

# High Availability on Linux - the SUSE way

**Roger Zhou**  
**SUSE Linux Enterprise Extension**  
**Senior Engineering Manager**  
**zzhou@suse.com**



# CURIOSITY

## in the land of Linux High Availability



# Agenda

1. History
2. Overview of HA architectural components
3. Cluster use case examples
4. Future outlook

Note: HA stack is quite broad, and don't surprise quit some terms/concepts involved.

# What is Cluster?

- HPC (super computing)
  - High performance computing.
- Load Balancer
  - Very high capacity.
- High Availability
  - 99.999% +
  - MTTR(mean time to repair)
  - SPOF(single point of failure)

# Challenge in High Availability

## Murphy's Law is Universal

- "Everything that can go wrong will go wrong"
  - Service outage and loss of data integrity
- Hardware crash, flood, fire, power outage, earthquake?
  - You might afford a five second blip, but can you afford a longer outage?
  - How much does downtime cost?
  - Can you afford low availability systems?



# "HA" term is widely used

- Often confusing, example, “SUSE HA Clustering” vs. “VMWare HA”
- VMWare vSphere HA ( a close-source company )
  - Focusing on hypervisor level and host hardware level.
  - It makes sure VM restarted somewhere else.
  - It is agnostic on Guest OS inside VM.
- SUSE HA ( all open-source )
  - It is a OS level solution to protect critical resources running on VM.
  - It provides \*high availability within Linux OS\*.
  - That said, for Windows Guest, you need Windows HA solution.
- More than that, more specific
  - HADOOP HA
  - OpenStack HA



# History of HA in Linux OS

- 1990s, simple solution from Heartbeat project. Only two nodes.
- Early 2000s, Heartbeat 2.0.
  - Heartbeat became too complex. Monitor the resources themselves
  - Industry demands to split to two projects and evolves.
    - 1) one for cluster membership: RH cman, Corosync(was OpenAIS)
    - 2) one for resource management: Pacemaker
- Today, ClusterLabs, a completely different solution in early days, merged more components of Heartbeat project.



# HA Hardware Components

- Multiple networks
  1. A user network for external user access. Can be high volume.
  2. A dedicated network for cluster communication: messaging & membership, STONITH.
  3. A dedicated storage network infrastructure.
- Switch for network bonding (aka. Link aggregation)
- Fencing/STONITH devices (remote “powerswitch”)
- Shared storage: NAS(nfs/cifs), SAN(fc/iscsi)



# Typical HA Problem - Split Brain

- Split partitions run the same service, what?
  - It just breaks data integrity !!!
- Two key concepts as the solution:

- Quorum

It means "majority". if the cluster doesn't have quorum, no actions will be taken in the cluster. That said, fencing and resource management are disabled without quorum.

- STONITH

It stands for "shoot the other node in the head", aka, fencing.

Cluster doesn't accept any confusing state.



# Architectural Software Components

# Architectural Software Components

- Corosync:
  - messaging and membership service.
- Pacemaker:
  - cluster resource manager
- Resource Agents (RAs):
  - manage and monitor availability of services
- Fencing Devices:
  - STONITH to ensure data integrity
- User Interface:
  - crmsh CLI tools and Hawk web UI
- Booth:
  - RAFT consensus algorithm for GEO Cluster.



# More

Quite some?! There are more, outside of "clusterlabs.org"

- LVS:
  - Kernel space, Layer 4, ip + port.
- HAproxy:
  - user space, Layer 7, HTTP based.
- Shared filesystem:
  - OCFS2 / GFS2
- Block device replication:
  - DRBD, cLVM mirroring, Cluster md raid1
- Shared storage:
  - SAN (FC / FCoE / iSCSI)
- Multipathing:
  - dm-mpio + multipath-tools



# Components in details

# Corosync: messaging and membership

- A consensus algorithm
  - "Totem Single Ring Ordering and Membership protocol"
- A closed process group communication model
  - You can do analogy with TCP/IP 3-way hand shaking.
    - Message ordering.
    - Membership handling.
- A in memory object database for Configuration engines and Service engines. Shared-nothing cluster.
- A quorum system that notifies applications when quorum is achieved or lost.



# Pacemaker: the resources manager

- The brain of the cluster.
- Policy engine for decision making.
  - To start/stop resources on a node according to the score.
  - To monitor resources according to interval.
  - To restart resources if monitor fails.
  - To fence/STONITH a node if stop operation fails.

# Resources Agents, STONITH

- Resources Agents (RAs)

- LSB shell scripts: start / stop / monitor
- Write RA for your applications
- More than hundred contributors in upstream github.

- STONITH (aka. Fencing)

- It stands for "shoot the other node in the head"
- Data integrity does not tolerate any confusing state.
- Before migrating resources to another node in the cluster, the cluster must confirm the suspicious node really is down.

NOTE: "Have a lot of Fun" at <http://ourobengr.com/ha/>

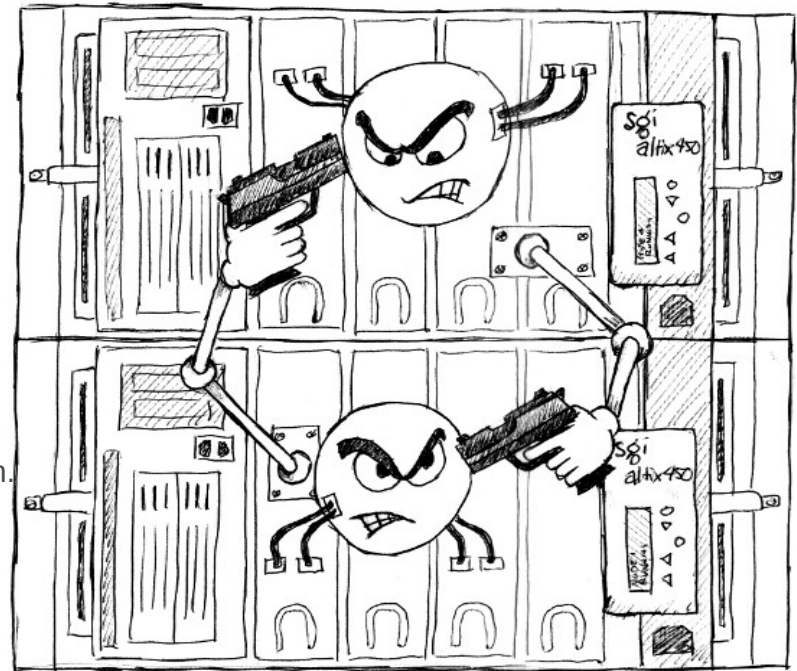
- Popular devices:

- APC PDU (networked powerswitch)
- Intel vPro AMT, HP iLO, Dell DRAC, IBM IMM, IPMI Alliance
- Software library to deal with KVM, Xen and VMware Vms.
- Software based : SBD (STONITH Block Device) to do self termination.

- Fencing devices can be chained.

NOTE: sbd is a software approach as last implicit option in the fencing topology.

- STONITH is mandatory and not optional for \*enterprise\* Linux HA clusters.

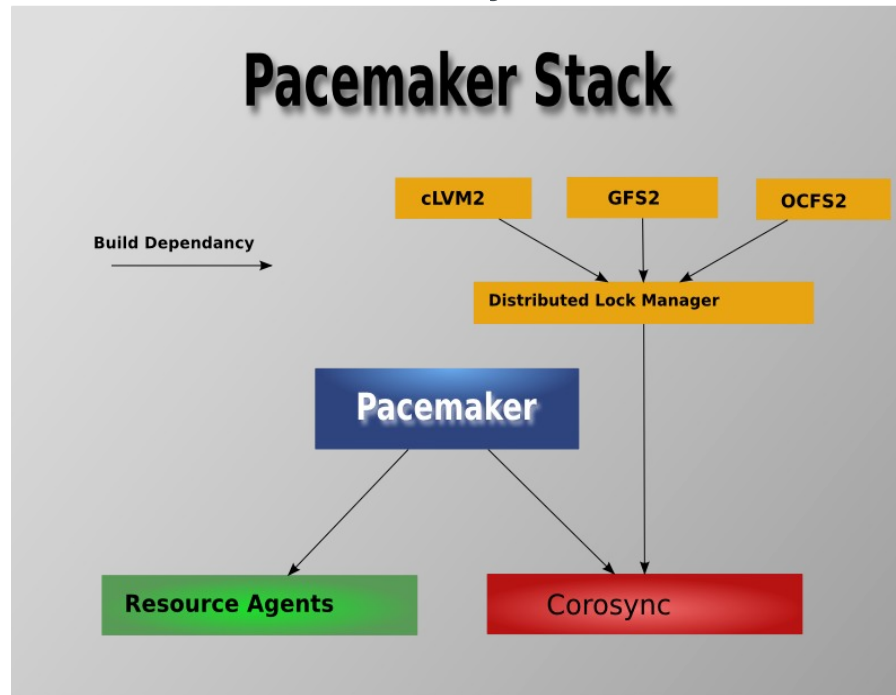


**DON'T ANYBODY MOVE ...**



# Cluster Filesystem on shared disk

- OCFS2 / GFS2
  - Filesystem on top of the shared storage.  
Note: there are many others non-open source implementation.
  - Multiple nodes concurrently access the same filesystem.



[http://clusterlabs.org/doc/fr/Pacemaker/1.1/html/Clusters\\_from\\_Scratch/\\_pacemaker\\_architecture.html](http://clusterlabs.org/doc/fr/Pacemaker/1.1/html/Clusters_from_Scratch/_pacemaker_architecture.html)



# Cluster Block Device

- DRBD
  - network based raid1
  - high performance data replication over network
- CLVM2
  - It enables the cluster to leverage the brilliant flexibility of lvm.
  - Multiple nodes can create and reallocate volumes on the shared disk.
  - clvmd distributes LVM metadata updates in the cluster.
  - Data replication speed is way too slow.
- Cluster md raid1
  - multiple nodes to use the shared disk as md-raid1 device.
  - High performance raid1 solution in cluster environment.



# Cluster Examples

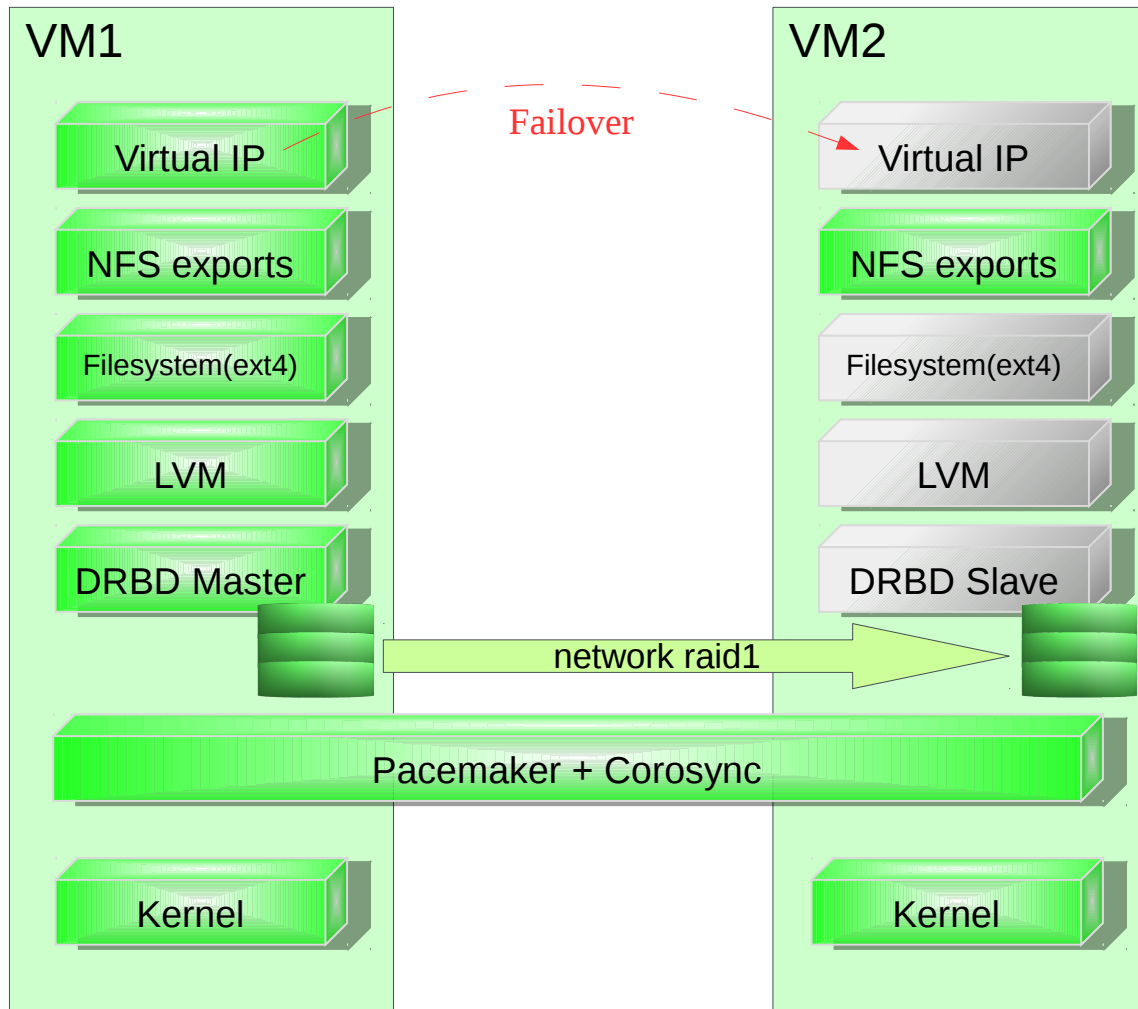
quite some terms, confusing?

No?!

Now, let's illustrate some examples.

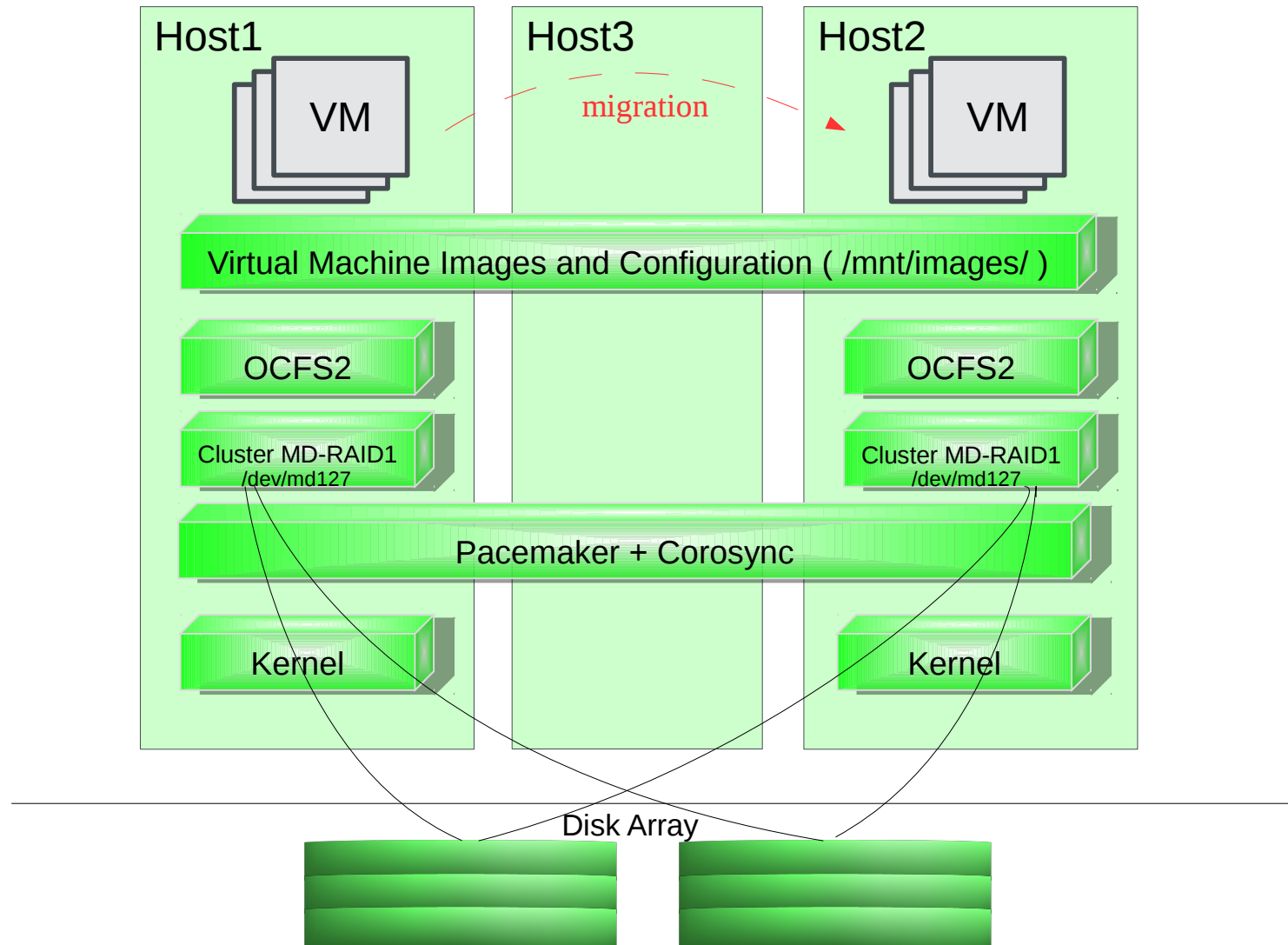
# NFS Server ( High Available NAS )

## Cluster Example in Diagram



# Cluster FS - OCFS2 on shared disk

Cluster Example in Diagram



# Future outlook

- Upstream activities

- docker: in early adoption
- openstack: adopted in control plane. WIP in compute domain.
- scalability of corosync/pacemaker
- ...

- Where you can contribute

- openSUSE Leap ( Tumbleweed )  
RHEL/CentOS ( Fedora )  
many other related upstreams
- Use it, test it, ask questions in related mailinglist.
- Submit patches for the upstream.



# Reference

All **\*open-source\*** in the whole stack. Go, googling, ...

- SUSE HA Doc: <https://www.suse.com/documentation/sle-ha-12/>
- HA clusterlabs portal: <http://clusterlabs.org/>
  
- iSCSI LIO: <http://linux-iscsi.org>
- OCFS2 wiki: <https://ocfs2.wiki.kernel.org/>
- LVM2 DM: <https://www.sourceware.org/lvm2/>
- DRBD portal: <http://drbd.linbit.com/>
  
- md raid1 doc: <https://www.kernel.org/doc/Documentation/md-cluster.txt>
- Corosync doc: <http://landley.net/kdocs/ols/2008/ols2008v1-pages-85-100.pdf>
  
- General HA Users: [users@clusterlabs.org](mailto:users@clusterlabs.org)



Q & A ?





**Corporate Headquarters**  
Maxfeldstrasse 5  
90409 Nuremberg  
Germany

+49 911 740 53 0 (Worldwide)  
[+www.suse.com](http://www.suse.com)

Join us on:  
[www.opensuse.org](http://www.opensuse.org)

## **Unpublished Work of SUSE. All Rights Reserved.**

This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

## **General Disclaimer**

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.

