Two Node Cluster with SUSE®

SUSE® Linux Enterprise Server High Availability Extension
DRBD and OCFS2
KVM / XEN

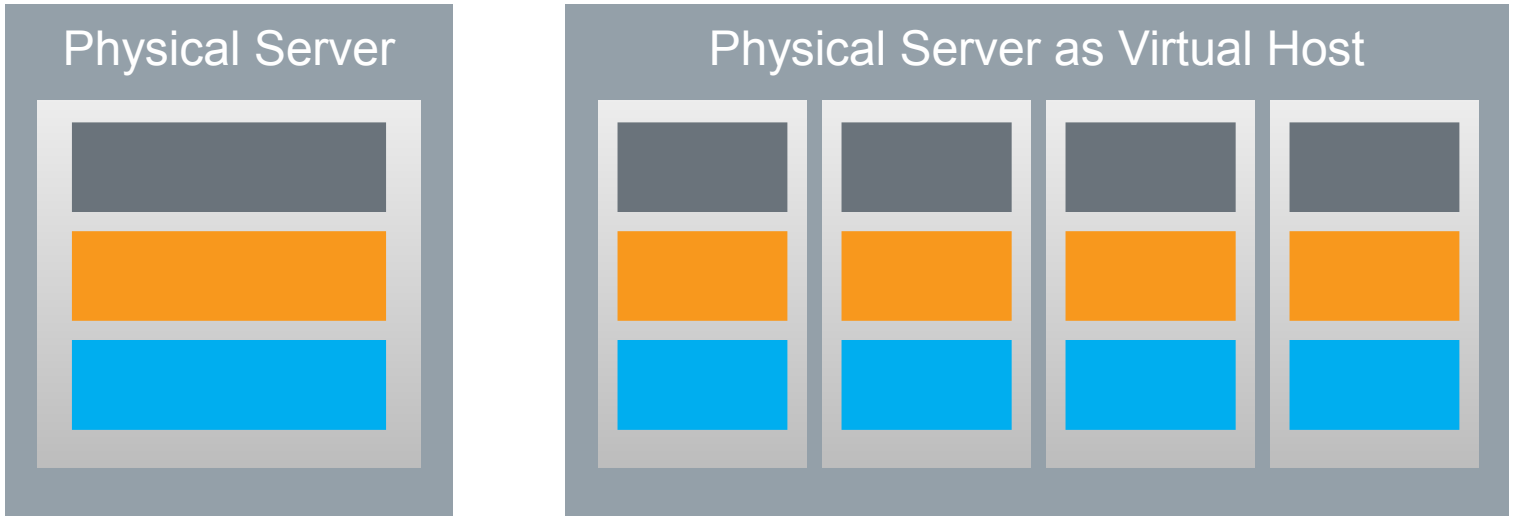
Mark Gardner
markgard@gmail.com

SUSE®
We adapt. You succeed.

In this Session

- This is a SUSE® Linux Experts Forum covering best practices.
- Learn how to create a simple two-node cluster using SUSE Linux Enterprise Server and the High Availability Extension. Free your workloads from the bonds of physical hardware. Host multiple workloads on this cluster. Learn various high availability techniques with VLS and/or OpenAIS/Pacemaker.
- In this session you will learn how to use technologies such as DRBD, Pacemaker, OCFS2, OpenAIS and Xen. Learn logical organization of clustered services, and become exposed to several configuration examples.

What is a Workload?



1 workload per physical server

Multiple workloads per physical server

Example Workloads

- Apache
- Application Servers (Tomcat/JBoss/Glassfish)
- DNS
- Database
- LDAP
- An entire virtualized guest operating system can be treated as a single workload.

Define Your Goals

- Keep them simple

- Two node KVM/Xen virtualization cluster
- Active/Active Distributed Filesystem
- Virtualized Guests with Live Migration, automatic failover on node failure
- SBD STONITH
- Pacemaker for supporting resources

What Are We Trying to Accomplish?

- Consolidation
- High Availability
- Disaster Recovery
- Better Hardware Utilization
- Better Service Levels

Cluster Intro

"A cluster is a type of parallel or distributed systems that: Consists of a collection of interconnected whole computers. It is used as a single unified computing resource"

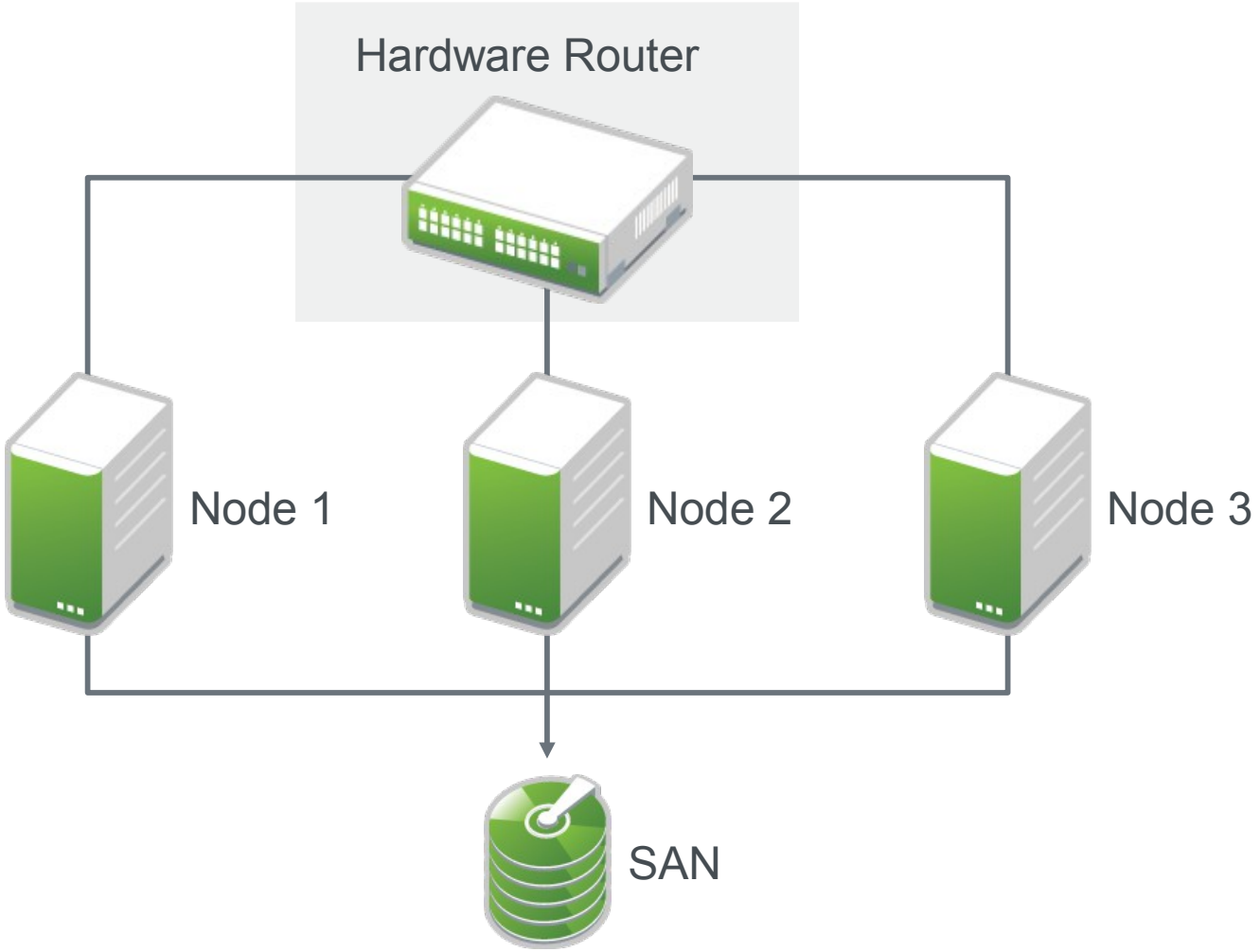
Dr. Gregory Pfister, In search of Clusters (1995)

Types of Clusters

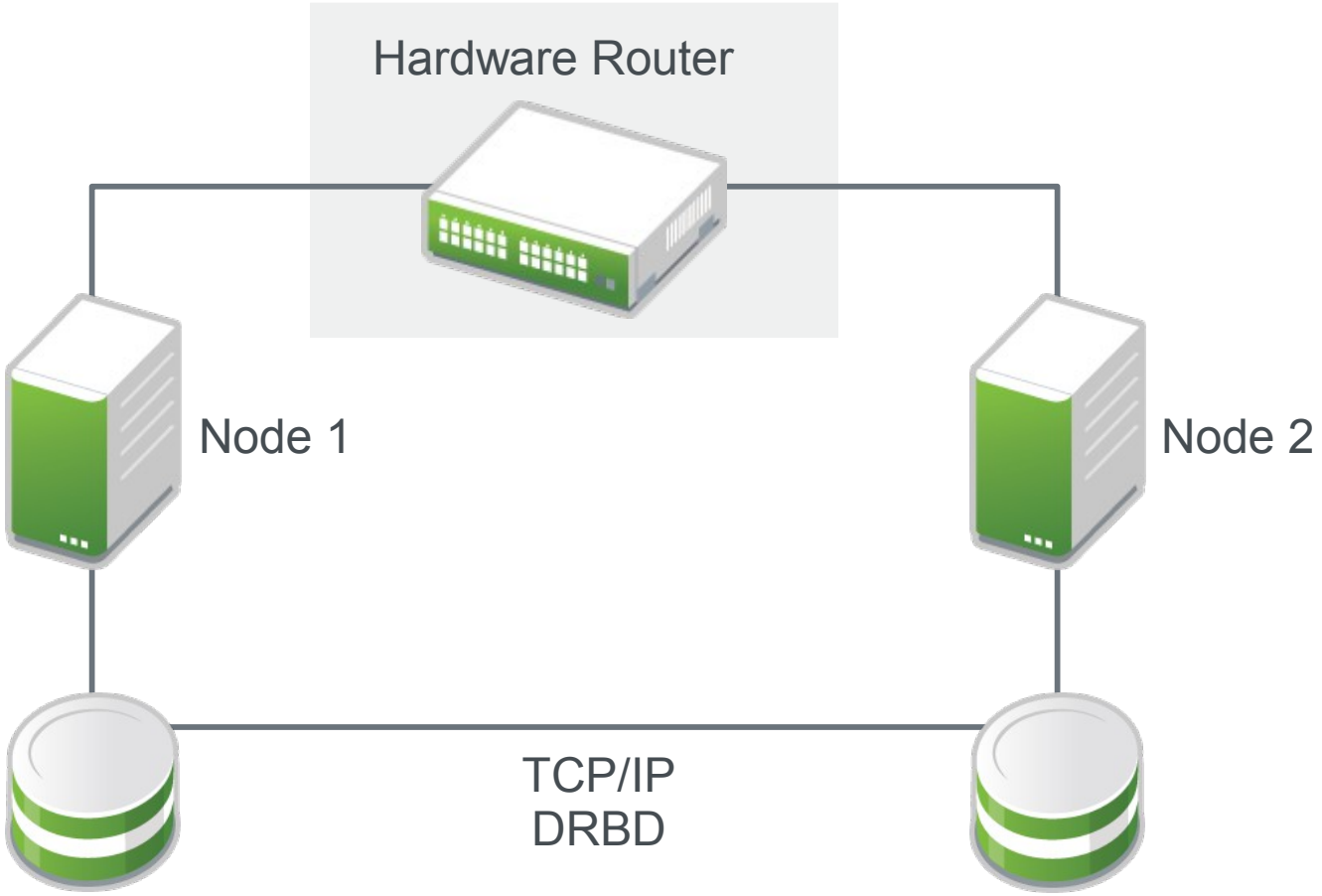
- Five well known cluster types
  - HA: High Availability Cluster
  - HTC: High Throughput Cluster
  - HPC: High Performance Cluster
  - VSC: Virtual System Cluster
  - Grid Computing

Common Cluster Configuration

Traditional Cluster



2 Node Setup with Replicated Storage

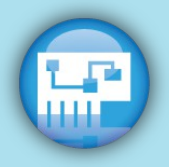


Cluster Building Blocks

Available Resources and Workloads

High Availability Engine

Operating System Setup



Network and Fencing

Local and Shared Storage

Cluster Building Blocks

Available Resources and Workloads

High Availability Engine

Operating System Setup

Network and Fencing

Local and Shared Storage

Shared Storage

- Shared storage delivery depends heavily on your workload profile.
  - Virtual Machines
  - Application Resources
- Shared Storage Types
  - Storage Area Network (SAN)
  - NFS
  - iSCSI Targets
  - DRBD Replicated Storage
- Partitioning Strategies

Local Storage and File Systems

- Local storage recommendations
- Local storage partitioning strategy

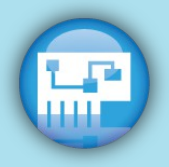
- File Systems (all need CLVM2)
  - OCFS2
  - GFS
  - LustreFS
  - CODA FS

Cluster Building Blocks

Available Resources and Workloads

High Availability Engine

Operating System Setup



Network and Fencing

Local and Shared Storage

Network Configuration

- Two TCP/IP channels
- Bonded channels
- Network must support multicast
  - Recent patch allows corosync to supports unicast
- Second most important component of clustering after reliable storage.

Network Bonding Modes

- Mode=0 (balance-rr): Round Robin
- Mode=1 (active-backup): Only one active
- Mode=2 (balance-xor): Transmit based on MAC
- Mode=3 (broadcast): Transmit on all slaves
- Mode=4 (802.3ad): Dynamic Link Aggregation !!!
  - Requires special switch configuration
- Mode=5 (balance-tlb): Transmit Load Balance
- Mode=6 (balance-alb): Adaptive Load Balance

The most common modes are: 1, 2, 0, & 4 (4 is best)

Fencing

- Fencing limits and many cases prevents situations that result in Split Brain
- Node Fencing
  - STONITH
  - XEN Guest destruction
  - SBD
- Resource Fencing
  - LUN Reservation
  - Quorum
  - Turning off switch ports

Cluster Building Blocks

Available Resources and Workloads

High Availability Engine

Operating System Setup

Network and Fencing

Local and Shared Storage

Cluster Building Blocks

Available Resources and Workloads

High Availability Engine

Operating System Setup

Network and Fencing

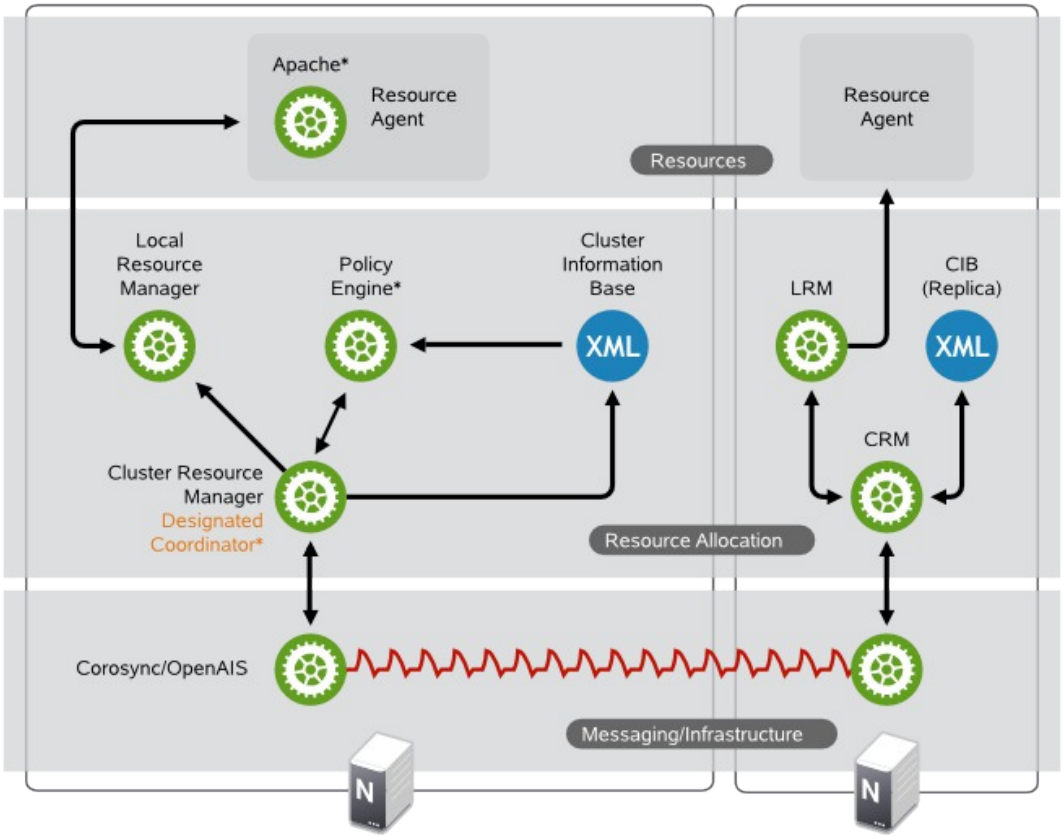
Local and Shared Storage

SUSE® Linux Enterprise High Availability Extension

HA Stack from 10 to 11

SLES 10	SLE HA 11	SLE HA 11 SP1
Heartbeat	OCFS2 general FS	Metro-Area Cluster
DRBD 0.7	Unified CLI	Storage Quorum Coverage
Yast2-HB	Pacemaker	Samba Cluster
OCFS2 / EVMS2	openAIS	Enhanced Data Replication
	HA GUI	Cluster Config Synchronization
	Yast2-DRBD	Node Recovery
	Yast2-Multipath	Web GUI
Part of SLES 10	Added in SLE HA 11	Added in SLE HA 11 SP1

Cluster Engine Architecture



Cluster Building Blocks

Available Resources and Workloads

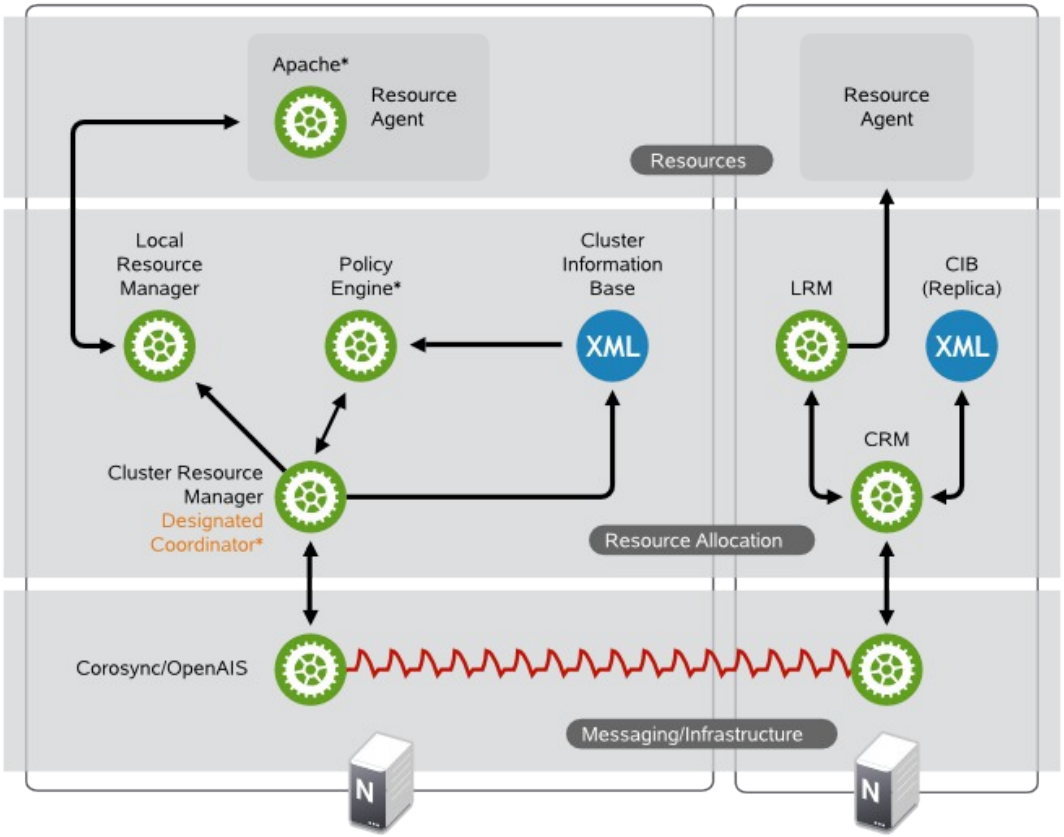
High Availability Engine

Operating System Setup

Network and Fencing

Local and Shared Storage

Who Controls What?



Resource Agents

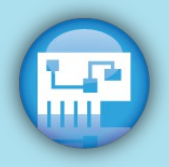
- Resource agents are the bits of code that bridge the gap between what happens in the cluster, and what a managed resource can do or is doing.
- Resource Agents can be written in any language.
- Search for Open Cluster Framework (OCF) agents before using an agent of another class.
- Avoid legacy Heartbeatv1 agents, they are only around for migration from old versions of heartbeat.

Cluster Building Blocks

Available Resources and Workloads

High Availability Engine

Operating System Setup



Network and Fencing

Local and Shared Storage

Sample Configurations

The End Goal

High Availability clustering solution

Service availability 24h a day

- Oracle Cluster File System 2 (OCFS2)

  Sharing and Scaling data-access by multiple nodes

- Distributed Remote Block Device 8 (DRBD)

  Provide low-cost “SAN” through disk replication via TCP

- User-friendly tools

Setup

- 2 Servers installed with SUSE® Linux Enterprise Server 11 SP1. Unallocated space on local storage for Xen Guests
- Bonded Network Interfaces (best if setup on different switches). This is our first line of defense against Split Brain scenarios
- High Availability Extension installed and patched to latest version.

Messaging Layer: Corosync

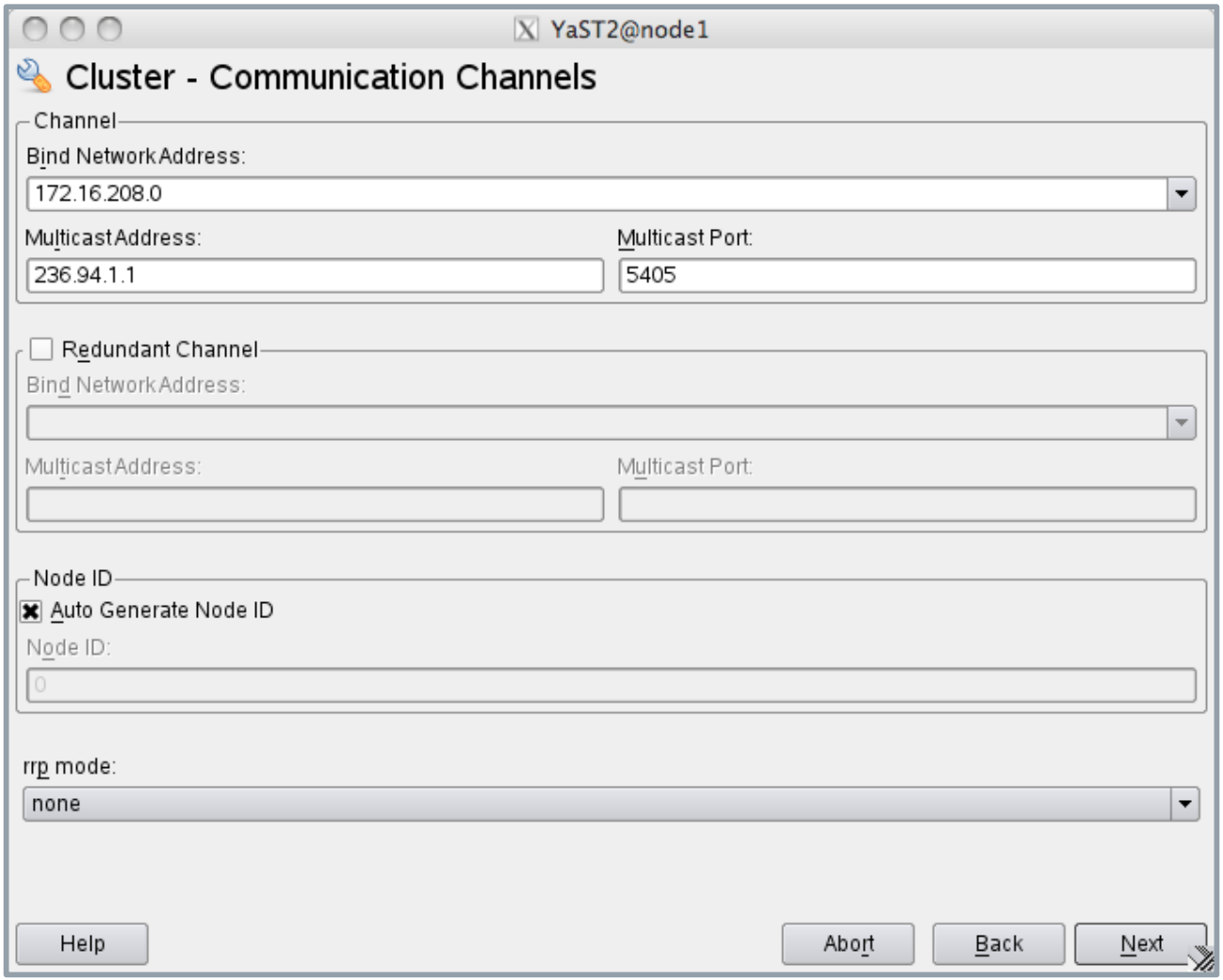
Setup Corosync

- This can be configured via the command line by configuring /etc/corosync/corosync.conf
- Requires network multicast for communication with nodes.
- IPV4 required if you want redundant rings (IPV6 does not yet support redundant rings)
- A recent patch allows for unicast messaging communication instead of multicast.

Corosync Setup (I)

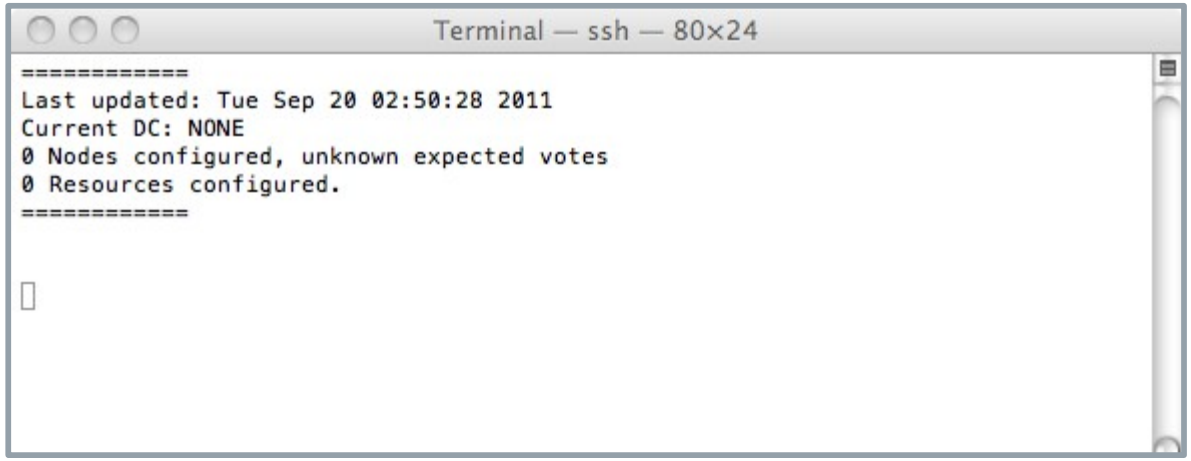
- Start YaST2
  - Set Communication Channel
  - Select Port 5405
  - Select Multicast Address 239.239.0.1
  - Make sure this address is unique, 229.239.0.1 is filtered out by some high end switches.
  - Select Auto Generate Node ID
  - Enable Security Auth
  - Generate authkey file on first node only
  - Copy authkey file and corosync.conf to all other cluster nodes

Corosync Setup (II)



Start Corosync

- Run command rcopenais start
- Run command crm_mon
  - You will get a screen like this for up to 2-3 minutes while the cluster configures itself for the first time.



Messaging Layer Complete

- Check Corosync

Terminal — ssh — 54×11

```
node1:~ # corosync-cfgtool -s
Printing ring status.
Local node ID 214962348
RING ID 0
        id      = 172.16.208.140
        status  = ring 0 active with no faults
node1:~ #
node1:~ #
```

Terminal — ssh — 76×19

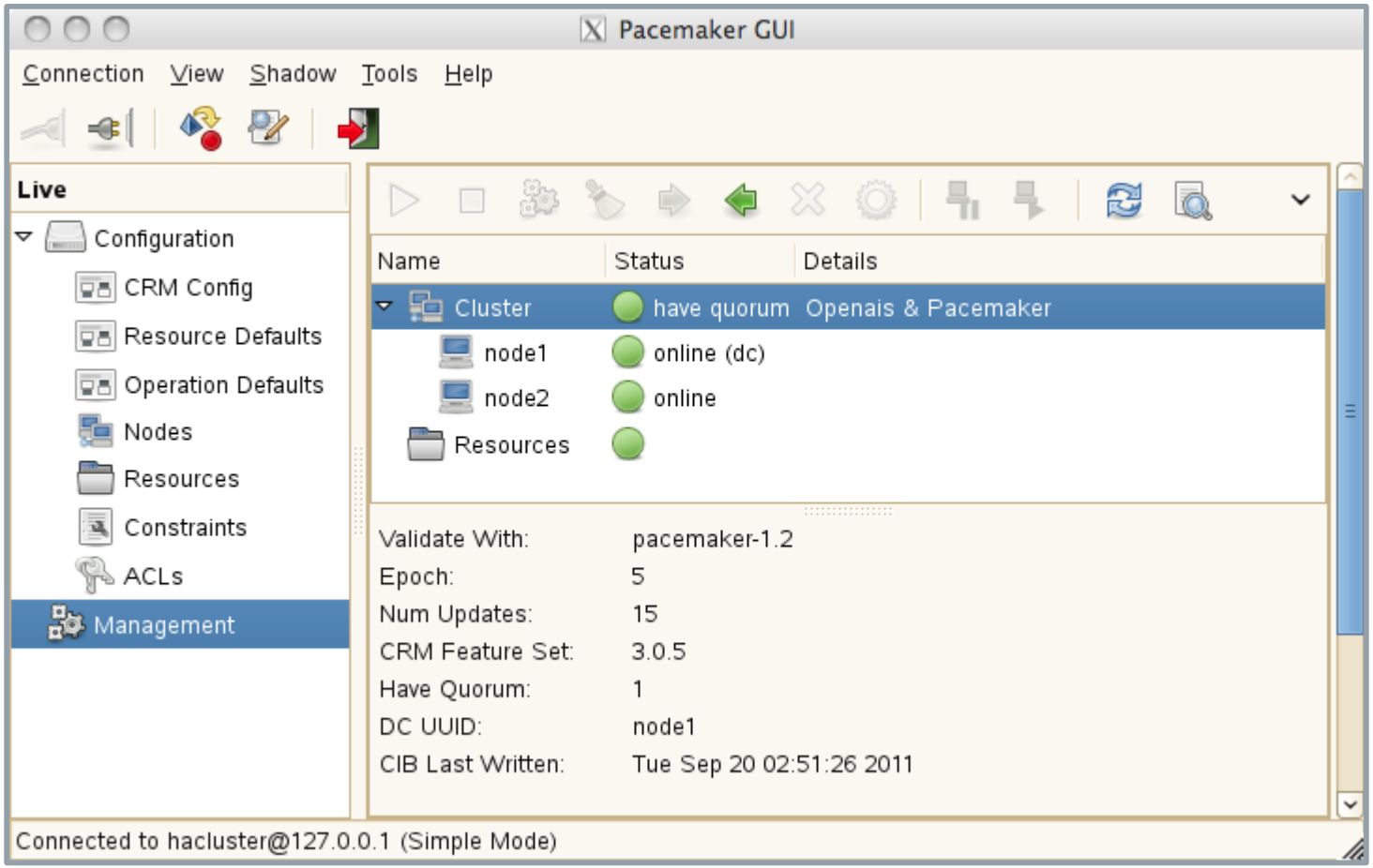
```
============
Last updated: Tue Sep 20 03:02:11 2011
Stack: openais
Current DC: node1 - partition with quorum
Version: 1.1.5-5bd2b9154d7d9f86d7f56fe0a74072a5a6590c60
2 Nodes configured, 2 expected votes
0 Resources configured.
============

Online: [ node1 node2 ]
```

Setup the Pacemaker GUI

- When you install pacemaker the hacluster user is created without a password
- 1) Run command: passwd hacluster
- 2) Run command: crm_gui
- 3) Enter in the password you set in step 1

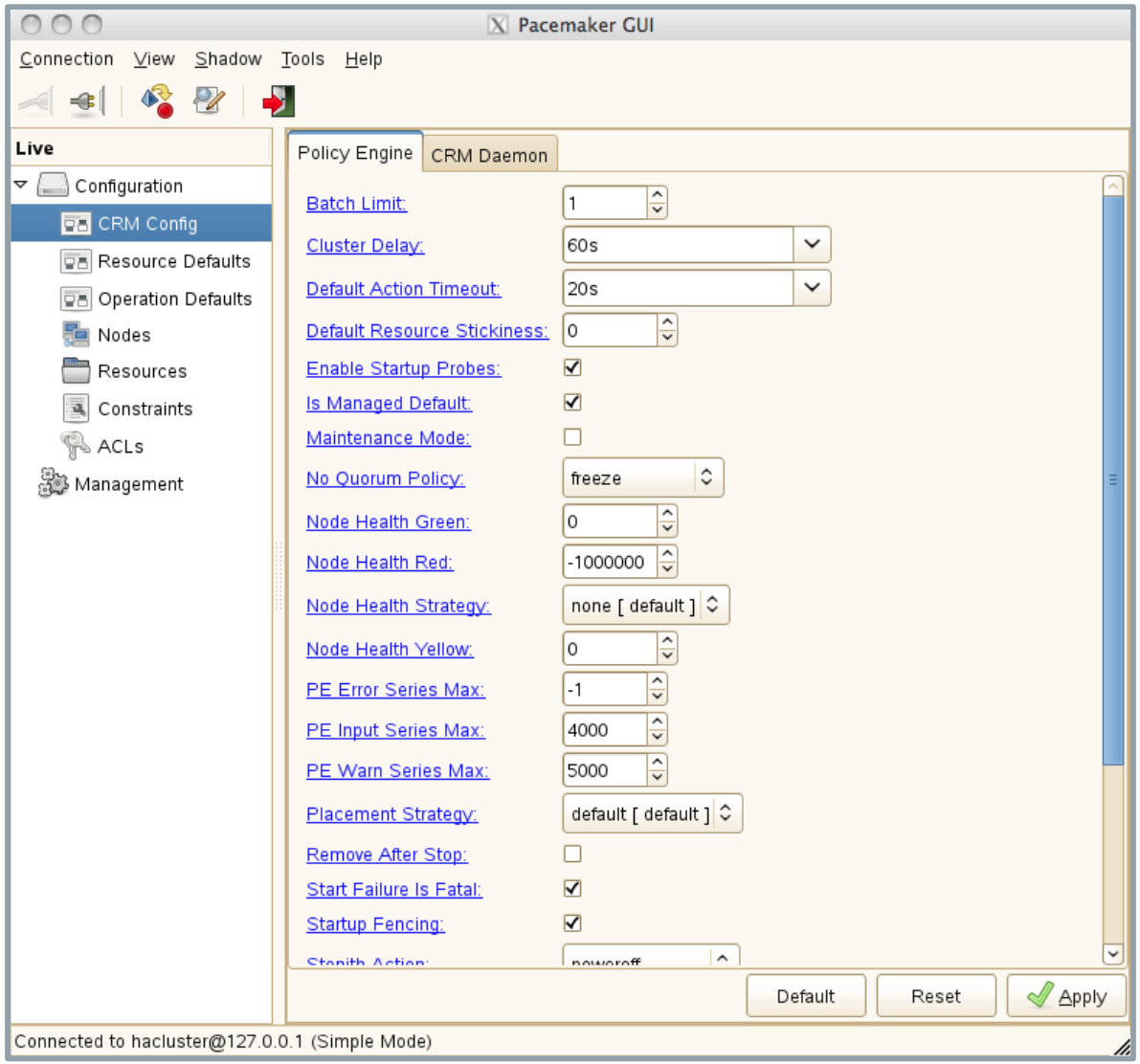
Pacemaker GUI



Configure the CRM

- Change Batch Limit from 30 to 1
- Change No Quorum Policy to freeze
- Change Stonith Action to poweroff
- Uncheck Stonith Enabled

Basic CRM Configuration



DRBD Setup

Initialize the Disks

- Check file syntax
  - Drbdadm dump all
- Copy the DRBD configuration files to the other node:
  - scp /etc/drbd.conf node2:/etc/
  - scp /etc/drbd.d/* node2:/etc/drbd.d/
- Initialize the meta data on both systems by entering the following on each node.
  - drbdadm -- --ignore-sanity-checks create-md r0
  - rcdrbd start
- Check with “rcdrbd status”

Initial Synchronization

- Start the resync process on your intended primary node (node1 in this case):
  - drbdadm -- --overwrite-data-of-peer primary r0
- Check the status again with rcdrbd status and you get:
  - ... m:res cs ro ds p mounted fstype 0:r0 Connected Primary/Secondary UpToDate/UpToDate C
- The status in the ds row must be UpToDate/UpToDate
- Set node1 as primary node:
  - drbdadm primary r0

Add DRBD to Pacemaker

```
primitive drbd-r0 ocf:linbit:drbd \
    params drbd_resource="r0" \
    op monitor interval="30" role="Slave" timeout="20" \
    op monitor interval="20" role="Master" timeout="20”
ms ms-drbd-r0 drbd-r0 \
    meta interleave="true" master-max="2" master-node-
max="1" notify="true" is-managed="true”
```

OCFS2 Setup

OCFS2 Setup

Run the CRM commands to configure the Distributed Lock Manager (DLM)

```
node2:~ # crm configure
primitive dlm ocf:pacemaker:controld \
    op monitor interval="60" timeout="60”
primitive o2cb ocf:ocfs2:o2cb \
    op monitor interval="60" timeout="60”
Group grp-o2cb dlm o2cb
Clone clone-o2cb grp-o2cb meta \
    interleave="true”
Commit
```

Create the File System

- Once the supportive pieces are in place we can create the file system with this command.
- `mkfs.ocfs2 -T vmstore /dev/drbd_r0`

- Test the new file system by mounting it, writing a file, reading the file from the second node.

```
node1:~ # mount -t ocfs2 /dev/drbd_r0 /data/

node1:~ # touch /data/HelloThereFromNode1

node2:~ # ls /data/

HelloThereFromNode1  lost+found
```

Add File System Resource

```
Primitive prim-ocfs2-data ocf:heartbest:Filesystem \
  prams device="/dev/drbd_r0 " directory="/data/" \
  fstype="ocfs2" \
  op monitor interval="20" timout="40" start-delay="10" \
  op start interval="0" timeout="60" \
  op stop interval="0" timeout="60"
clone clone-ocfs prim-ocfs2-data \
  meta interleave="true"
Delete base-clone base-group
group grp-ocfs2 dlm o2cb prim-ocfs2-data
Clone ocfs2-clone grp-ocfs2 meta interleave="true"
order drbd-before-ocfs2 inf: ocfs2-clone ms-drbd-r0 \
    clone-o2cb
```

KVM / XEN

Xen Guest Resource Agent

```
primitive xen-vm1-vm ocf:heartbeat:Xen \
  meta allow-migrate="true" priority="4" \
  target-role="Started" \
  params xmfile="/etc/xen/vm/xen-vm1" \
  op monitor interval="60" timeout="240" \
  op start interval="0" timeout="120" \
  op stop interval="0" timeout="1800" \
  op migrate_to interval="0" timeout="1800" \
  op migrate_from interval="0" timeout="1800"
```

KVM Guest Resource Agent

```
primitive kvm-guest1 VirtualDomain \
  hypervisor="qemu:///system"\
  config="/root/guest1.xml"\
  meta allow-migrate="true" priority="4"
\
  target-role="Started" \
  op monitor interval="60" timeout="240"
\
  op start interval="0" timeout="120" \
  op stop interval="0" timeout="1800" \
  op migrate_to interval="0"
timeout="1800" \
```

Additional Configuration

- Later versions of SBD allow dual shared storage. One project may be to make each xen host an iscsi target for the sbd service.
- Then your xen guests can take adavantage of sbd

Otherwise there is a stonith agent that will simply destroy a xen guest. As well.

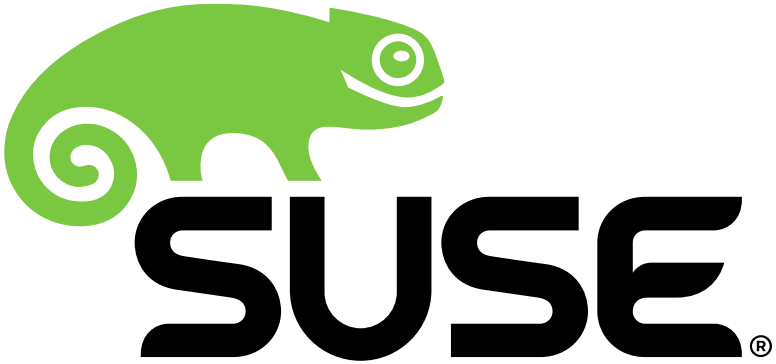
New In SUSE Linux Enterprise 12

What's New in SUSE Linux Enterprise Server 12

- Complete stack refresh
  - Latest version of products that make up HAE
- DRBD is still 8.4
- SBD is MUCH more stable
- Hawk Is preferred cluster config tool.

Questions?

Thank you.


SUSE®
We adapt. You succeed.