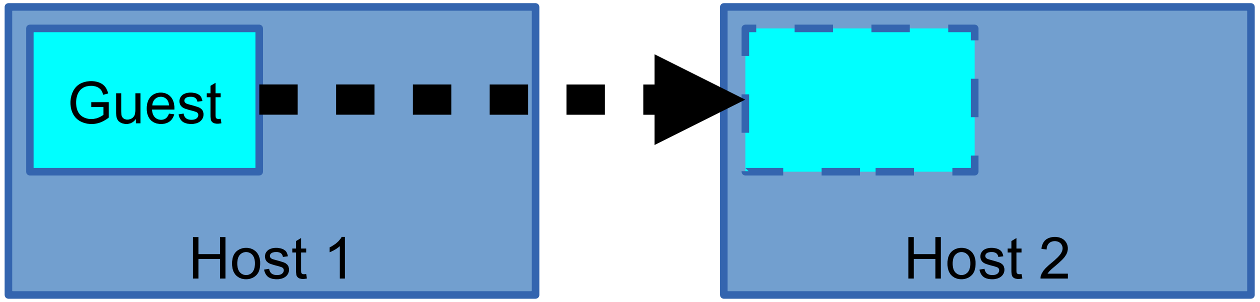
VM Migration and Clustering

UNIVERSITY OF OREGON

NSRC
Network Startup Resource Center

Migration

- Moving a VM from one host to another



Applications include…

- Load balancing
  - Move VMs to a less busy host
  - Make use of newly-added capacity
- Maintenance
  - Move VMs off a host before it is shut down
- Recovery from host failure
  - Restart VM on a different host
- These are major benefits over traditional IT

Types of migration

- "Cold" or "Offline"
  - Shutdown VM on host 1, restart on host 2
  - Safe and simple, but causes a short outage
- "Live" or "Online"
  - Copy across RAM *while VM continues to run*
  - Mark "dirty" (changed) pages & re-copy
  - Brief suspension for final copy (<< 1 sec)
  - Even network sessions remain up. Users don't notice anything at all!

Migration and Storage

- The VM disk image has to be accessible from the new host after the migration
- Simple: just copy the disk image across
  - But it takes a long time, if the volume is large
- What about live migration of storage?
  - Yes, Proxmox can do this
  - It sets up temporary network access to storage volume (via "nbd") during the copy
- If you have network-accessed storage, this issue goes away

Coordination between hosts

- Migration (especially live migration) requires cooperation between the sending and receiving hosts
- Many systems achieve this by joining nodes into a "cluster" with a common control plane
- Gives overall view and management of the set of nodes

Proxmox clustering

- Enable clustering on first node, then join second, third etc
  - The newly-joined nodes must be initially empty (no VMs)
- All nodes get an identical copy of the cluster database
  - Exposed as mounted filesystem on `/etc/pve` (fuse "pmxcfs")
  - Any change immediately replicates to all other nodes
  - Proxmox VE configures all this for you
- You can manage everything through *any* node's GUI

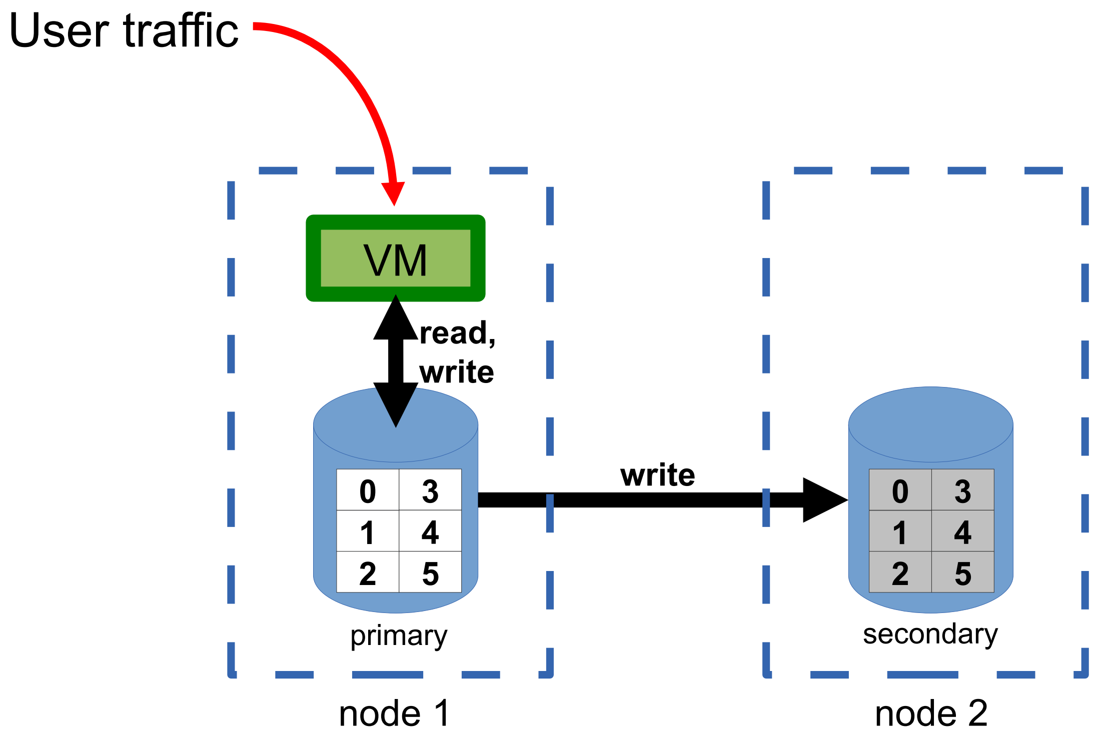
Design considerations

- Nodes coordinate via "corosync" which sends regular heartbeats
- Reliable communication is critical
  - Recommended best practice is a separate NIC (1G) dedicated to corosync traffic - or even two
  - Cluster nodes should be close to each other (low latency)
  - Don't span a cluster over a wide geographical area
- Limit cluster size to 15-20 nodes
  - Beyond that, create multiple independent clusters

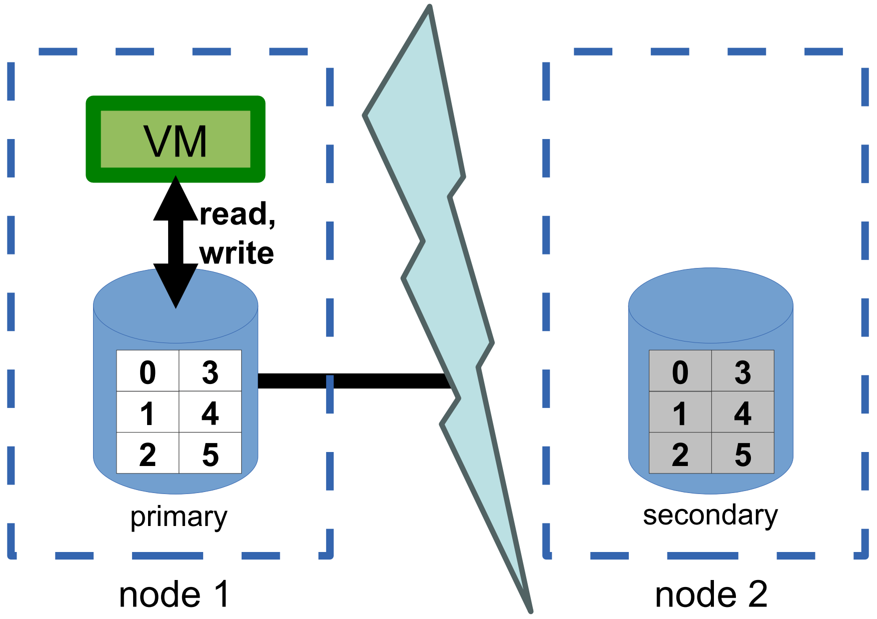
Dangers of clusters

- Do clusters give you "fault tolerance"? "high availability"?
- Maybe…
- There are some fundamental risks you must be aware of
- Number one is "split brain"
- Consider the following scenario:
  - Node 1 is running an application in a VM which stores data
  - Node 2 is a mirror image kept up-to-date in real time

VM with standby



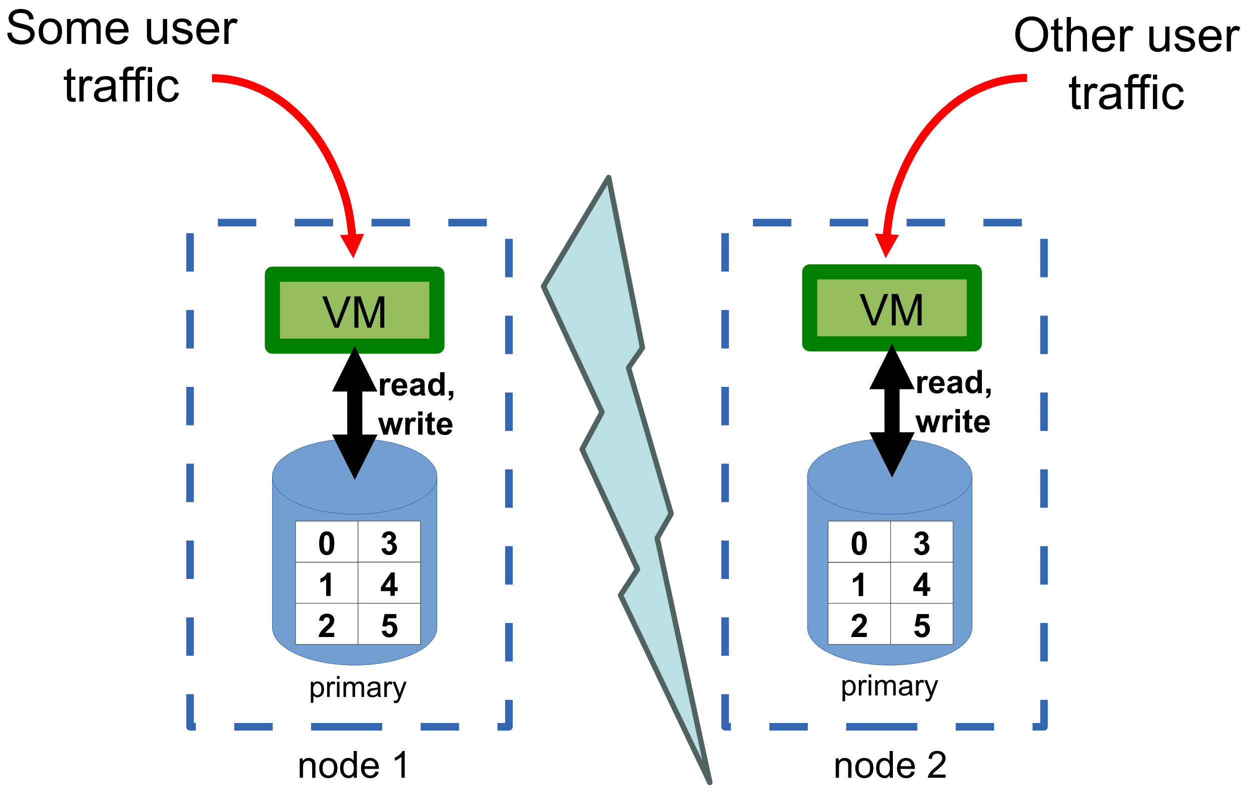
Network partition



node 2 says…

- *I can't see node 1 any more!*
- *It must have gone down!*
- *I'd better start up my local copy of the VM!* (High Availability)
- …but what if node 1 is actually still running?
  - maybe it's just running very slowly, and not responding to heartbeats
  - maybe there's a network problem between the nodes

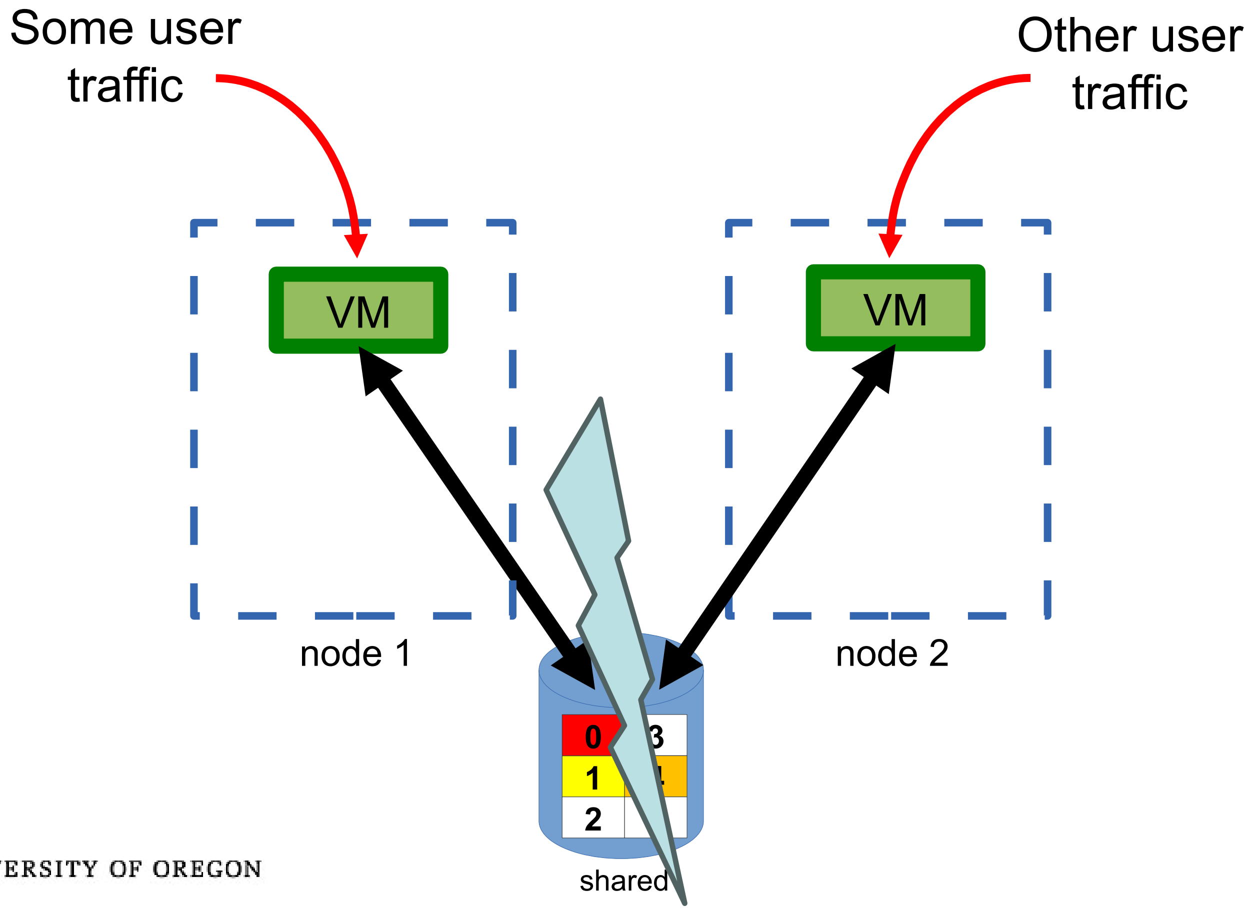
Split brain!


Some user traffic
Other user traffic
VM
read, write
0
3
1
4
2
5
primary
node 1
VM
read, write
0
3
1
4
2
5
primary
node 2

Split brain!

- Now we have two independent VMs running
- Some transactions are recorded by node 1, some by node 2
  - The two sides diverge
  - *Neither* side has a complete, correct view of the world any more
  - If you sync node 1 to node 2, or vice versa, *you will lose data*
- Very, very hard to recover from
- If the two running VMs are using *shared* block storage, it's even worse…

Filesystem corruption!



Beware of any "HA" automatic start

- There is only one safe way to do this, and it involves node 2 physically turning off the power to node 1
  - **STONITH**: "Shoot The Other Node In The Head"
  - Guarantees no split brain
  - Requires hardware support (controllable power bars)
- The general term for this subject area is "fencing"
- Also remember, servers don't actually fail very often
  - Most big outages are due to software errors or misconfiguration
  - See examples from Cloudflare, Microsoft, Google, AWS etc

Alternatives to "HA" automatic start

- Run multiple, discrete instances of your application "live-live"
  - For some application like DNS, this is a trivial
  - You can put a load-balancer in front which does health checks to direct traffic to working node(s)
  - Or pair of load-balancers with a virtual IP that moves between them
- Still need to deal with state
  - e.g. there may be a SQL database, or other shared data storage, which you still need to manage
  - or else your application needs to be written for "eventual consistency", where changes from independent nodes are merged into a common view

The CAP Theorem

- Consistency, Availability, or Partition Tolerance: you can only have two. This is a fundamental limit.
  - Consistency = never reading out-of-date information
  - Availability = system can accept read and write requests
  - Partition = when nodes not able to communicate with each other

Quorum

- The cluster database could also suffer split brain; we want to avoid this
- There are protocols to ensure consistency, e.g. Raft, Paxos, Totem
- They require a *quorum* of nodes to be online and communicating to agree on cluster state
  - quorum is *more than half:* n/2 + 1 (rounded down)
  - e.g. for a 3 node cluster, the quorum is 2. For a 4 or 5 node cluster, the quorum is 3
  - in a network partition, the side which has quorum can continue

Cluster availability

- Quorum systems are "CP": they give Consistency in face of Partitions, at the expense of Availability
- If you have a 3 node system, and 2 nodes fail, it will stop working
  - it will be impossible to operate the remaining node
- By this measure, a cluster of 3 nodes has *lower* availability than 3 individual nodes
- It's a trade-off
  - if you want cluster features like live migration and global management
- To build a 2-node cluster, you should add a third (tiny) node

Proxmox Cluster and VM Migration Lab

UNIVERSITY OF OREGON

NSRC
Network Startup Resource Center