster – Next Generation
  - Hardware only procurement
  - Will be deployed with CentOS 7
  - Planned upgrade to CentOS 8 in its lifetime
  - Same configuration repository
  - Expected smooth transition based on past performance

Questions?

The research was performed using EMSL (grid.436923.9), a DOE Office of Science User Facility sponsored by the Office of Biological and Environmental Research.

kenneth.schmidt@pnnl.gov

www.emsl.pnnl.gov

EMSL

Pacific Northwest NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY