Optimizing Longhorn for High-Performance Hardware

Software Defined Storage Devroom, FOSCOM 2025

Konstantinos Kampadais, Antony Chazapis, Angelos Bilas • FORTH-ICS/CARV

Kubernetes, the container orchestration platform

- A cloud *operating system*?
  - Resource management
  - Networking
  - Failures
- The thin line between hardware and software—an abstraction
  - Primitives and conventions
  - Managed platform for development and deployment
- Portability of operations across platforms (with exceptions, as always)
  - Local (minikube, MikroK8s, k3s, kind, ...)
  - Cloud (Amazon EKS, Azure AKS, Google GKE, DigitalOcean Kubernetes, ...)
- Many extensions and third-party tools

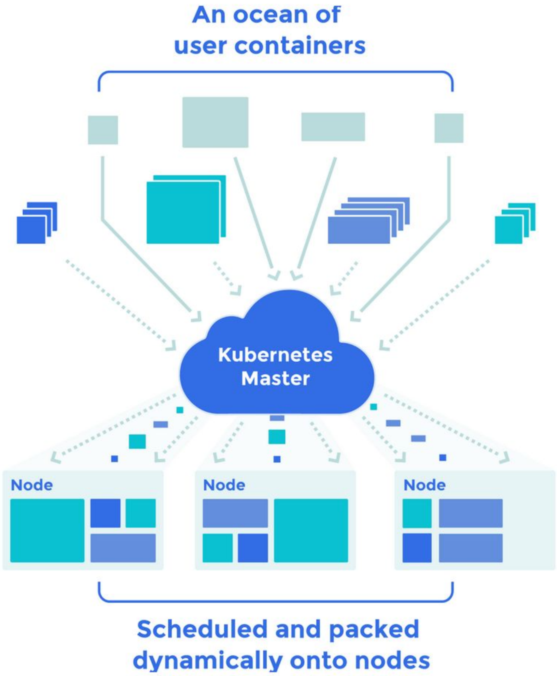

An ocean of user containers
Kubernetes Master
Node
Node
Node
Scheduled and packed dynamically onto nodes

Image source: https://discuss.newrelic.com/t/relic-solution-what-you-need-to-know-about-new-relic-when-deploying-with-docker/52492

Volume objects in Kubernetes

- The standard API uses PersistentVolumeClaims and PersistentVolumes
- A PersistentVolume represents actual storage space
    - Many StorageClasses (from one or more plugins) may be available
- A PersistentVolumeClaim is a request for storage → PVCs consume PVs

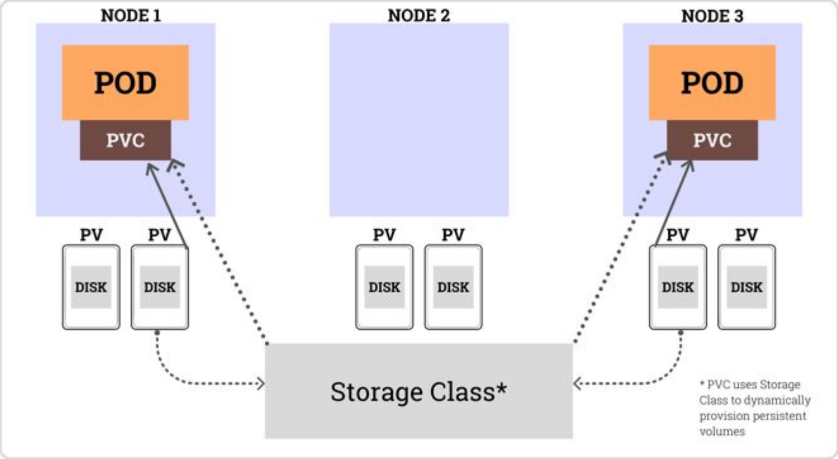


Image source: https://blog.mayadata.io/kubernetes-storage-basics-pv-pvc-and-storageclass

Container Storage Interface (CSI)

- API to interface with storage offerings
  - Implements the lifecycle of volumes and block devices
  - Adds/removes storage to a container
  - Supports snapshots, volume cloning
- Started as an effort to remove storage implementations from the Kubernetes source
- Many implementations → Mostly interfaces to external storage providers

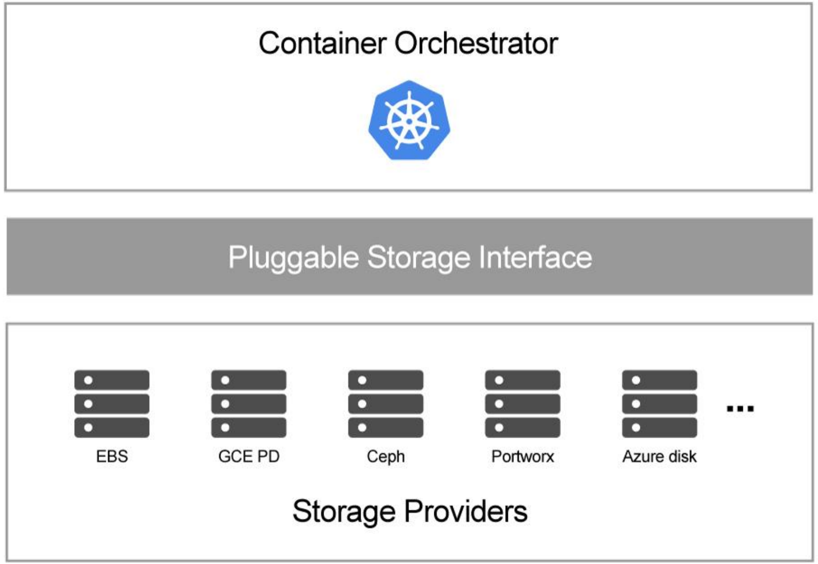
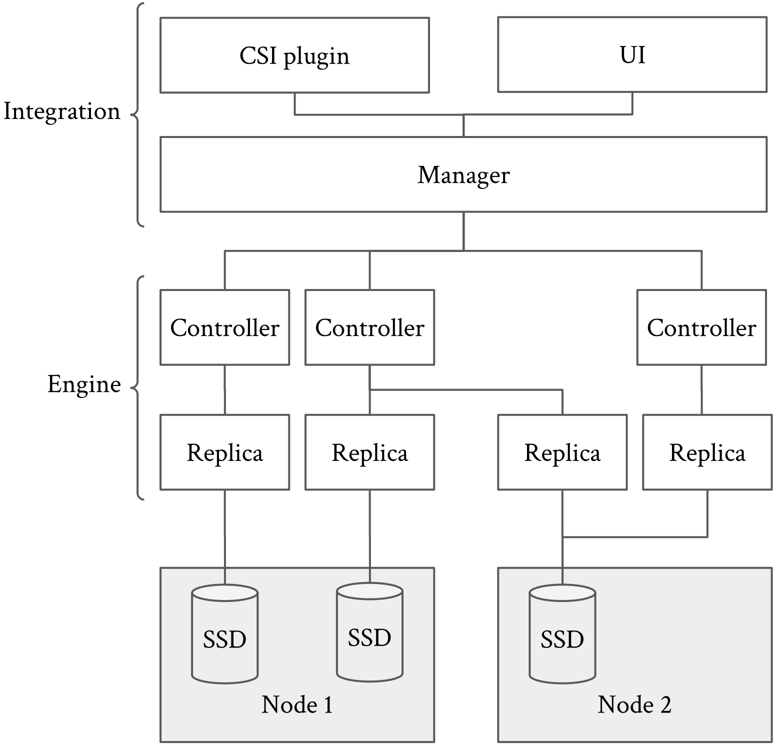


Image source: https://www.tuxera.com/blog/examining-container-storage-interface-csi-support-fusion-file-share-by-tuxera-in-kubernetes/

Longhorn

- Open-source, part of CNCF
- Complete software defined storage engine
  - “World’s smallest storage controller”
  - No reliance to a third-party storage provider
  - Advanced features (snapshots, backups)
- Modular architecture
  - CSI plugin → Interface to Kubernetes
  - UI → User-friendly operations dashboard
  - Manager → Control plane, volume orchestrator
  - Engine → Volume implementation, I/O path
- Each volume is implemented by one controller and multiple replicas

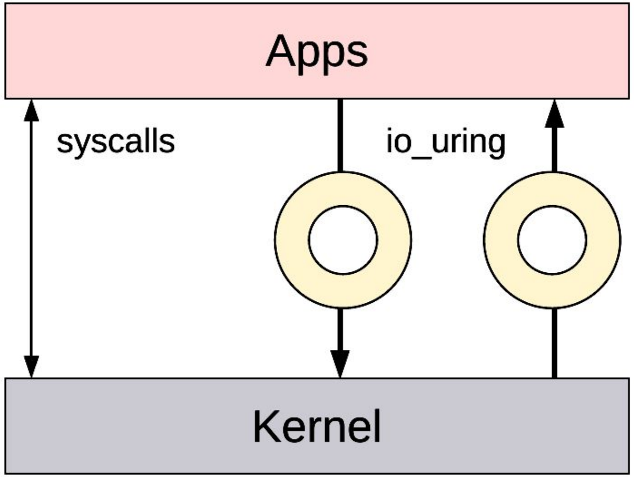


Issues and approach

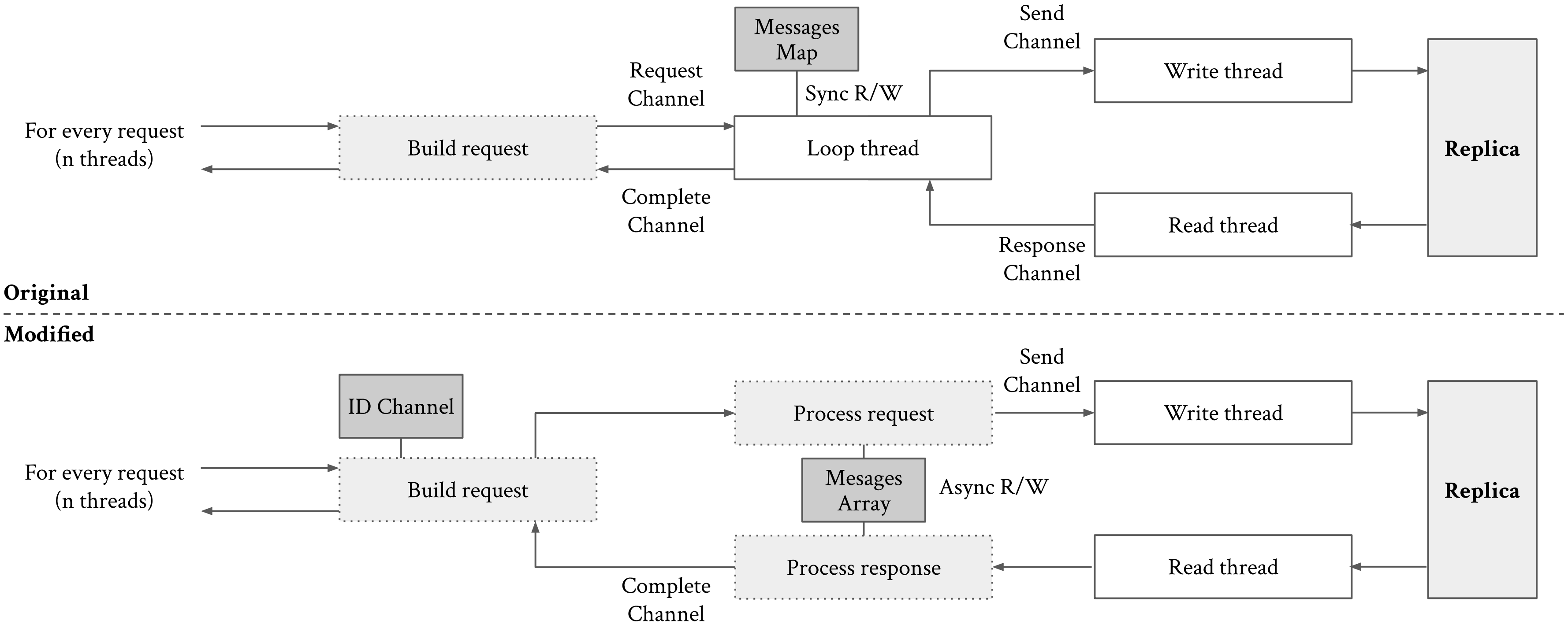
- Performance ok for cloud servers, but not on-prem → high-speed network and NVMe
  - Limited to about 50k/25k read/write IOPS, where the device supports > 400k IOPS (dev setup)
  - Cloud VMs usually have performance limits (~40k @ AWS, unless you pay more…)
- Isolate layers and identify bottlenecks at each step
  - Frontend → iSCSI implementation too slow (based on TGT)
  - Controller → Controller-replica communication protocol implementation serializes all operations
  - Replica → Sparse-file implementation limits write performance (especially with multiple snapshots)
- Explore alternatives
  - Frontend → Use ublk based on io_uring (available in Linux 6.x, Ubuntu 24.10)
  - Controller → Reimplement controller-replica communication
  - Replica → Implement Direct Block Store (DBS), a custom, direct-to-disk storage layer

ublk and io_uring

- io_uring is a Linux kernel system call interface
  - Supports asynchronous operations
  - Uses two circular buffers shared between the kernel and application
  - Extremely fast! (batching and less memory copies)
- ublk is a generic userspace block device leveraging io_uring technology
  - ublk driver in the kernel → Creates virtual device
  - ublksrv in userspace → Implements I/Os (modular)
  - Used for I/Os and admin tasks (add, remove devices)

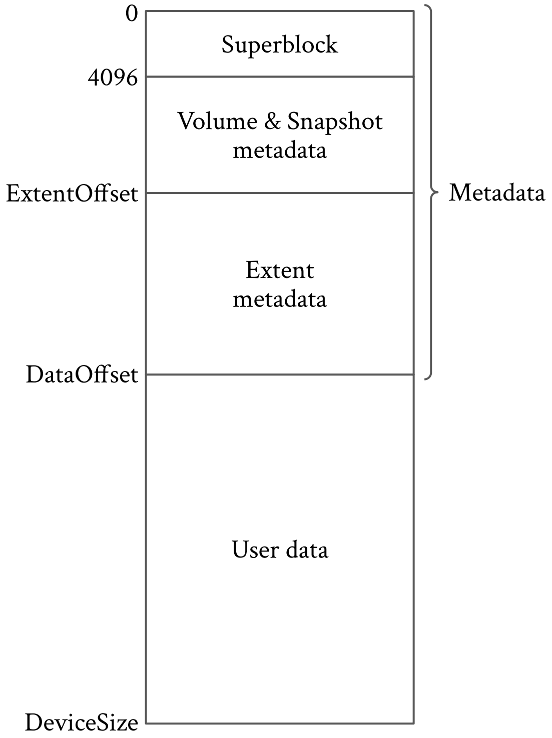


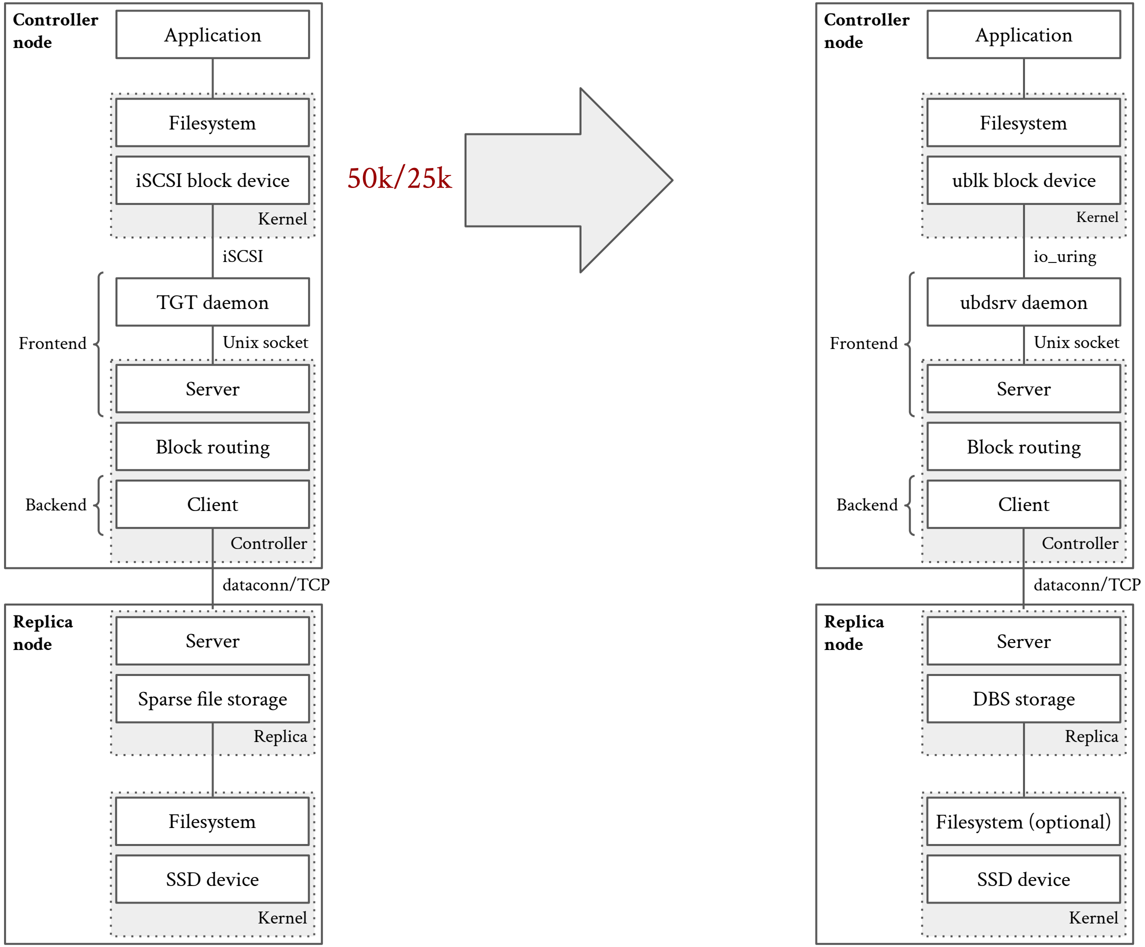
Controller-replica communication

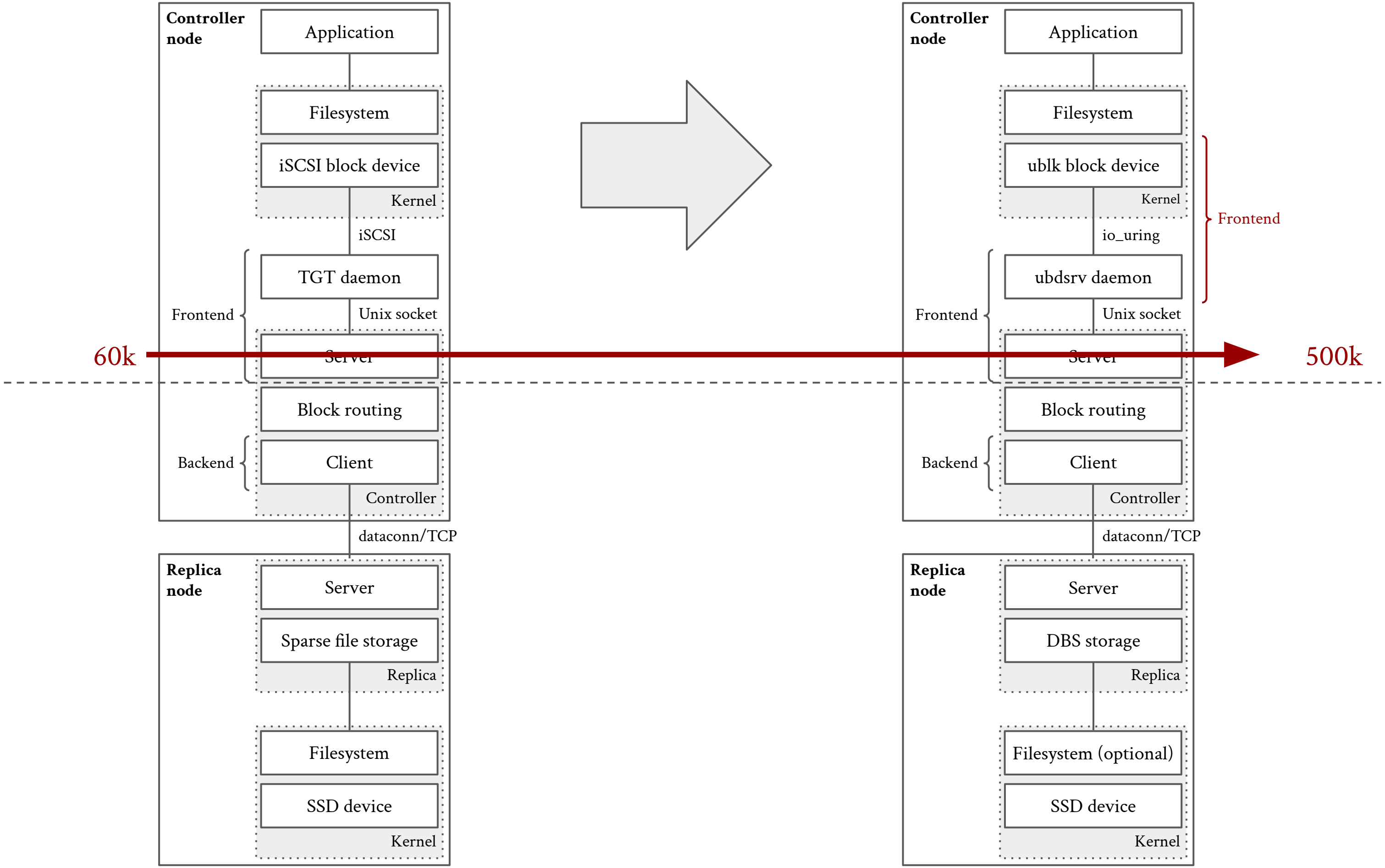


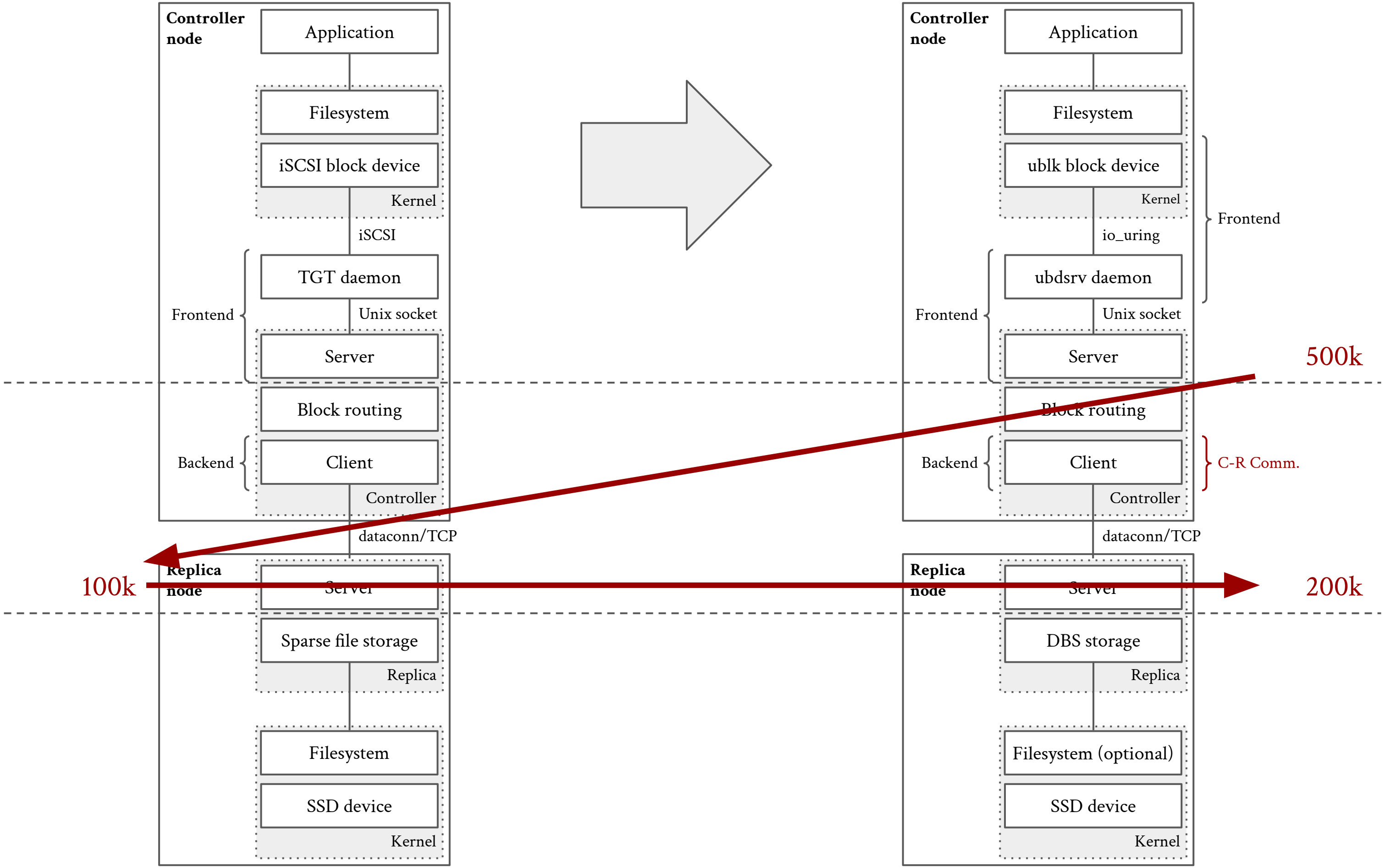
Direct Block Store (DBS)

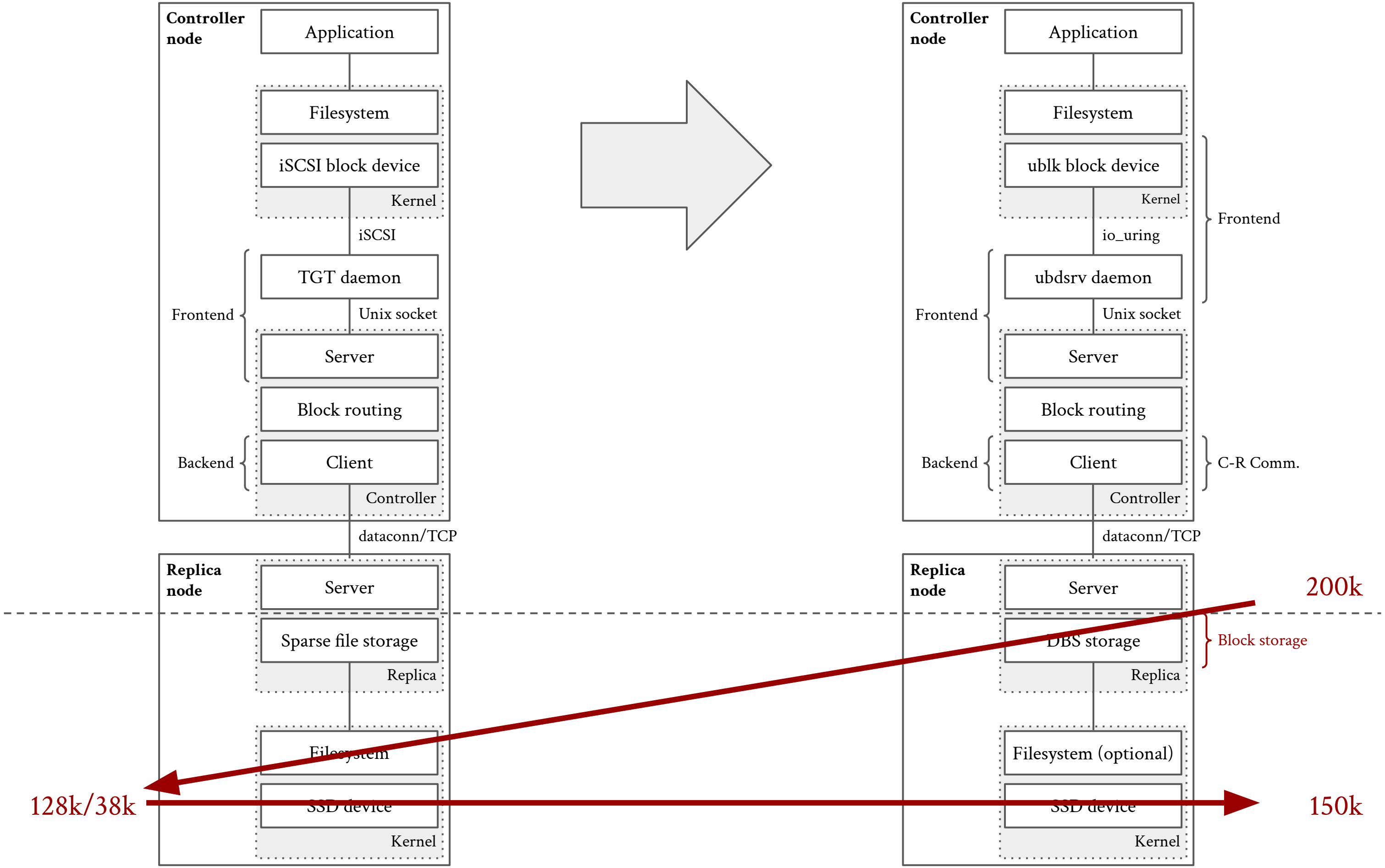
- Can use a file or directly a device
- Supports multiple volumes, snapshots per volume (extensive API)
- Divides storage in four regions
  - Superblock
  - Volume & snapshot metadata
  - Extent metadata
  - User data → Actual blocks
- Light-weight and fast
  - Volume (snapshot) extent maps are kept in memory (~40 MB for 1 TB volume)
  - Extensive use of bitmaps
- Written in Golang and open source

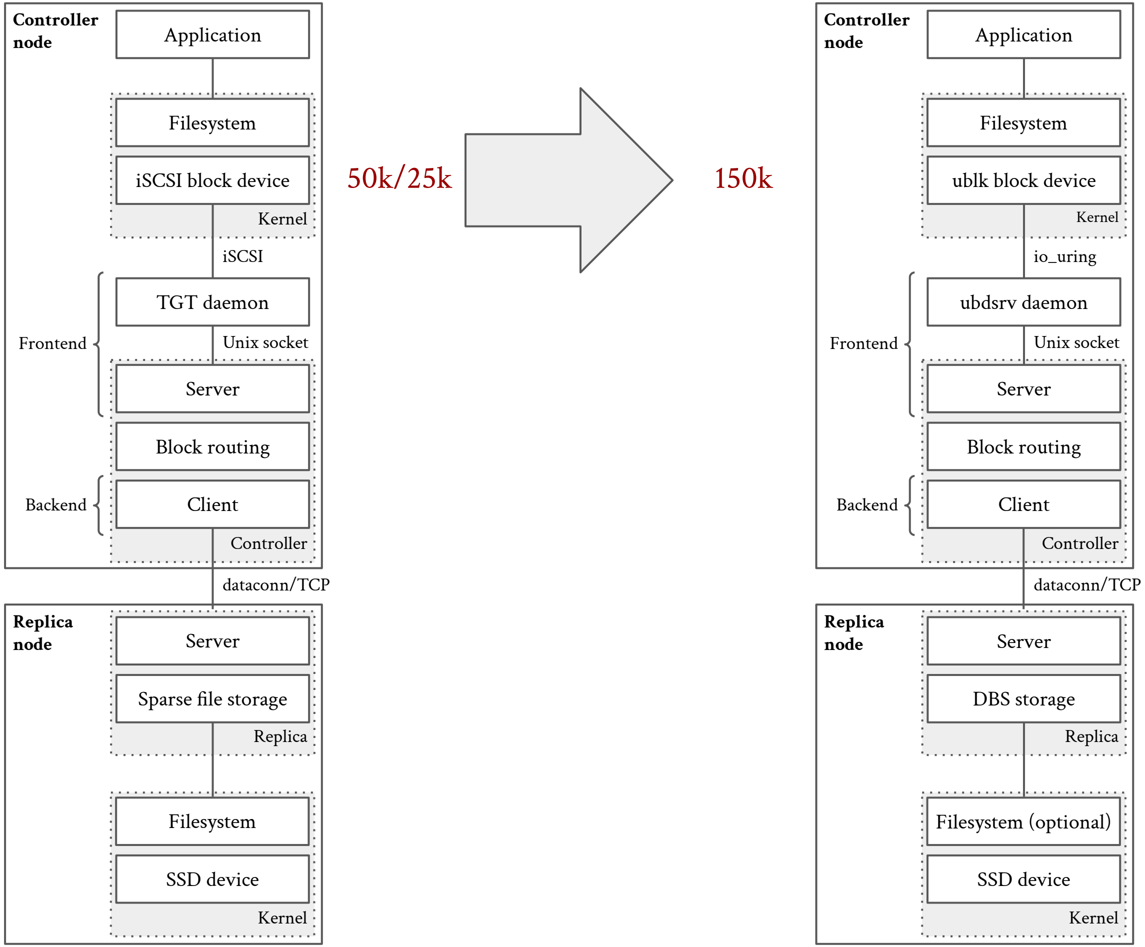



Controller node
Application
Filesystem
iSCSI block device
Kernel
iSCSI
TGT daemon
Frontend
Unix socket
Server
Block routing
Backend
Client
Controller
dataconn/TCP
Replica node
Server
Sparse file storage
Replica
Filesystem
SSD device
Kernel
50k/25k
Controller node
Application
Filesystem
ublk block device
Kernel
io_uring
ubdsrv daemon
Frontend
Unix socket
Server
Block routing
Backend
Client
Controller
dataconn/TCP
Replica node
Server
DBS storage
Replica
Filesystem (optional)
SSD device
Kernel


Controller node
Application
Filesystem
iSCSI block device
Kernel
iSCSI
TGT daemon
Frontend
Unix socket
Server
60k
Block routing
Backend
Client
Controller
dataconn/TCP
Replica node
Server
Sparse file storage
Replica
Filesystem
SSD device
Kernel
Controller node
Application
Filesystem
ublk block device
Kernel
io_uring
ubdsrv daemon
Frontend
Frontend
Unix socket
Server
500k
Block routing
Backend
Client
Controller
dataconn/TCP
Replica node
Server
DBS storage
Replica
Filesystem (optional)
SSD device
Kernel


Controller node
Application
Filesystem
iSCSI block device
Kernel
iSCSI
Frontend
TGT daemon
Unix socket
Server
Block routing
Backend
Client
Controller
dataconn/TCP
Replica node
Server
Sparse file storage
Replica
Filesystem
SSD device
Kernel
100k
Controller node
Application
Filesystem
ublk block device
Kernel
io_uring
Frontend
ubdsrv daemon
Frontend
Unix socket
Server
500k
Block routing
Backend
Client
C-R Comm.
Controller
dataconn/TCP
Replica node
Server
200k
DBS storage
Replica
Filesystem (optional)
SSD device
Kernel

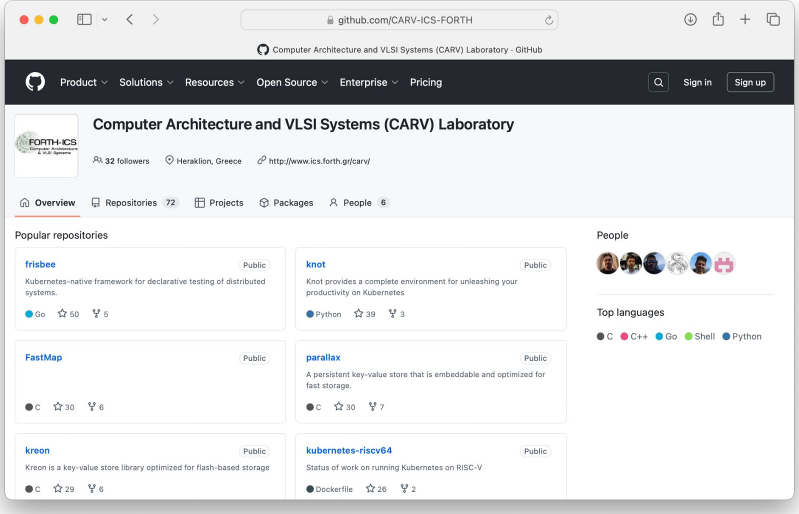

Controller node
Application
Filesystem
iSCSI block device
Kernel
iSCSI
TGT daemon
Frontend
Unix socket
Server
Block routing
Backend
Client
Controller
dataconn/TCP
Replica node
Server
Sparse file storage
Replica
Filesystem
SSD device
Kernel
128k/38k
Controller node
Application
Filesystem
ublk block device
Kernel
io_uring
ubdsrv daemon
Frontend
Frontend
Unix socket
Server
Block routing
Backend
Client
C-R Comm.
Controller
dataconn/TCP
Replica node
Server
200k
DBS storage
Block storage
Replica
Filesystem (optional)
SSD device
Kernel
150k


Controller node
Application
Filesystem
iSCSI block device
Kernel
iSCSI
Frontend
TGT daemon
Unix socket
Server
Block routing
Backend
Client
Controller
dataconn/TCP
Replica node
Server
Sparse file storage
Replica
Filesystem
SSD device
Kernel
50k/25k
150k
Controller node
Application
Filesystem
ublk block device
Kernel
io_uring
Frontend
ubdsrv daemon
Unix socket
Server
Block routing
Backend
Client
Controller
dataconn/TCP
Replica node
Server
DBS storage
Replica
Filesystem (optional)
SSD device
Kernel

Conclusion

- ublk has the biggest impact in accelerating I/O
  - Is NVMe-oF necessary? (especially when over-the-network operations are not necessary)
- Further performance improvements should be possible (esp. in the controller)
  - Working on optimizations and more features
- DBS is not novel, but may prove a helpful utility for other projects

PRs have been submitted, DBS and other projects are available at https://github.com/CARV-ICS-FORTH



Computer Architecture and VLSI Systems (CARV) Laboratory

32 followers · Heraklion, Greece · http://www.ics.forth.gr/carv/

Popular repositories

frisbee Public
Kubernetes-native framework for declarative testing of distributed systems.
Go ☆ 50 ⑂ 5

knot Public
Knot provides a complete environment for unleashing your productivity on Kubernetes
Python ☆ 39 ⑂ 3

FastMap Public
C ☆ 30 ⑂ 6

parallax Public
A persistent key-value store that is embeddable and optimized for fast storage.
C ☆ 30 ⑂ 7

kreon Public
Kreon is a key-value store library optimized for flash-based storage
C ☆ 29 ⑂ 6

kubernetes-riscv64 Public
Status of work on running Kubernetes on RISC-V
Dockerfile ☆ 26 ⑂ 2

People

Top languages
● C ● C++ ● Go ● Shell ● Python

Acknowledgements

REBECCA