Migrating to Linstor: a sweet spot for VM block storage

Brian Candler

Network Startup Resource Center

UNIVERSITY OF OREGON

NSRC

Network Startup Resource Center

Building a VM platform

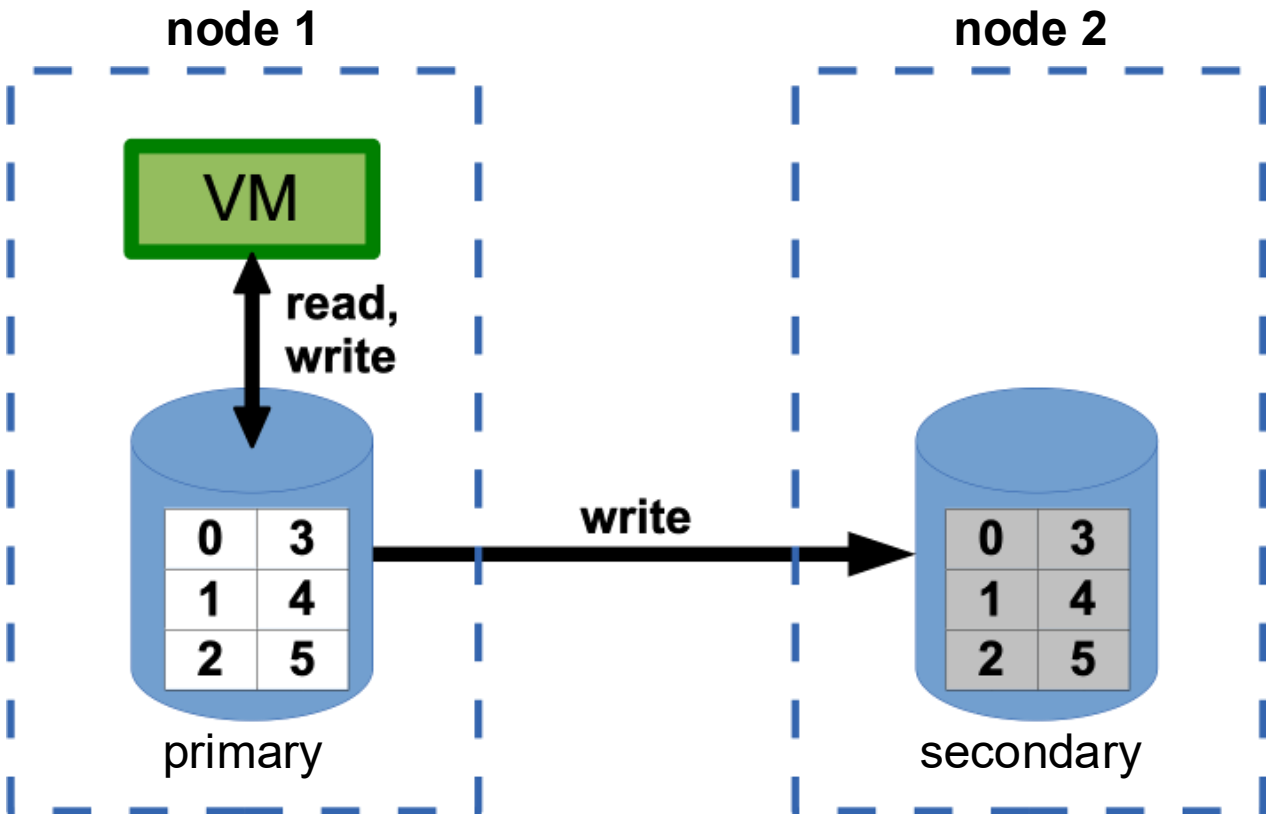
- Scaling the CPU is easy
- Scaling the RAM is easy
- But what do you do about **storage?**

Traditional storage options

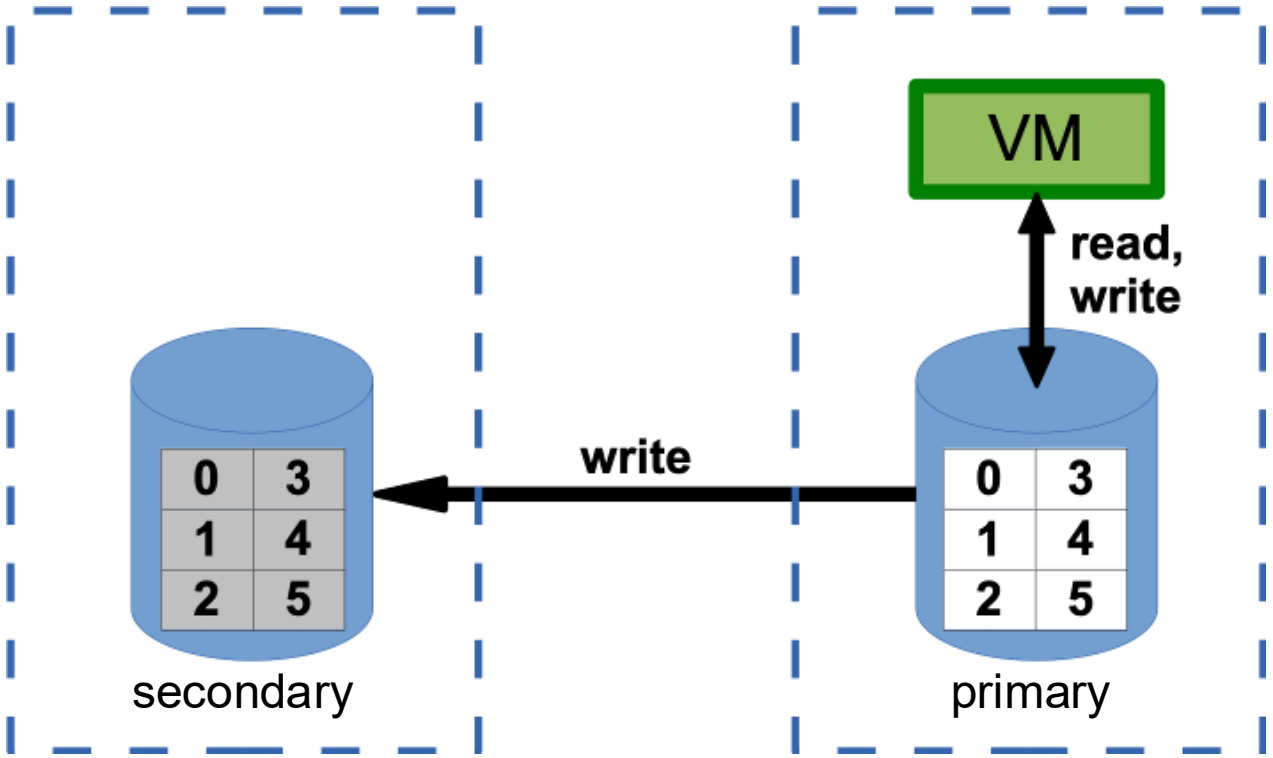
- Local storage (with some sort of RAID)
  - Migrations require copying the entire volume (slooooow)
  - Loss of node = loss of VMs on that node
- SAN or NAS
  - The storage server/network is a single point of failure. $$$
- Distributed software-defined storage e.g. Ceph
  - There are failure modes that can lock up the whole system
  - Performance issues are hard to diagnose
  - Your data is stored in opaque blobs

Middle ground: block-level replication

- "RAID1 over the network": access local disk, writes replicated

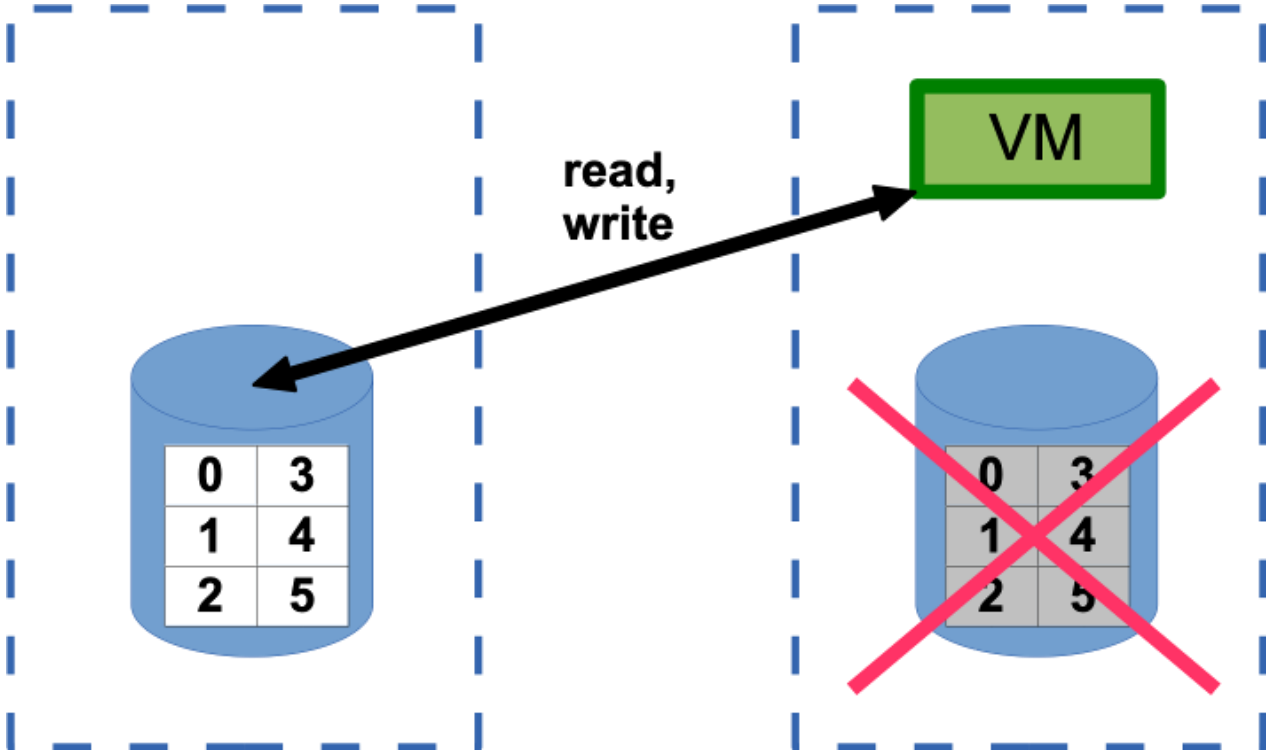


Reverse roles for VM migration



Disk failure

- Transparently redirect all I/O over network



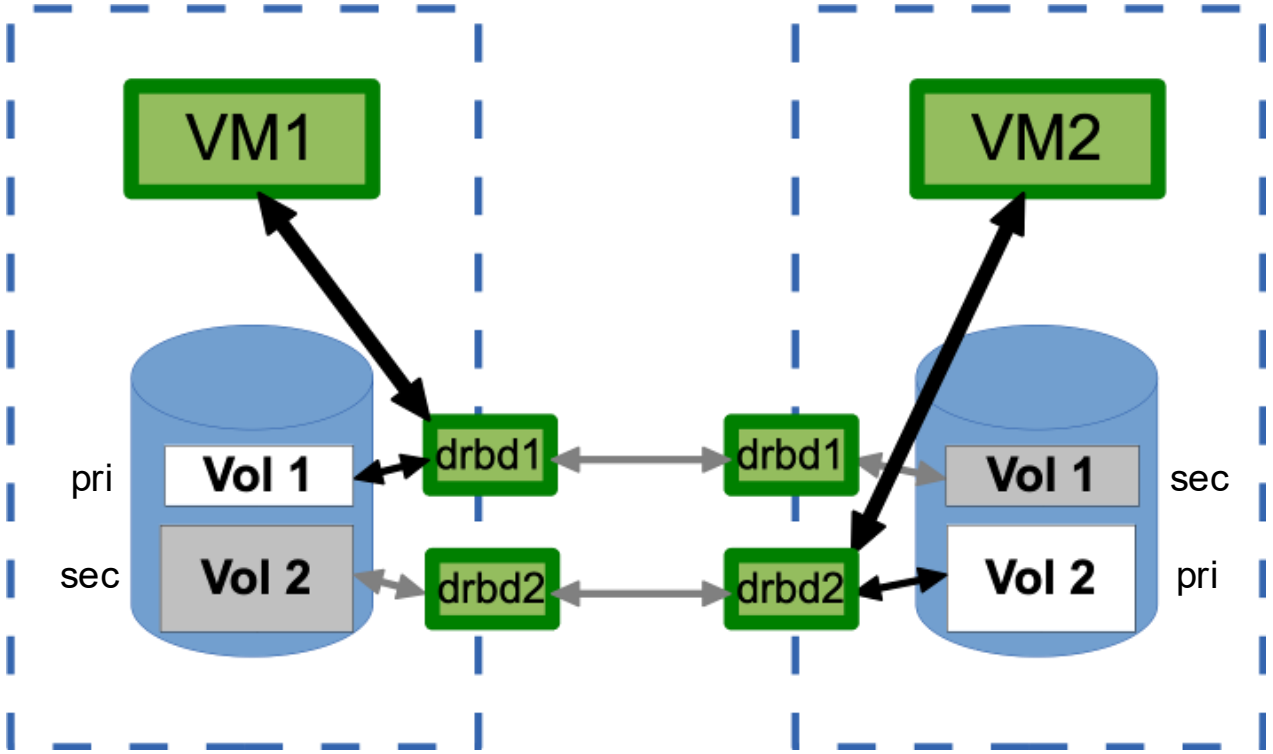
(Catch up later when disk replaced)

DRBD: Distributed Replicated Block Device

- Has been part of the Linux kernel for many years
  - Hugely robust and well tested. It's "only" mirroring…
- Your data is stored **as-is** on the underlying storage
  - Plus a little bit of metadata at the end to track replication state
- Out-of-the-box uses **static** configuration in drbd.conf, e.g. mirror /dev/sda1 on one machine to /dev/sda1 on another
- For VMs we'd like a separate DRBD instance *per virtual machine*, so they can be migrated individually

Layered DRBD

- DRBD-over-LVM, or DRBD-over-ZVOL



Old implementation: **Ganeti**

- A VM platform used in-house by Google in the early days
- Uses DRBD 8
- Thrown over the wall as an open-source project
- Very small community, little development
- CLI/API only
- Mixture of Python and Haskell

Modern alternative: **Linstor**

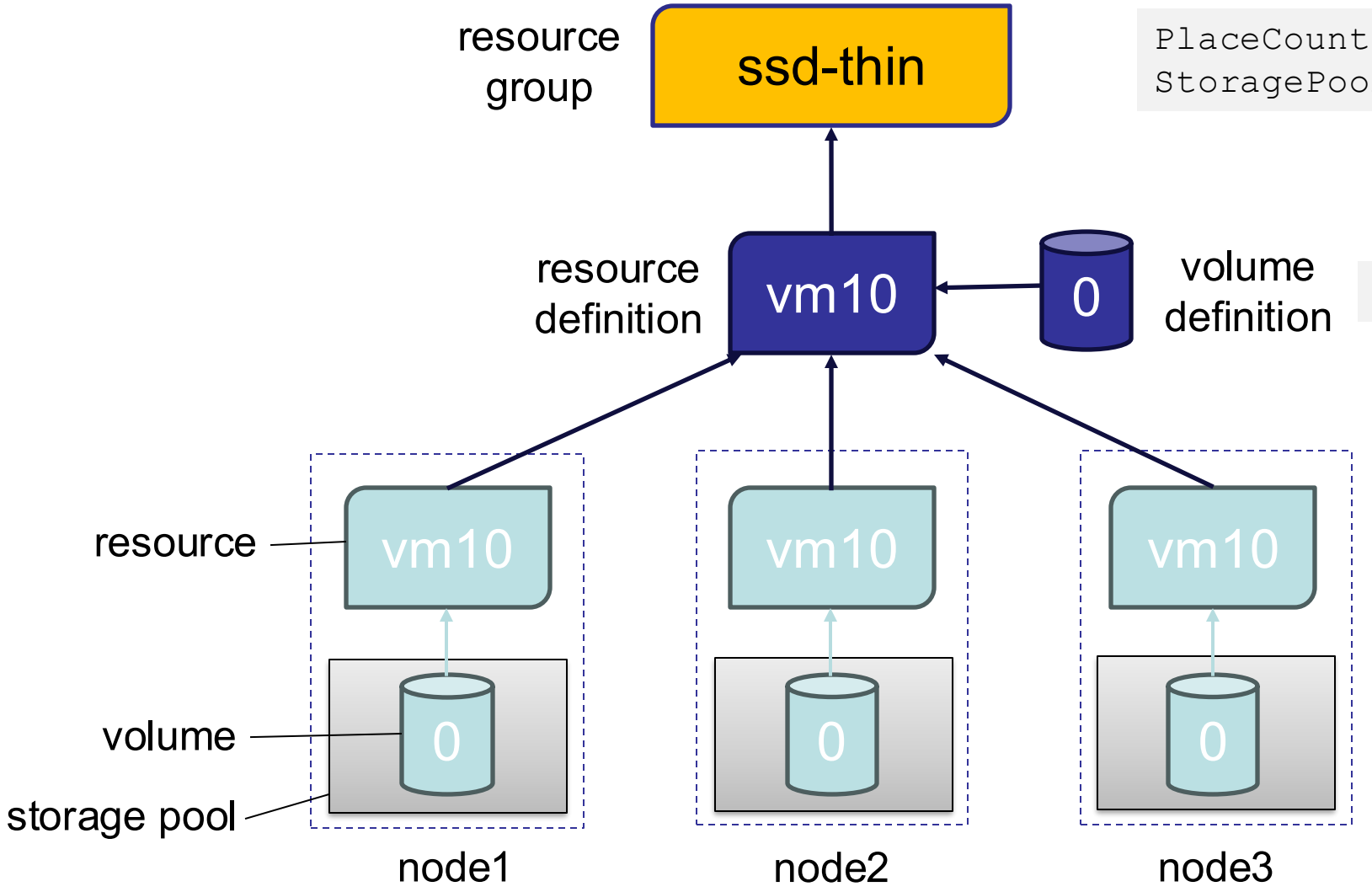
- Written by Linbit, the originators of DRBD
- Management layer on top of DRBD 9
  - Allows up to 32 replicas of each volume
  - Allows diskless clients (i.e. VM can run even where there's no replica)
- Free and open source, but commercial support available
- Integrations for Proxmox VE, Kubernetes, OpenStack etc
- Prometheus metrics and Grafana dashboard
- Why is this not more widely known??

Linstor architecture

- A "satellite" on each node, plus a central "controller" with database
- All in Java ☹
  - But it stays completely out of the data path
  - Even if you restart it, replication is completely unaffected
- Some initial terminology to learn
  - A "resource definition" + "volume definition" is effectively a "virtual disk"
  - A "resource" + "volume" is one of the mirrors of that disk, on a node
  - Allocates space from a "storage-pool" (LVM volume group or ZFS data set)
  - A "resource group" provides inherited settings, e.g. number of replicas

https://brian-candler.medium.com/linstor-networked-storage-without-the-complexity-c3178960ce6b
https://brian-candler.medium.com/linstor-concepts-and-configuration-e5b0c8e10d27

Linstor configuration



NSRC migration

- We decided to migrate from Ganeti to Proxmox VE + Linstor
- Straightforward migration plan:
  - Build one new Proxmox VE node
  - Move some VMs (using simple "dd" copy)
  - Evacuate and remove one Ganeti node and convert it into PVE
  - Repeat until finished

```
# gnt-instance shutdown foo.nsrc.org
# gnt-instance activate-disks foo.nsrc.org
virtual4.nsrc.org:disk/0:/dev/drbd2

# ssh root@virtual4 gzip -c1 /dev/drbd2 |
  gzip -dc | dd of=/dev/drbd1003 conv=sparse bs=4M
```

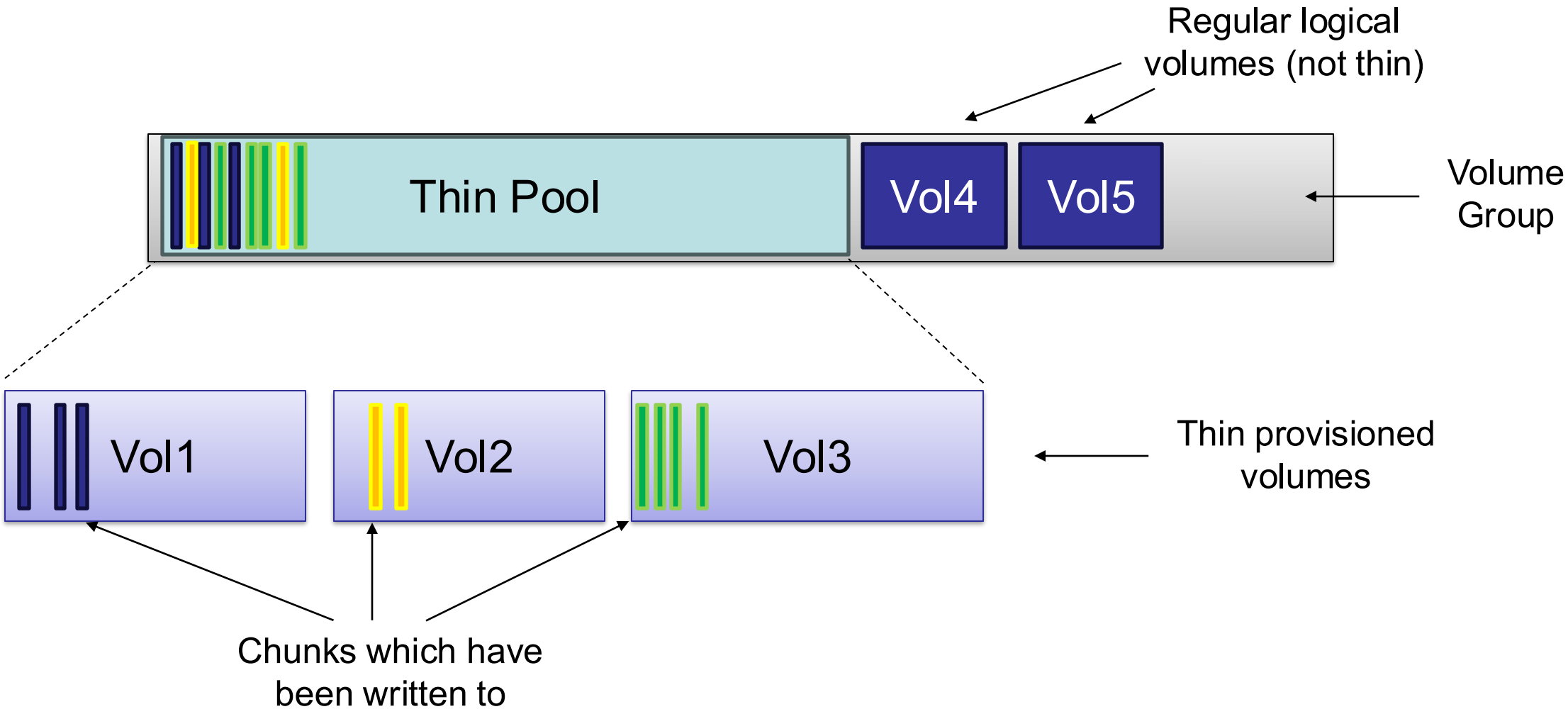
UNIVERSITY OF OREGON

NSRC
Network Startup Resource Center

Design Considerations

- Use LVM or ZVOL storage pools underneath DRBD?
  - We chose LVM
  - Avoids some potential performance problems with ZFS as a block device, e.g. ZFS guest VM running inside ZFS ZVOL
- Do you want snapshots? If using LVM, both Proxmox VE and Linstor only support snapshots when using **thin pools**

LVM thin provisioning



Notes about LVM thin pools

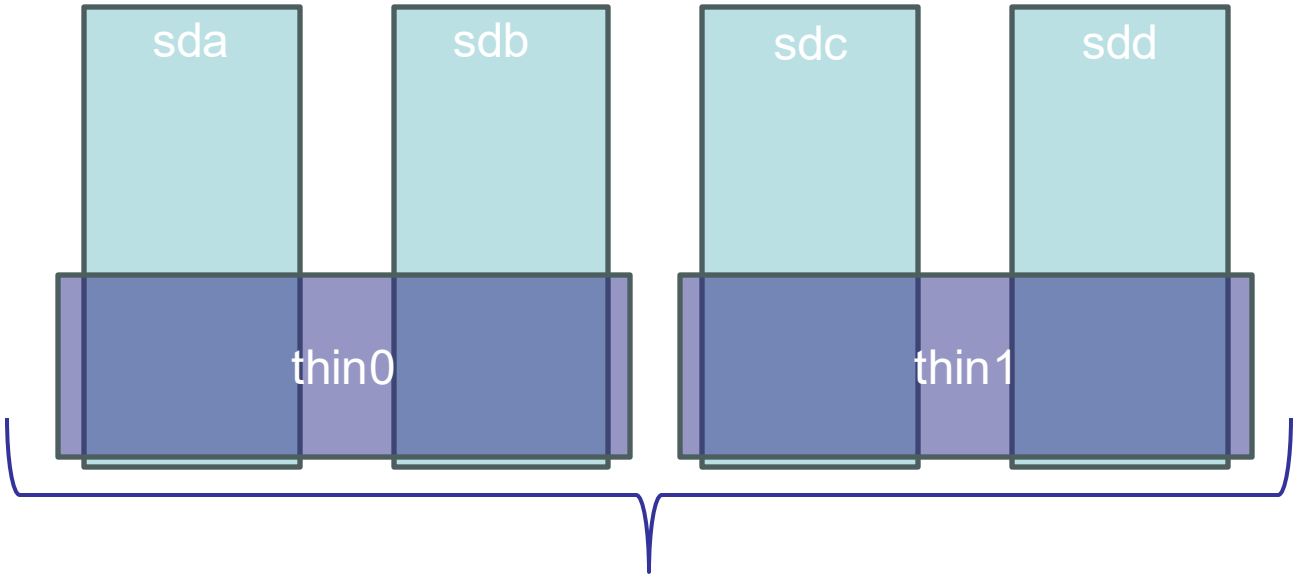
- "Lazy" allocation of space: allocates 64KiB chunks on first write
- Lets you overcommit, but there's a risk of running out of space
  - You can grow the thin pool if required (shrinking is hard though)
- If the thin pool is allocated over multiple disks, these delayed writes mean that some chunks of your VM are liable to be appear on all disks
- Hence an increased "blast radius" if a single disk fails
- Solution? Create multiple thin pools, one per disk

Design Chosen

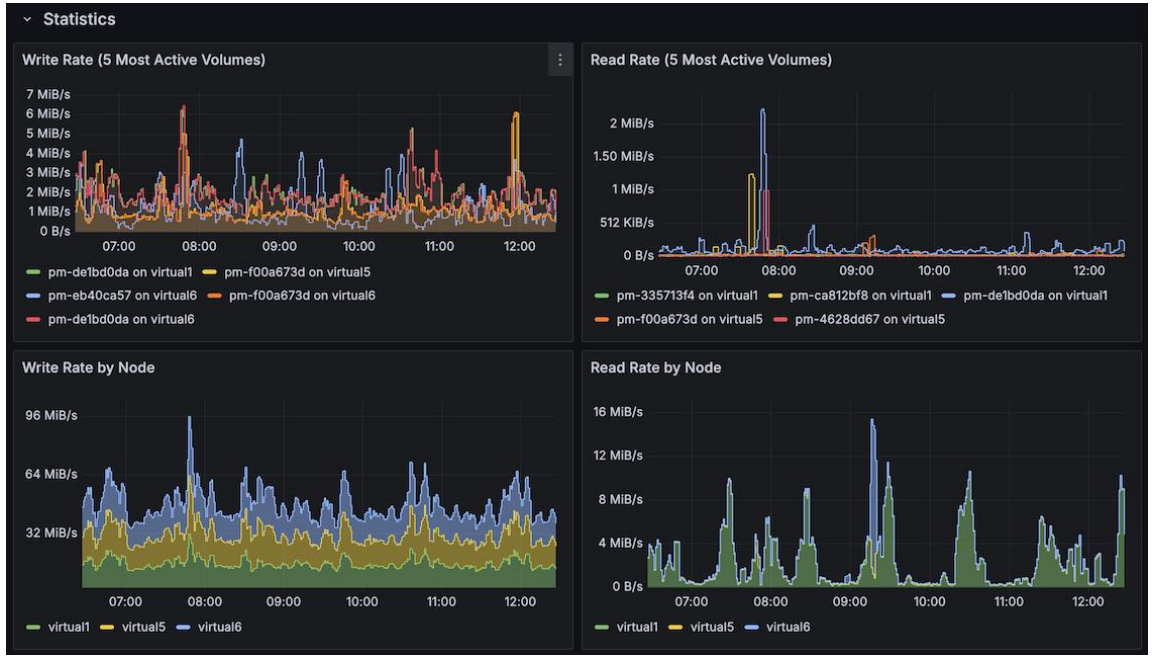
- We created two Linstor resource groups
- **linstor-thick** uses the underlying LVM volume group directly
  - Used for large VMs which have ZFS inside, and their own snapshotting
  - We use these for VMs hosting incus containers
  - 3-way DRBD replication (place-count=3)
- **linstor-thin** uses pre-created thin pools
  - Used for smaller VMs which need Proxmox-level snapshotting

Our chosen approach

- We are currently using 4 SSDs per node
- Decided to make two mirrored thin pools using LVM mirroring
  - Gives us some local redundancy (+ we use 2-way DRBD on top)
  - Constrains all the chunks for a single VM to be on the same disk pair



Prometheus & Grafana Integration



Some fiddling with Prometheus scraping to make the dashboard work properly
https://forums.linbit.com/t/doc-note-deploying-linstor-with-proxmox/779/5

Getting Linstor and DRBD 9

- Free repository for Debian, including Proxmox plugin

  http://packages.linbit.com/public/ proxmox-<vers> drbd-9
- Free repository for Ubuntu

  https://launchpad.net/~linbit/+archive/ubuntu/linbit-drbd9-stack
- Production-grade repositories for various distros, including RHEL-based ones, available by subscription
- Or you can build yourself from source

Conclusion

- Proxmox VE + Linstor = match made in heaven
- A platform we can easily manage and understand
- Actively maintained, support available if we wanted it
- Good live VM mobility
- Minimal management overhead
  - But keep an eye on the size of the thin pools!

https://nsrc.org/workshops/2025/btnog12/nsrc-btnog-virt/
Detailed Workshop Agenda > materials on LVM and Linstor