Resilient and Fast Persistent Container Storage
Leveraging Linux’s Storage Functionalities
Philipp Reisner, CEO LINBIT
COMPANY OVERVIEW

- Developer of DRBD
- 100% founder owned
- Offices in Europe and US
- Team of 30 highly experienced Linux experts
- Partner in Japan

REFERENCES

Google  SanDisk  Intel  Cisco  IBM  Micro Focus  Red Hat  NTT
Linux Storage Gems

LVM, RAID, SSD cache tiers, deduplication, targets & initiators
Linux's LVM

Volume Group

- logical volume
- snapshot
- logical volume
- physical volume
- physical volume
- physical volume
Linux's LVM

- based on device mapper
- original objects
  - PVs, VGs, LVs, snapshots
  - LVs can scatter over PVs in multiple segments
- thinlyv
  - thinpools = LVs
  - thin LVs live in thinpools
  - multiple snapshots became efficient!
Linux's LVM
Linux's RAID

- original MD code
  - `mdadm` command
  - Raid Levels: 0,1,4,5,6,10
- Now available in LVM as well
  - device mapper interface for MD code
  - do not call it ‘dmraid’; that is software for hardware fake-raid
- `lvcreate --type raid6 --size 100G VG_name`
SSD cache for HDD

• dm-cache
  • device mapper module
  • accessible via LVM tools

• bcache
  • generic Linux block device
  • slightly ahead in the performance game
Linux’s DeDupe

- Virtual Data Optimizer (VDO) since RHEL 7.5
  - Red hat acquired Permabit and is GPLing VDO
- Linux upstreaming is in preparation
- in-line data deduplication
- kernel part is a device mapper module
- indexing service runs in user-space
- async or synchronous writeback
- Recommended to be used below LVM
Linux’s targets & initiators

- Open-ISCSI initiator
- ietd, STGT, SCST
  - mostly historical
- LIO
  - iSCSI, iSER, SRP, FC, FCoE
  - SCSI pass through, block IO, file IO, user-specific-IO
- NVMe-OF
  - target & initiator
ZFS on Linux

- Ubuntu eco-system only
- has its own
  - logic volume manager (zVols)
  - thin provisioning
  - RAID (RAIDz)
  - caching for SSDs (ZIL, SLOG)
  - and a file system!
Put in simplest form
DRBD – think of it as ...

- RAID1

- Initiator

- Target

IO-requests → data/completion
DRBD Roles: Primary & Secondary

Primary ➔ replication ➔ Secondary
DRBD – multiple Volumes

• consistency group
DRBD – up to 32 replicas

- each may be synchronous or async
DRBD – Diskless nodes

- intentional diskless (no change tracking bitmap)
- disks can fail
DRBD - more about

- a node knows the version of the data is exposes
- automatic partial resync after connection outage
- checksum-based verify & resync
- split brain detection & resolution policies
- fencing
- quorum
- multiple resources per node possible (1000s)
- dual Primary for live migration of VMs only!
DRBD Roadmap

- performance optimizations (2018)
  - meta-data on PMEM/NVDIMMS
  - zero copy receive on diskless (RDMA-transport)
  - no context switch send (RDMA & TCP transport)
- Eurostars grant: DRBD4Cloud
  - erasure coding (2019)
The combination is more than the sum of its parts
LINSTOR - goals

- storage build from generic (x86) nodes
- for SDS consumers (K8s, OpenStack, OpenNebula)
- building on existing Linux storage components
- multiple tenants possible
- deployment architectures
  - distinct storage nodes
  - hyperconverged with hypervisors / container hosts
- LVM, thin LVM or ZFS for volume management (stratis later)
- Open Source, GPL
LINSTOR w. failed Hypervisor

LIN:BIT

- hypervisor
  - VM
  - VM

- storage node

- DRBD

- hypervisor
  - VM
  - VM

- storage node

- DRBD

- storage node

- DRBD

- storage node
LINSTOR w. failed storage node

- Hypervisor
- VM
- DRBD
- Storage node

- Hypervisor
- VM
- DRBD
- Storage node

- Hypervisor
- VM
- DRBD
- Storage node
LINSTOR - Hyperconverged
LINSTOR - VM migrated
LINSTOR - add local storage

Diagram showing hypervisor & storage connections with VMs.
LINSTOR - remove 3rd copy

- hypervisor & storage
- hypervisor & storage
- hypervisor & storage
- VM

- hypervisor & storage
- hypervisor & storage
- hypervisor & storage
- VM

- hypervisor & storage
- hypervisor & storage
- hypervisor & storage
LINSTOR Architecture
LINSTOR Roadmap

- Swordfish API
  - volume management
  - access via NVMe-oF
  - inventory sync from Redfish/Swordfish
- support for multiple sites & DRBD-Proxy (Dec 2018)
- north bound drivers
  - Kubernetes, OpenStack, OpenNebula, Proxmox, XenServer
Case study - intel

Intel® Rack Scale Design (Intel® RSD) is an industry-wide architecture for disaggregated, composable infrastructure that fundamentally changes the way a data center is built, managed, and expanded over time.

LINBIT working together with Intel

LINSTOR is a storage orchestration technology that brings storage from generic Linux servers and SNIA Swordfish enabled targets to containerized workloads as persistent storage. LINBIT is working with Intel to develop a Data Management Platform that includes a storage backend based on LINBIT’s software. LINBIT adds support for the SNIA Swordfish API and NVMe-oF to LINSTOR.
Thank you

https://www.linbit.com