Scalable High Availability for Lustre with Pacemaker

LUG17

Christopher J. Morrone

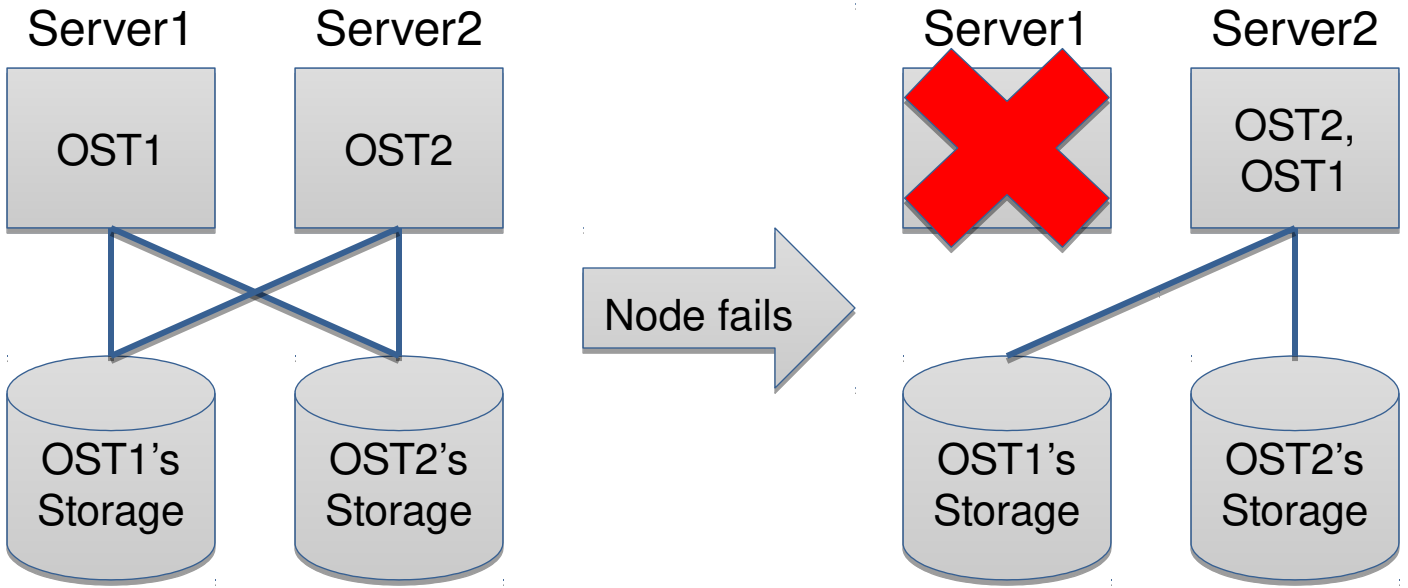
June 1, 2017

LLNL-PRES-731486

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

Lustre High Availability



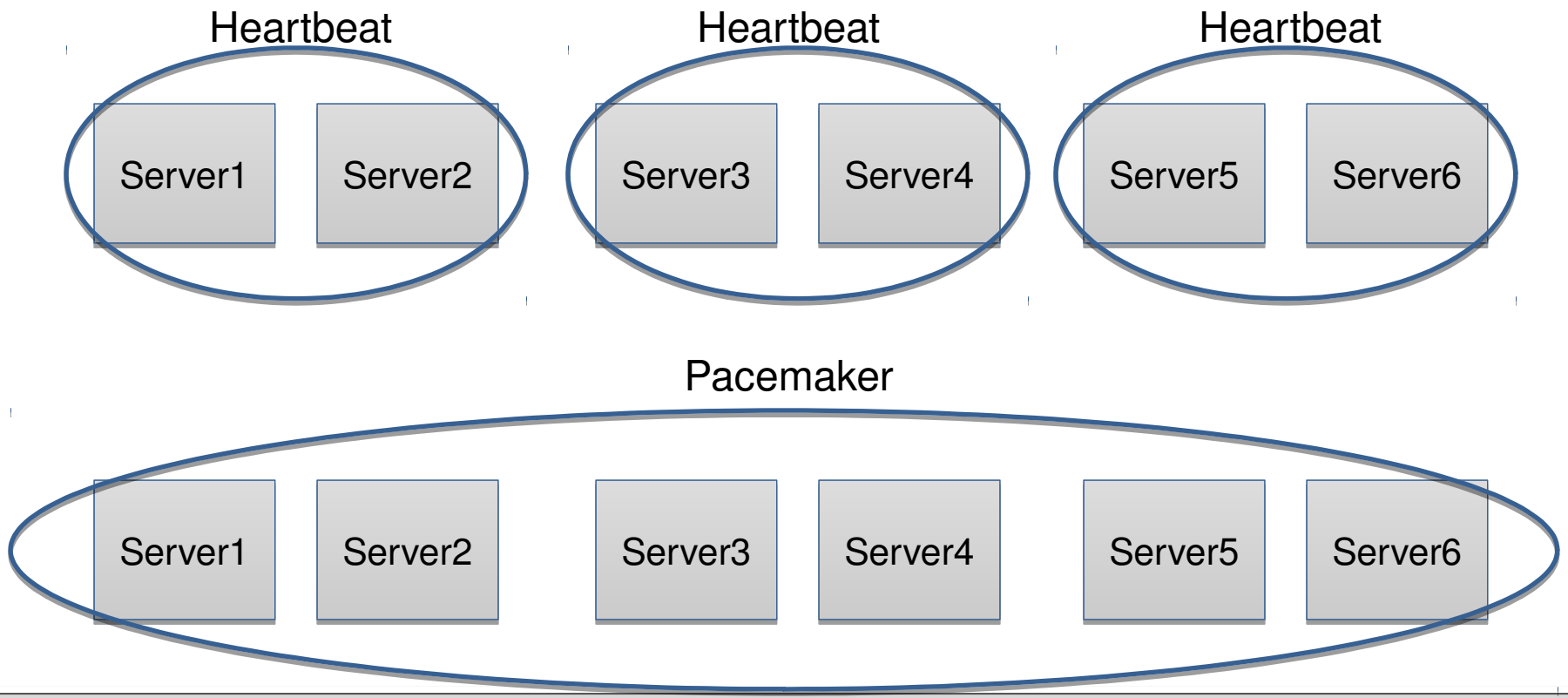
Motivation: Migrate from Heartbeat to Pacemaker

- Heartbeat missing from RHEL7
- Official RHEL7 HA stack
  - Pacemaker 1.1
  - Corosync 2.X
  - PCS

HA Stack Components/Terms

- Pacemaker – Resource manager
- Corosync – Messaging layer, quorum
- pcs – Unified Pacemaker/Corosync command shell
- Resource Agent (RA) – Script/program interface to a single resource type
- Fence Agent (FA) – STONITH device driver (script)

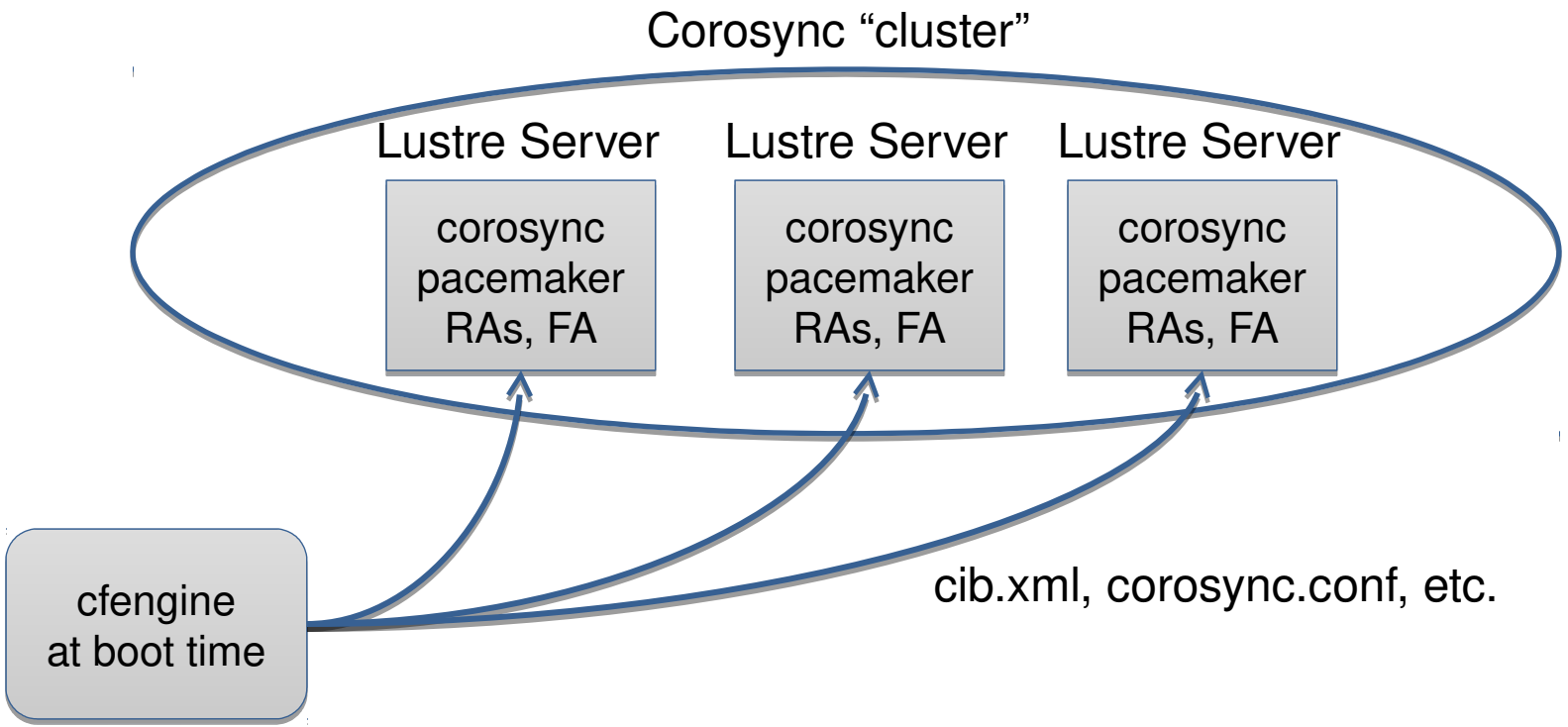
Single Pacemaker Cluster



LLNL Constraint: Stateless Servers

- Issue – Pacemaker assumes unique local storage
- Issue – Pacemaker cib.xml direct edits forbidden (use cibadmin/pcs)
- LLNL configuration through cfengine

The Wrong Approach: Script-Generated cib.xml, corosync.conf



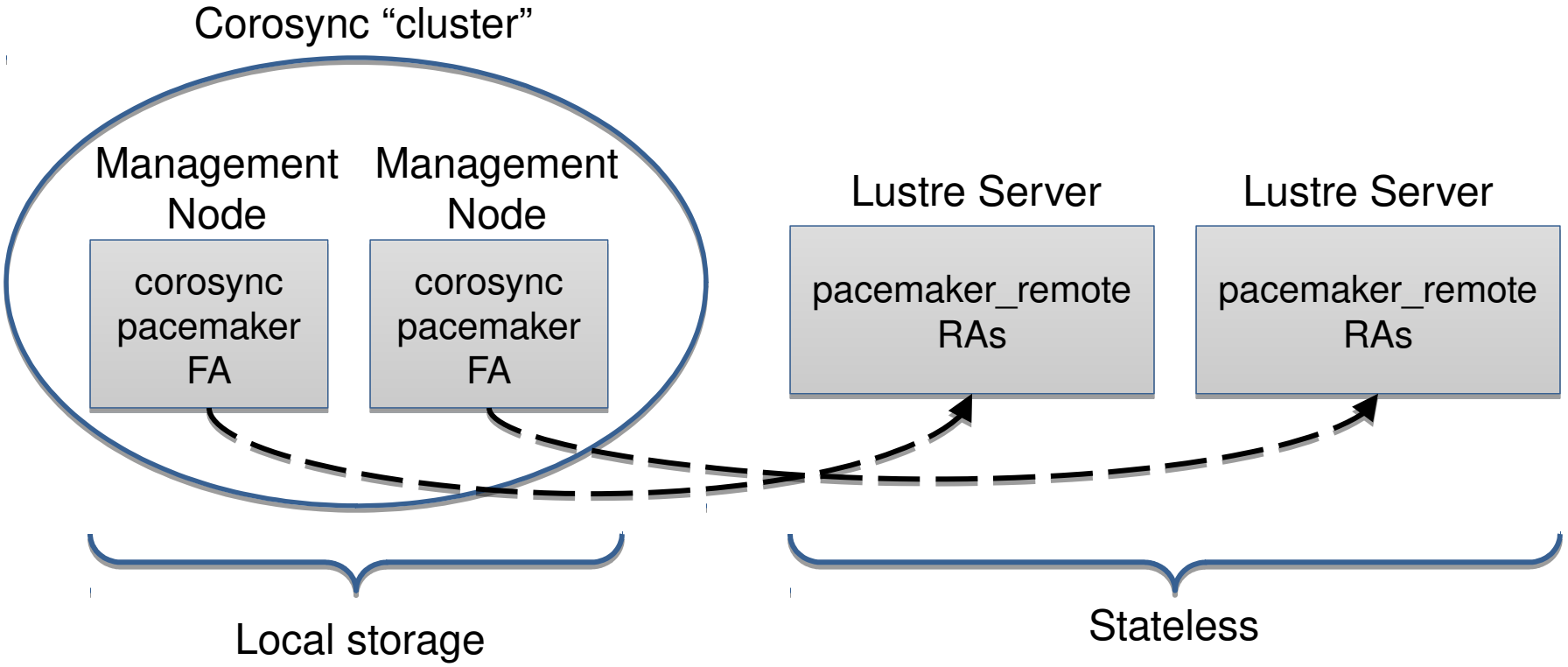
Issue: Corosync Does Not Scale

- Rule of thumb: 16 node corosync cluster limit
- RH support ends at 16 corosync nodes

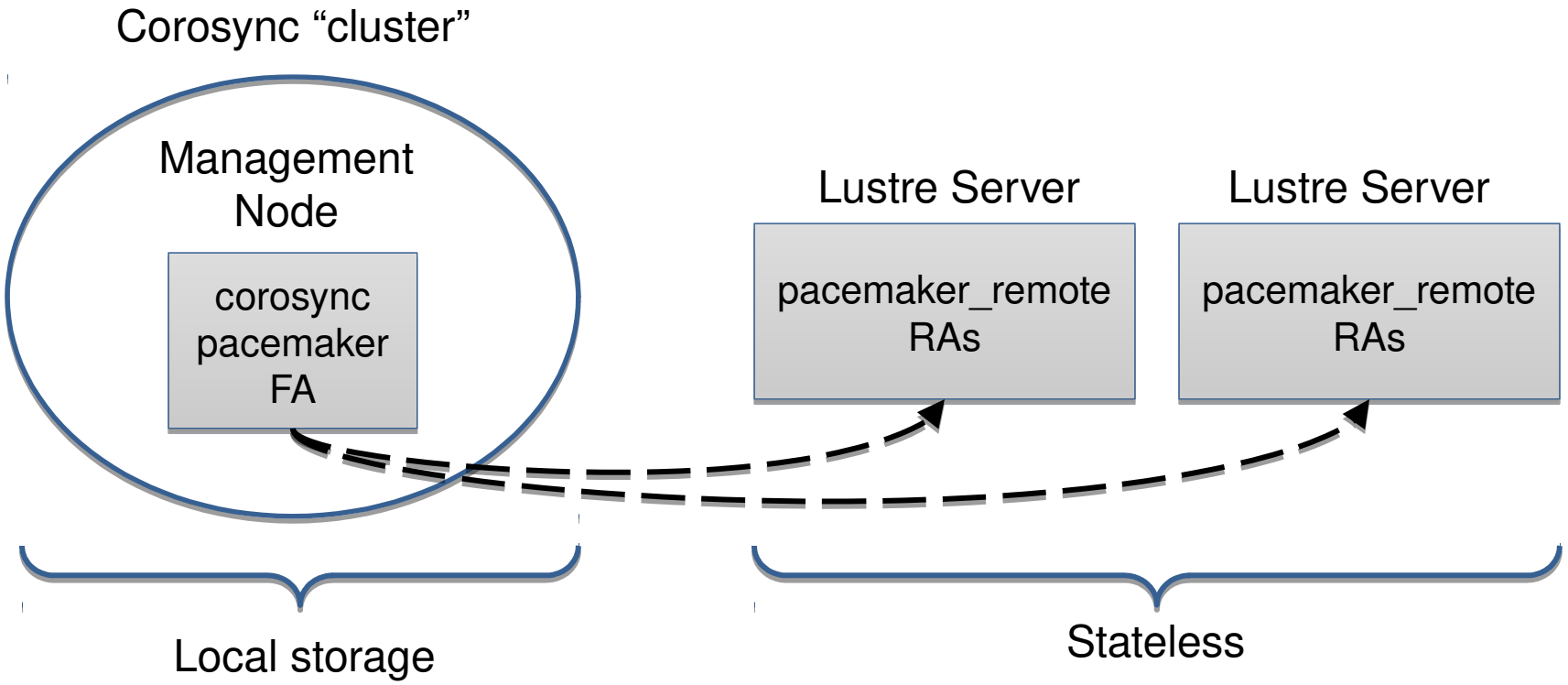
Better Approach: pacemaker_remote

- Allows corosync/pacemaker cluster to control non-corosync nodes
- Only configuration non-corosync nodes is /etc/packemaker/authkey
- Use RHEL standard *pcs* command
- Red Hat support
- Not documented in main Pacemaker manual (separate manual)

pacemaker_remote Architecture



LLNL's pacemaker_remote Architecture

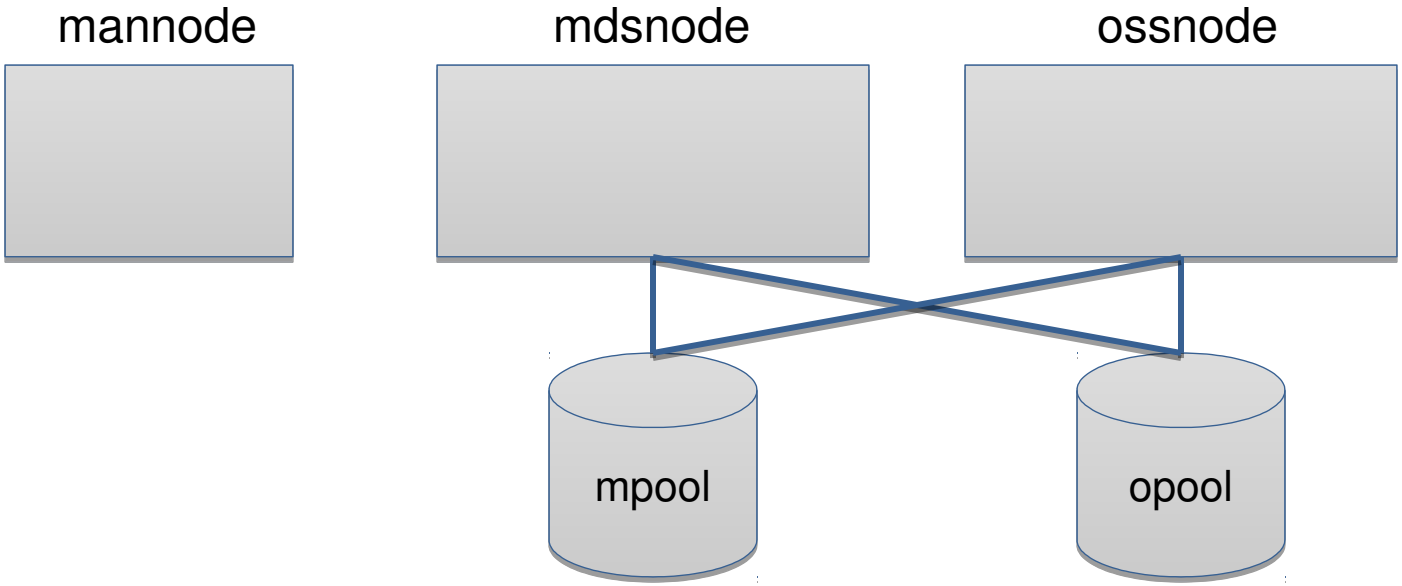


Configuration Details

LLNL Developed Agents

- *lustre* – RA for MGS, MDT, and OST resources
  - /usr/lib/ocf/resource.d/llnl/lustre
  - ocf:llnl:lustre
- *zpool* – RA for ZFS zpool resource
  - /usr/lib/ocf/resource.d/llnl/zpool
  - ocf:llnl:zpool
- *fence_powerman* – FA for *powerman* node power control
  - /usr/sbin/fence_powerman
  - fence_powerman

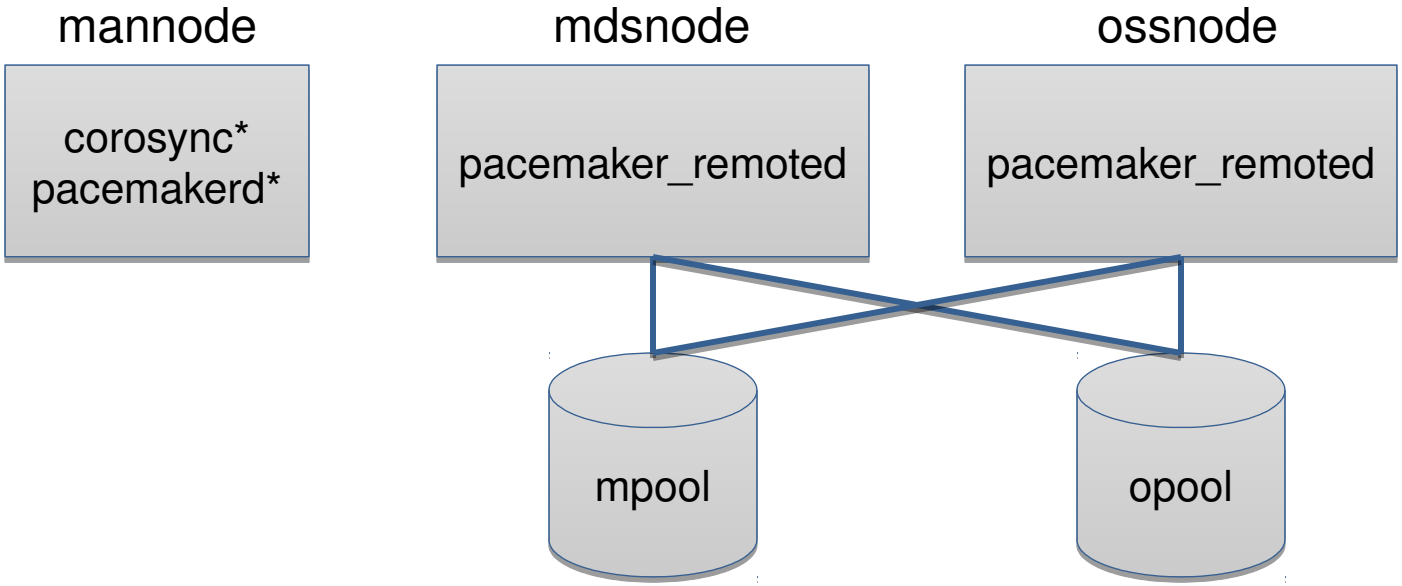
Example Cluster



Authentication

- Generate authkey
  - dd if=/dev/urandom of=authkey bs=4096 count=1
- Install at */etc/pacemaker/authkey* on all nodes
- Permission 0440, group *haclient*

Start Daemons



* The “pcs cluster start” command will start these for us

Initial Setup (On mannode)

- pcs cluster setup --name mycluster mannode
- pcs cluster start
  - starts both corosync and pacemaker

Symmetric Property

- pcs property set symmetric-cluster=false

Stonith Property

- pcs property set stonith-action=off

Batch Limit Property

- set property set batch-limit=100

Recheck Interval Property

- pcs property set cluster-recheck-interval=60

Fencing Setup

- pcs stonith create fence_pm fence_powerman ipaddr=localhost ipport=10101 pcmk_host_check="mdsnode,ossnode"
- pcs constraint location fence_pm prefers mannode

Configure Remote Nodes

- pcs resource create mds ocf:pacemaker:remote server=mdsnode reconnect_interval=60
- pcs resource create oss ocf:pacemaker:remote server=ossnode reconnect_interval=60
- pcs constraint location mds prefers mannode
- pcs constraint location oss prefers mannode

Configure ZFS zpool for MGS & MDT

- pcs resource create mpool ocf:llnl:zpool import_options="-f –N –d /dev/disk/by_vdev" pool=mpool
- pcs constraint location add mpool_l1 mpool mdsnode 20 ***resource-discovery=exclusive***
- pcs constraint location add mpool_l2 mpool ossnode 10 ***resource-discovery=exclusive***

Configure ZFS zpool for OST

- pcs resource create mpool ocf:llnl:zpool import_options="-f –N –d /dev/disk/by_vdev" pool=opool
- pcs constraint location add opool_l1 opool ossnode 20 ***resource-discovery=exclusive***
- pcs constraint location add opool_l2 opool mdsnode 10 ***resource-discovery=exclusive***

Configure Lustre MGS

- pcs resource create MGS ocf:llnl:lustre dataset=mpool/mgs mountpoint=/mnt/lustre/MGS
- pcs constraint order mpool then MGS
- pcs colocation add MGS with mpool score=INFINITY
- pcs constraint location add MGS_l1 MGS mdsnode 20 resource-discovery=exclusive
- pcs constraint location add MGS_l2 MGS ossnode 10 resource-discovery=exclusive

Configure Lustre MDT

- pcs resource create MDT ocf:llnl:lustre dataset=mpool/mds mountpoint=/mnt/lustre/MDT
- pcs constraint order mpool then MDT
- ***pcs constraint order MGS then MDT kind=Optional***
- pcs colocation add MDT with mpool score=INFINITY
- pcs constraint location add MDT_l1 MDT mdsnode 20 resource-discovery=exclusive
- pcs constraint location add MDT_l2 MDT ossnode 10 resource-discovery=exclusive

Configure Lustre OST

- pcs resource create OST ocf:llnl:lustre dataset=opool/ost mountpoint=/mnt/lustre/OST
- pcs constraint order opool then OST
- ***pcs constraint order MGS then OST kind=Optional***
- pcs colocation add OST with opool score=INFINITY
- pcs constraint location add OST_l1 OST ossnode 20 resource-discovery=exclusive
- pcs constraint location add OST_l2 OST mdsnode 10 resource-discovery=exclusive

pcs status

Cluster name: mycluster
Stack: corosync
Current DC: mannode (version 1.1.15-11.el7_3.2-e174ec8) – partition with quaorum
Last updated: Fri May 12 13:25:59 2017 Last change: …

3 nodes and 5 resources configured

Online: [mannode]
RemoteOnline: [mdsnode ossnode]

pcs status (continued)

Full list of resources:

```
fence_pm     (stonith:fence_powerman): Started
mds   (ocf::pacemaker:remote):       Started mannode
oss    (ocf::pacemaker:remote):      Started mannode
mpool     (ocf::llnl:zpool):         Started mds
opool     (ocf::llnl:zpool):         Started oss
MGS       (ocf::llnl:lustre):        Started mds
MDT       (ocf::llnl:lustre):        Started mds
OST        (ocf::llnl:lustre):       Started oss
```

Script Initial Setup

- Test Cluster of 20 Lustre servers (16 MDS & 4 OSS)
  - 45 seconds
  - 233 commands
- Production Cluster of 52 Lustre servers (16 MDS & 36 OSS)
  - 2-3 minutes
  - 585 commands
- ldev2pcs – Read ldev.conf and generate pcs commands

Positive Results

- No pacemaker state needed on stateless Lustre servers
- Single pacemaker instance for entire filesystem
  - (Start/stop entire filesytem with *systemctl start/stop pacemaker)*
- Manage all failover from cluster login/management node
- Scales to 50+ nodes in production

Future Work

- *pcs status* compact form
- pacemaker administration learning curve
- Better timeout values
- Test scaling limits
- Improve Resource Agents' monitor actions
- Add LNet Resource Agent

Get The Source

http://github.com/LLNL/lustre-tools-llnl

/tree/master/scripts/[lustre,zpool,ldev2pcs]


Lawrence Livermore
National Laboratory