FreEBS

Igor Canadi, Rebecca Lam, Jim Paton

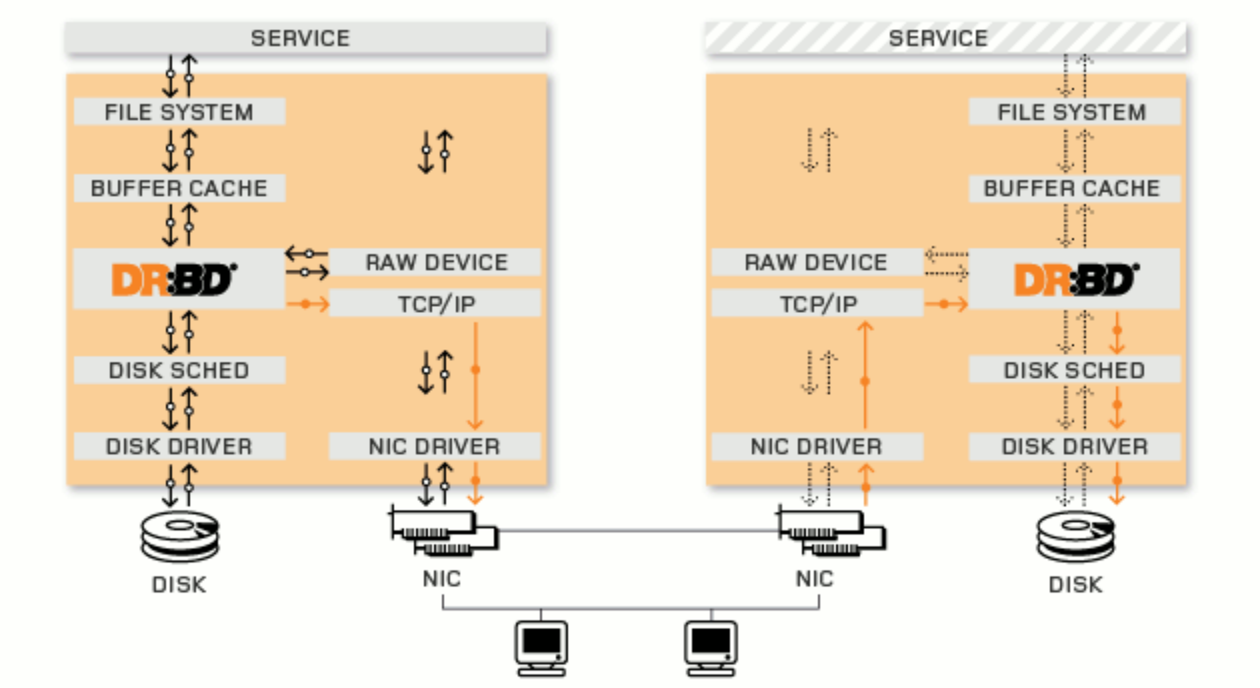
Introduction

- Motivation
    - Build a free, open version of Amazon EBS

- Background - Amazon EBS
    - Virtual, mountable block device for EC2 instances
    - Features
        - Replication
        - Snapshots

What do we know about EBS?

- Likely based on DRBD



Disadvantages of DRBD

- All the logic and hard stuff is in the kernel driver
- Nothing equivalent to dynamically growing disk images
- More than two replicas => stacked DRBD

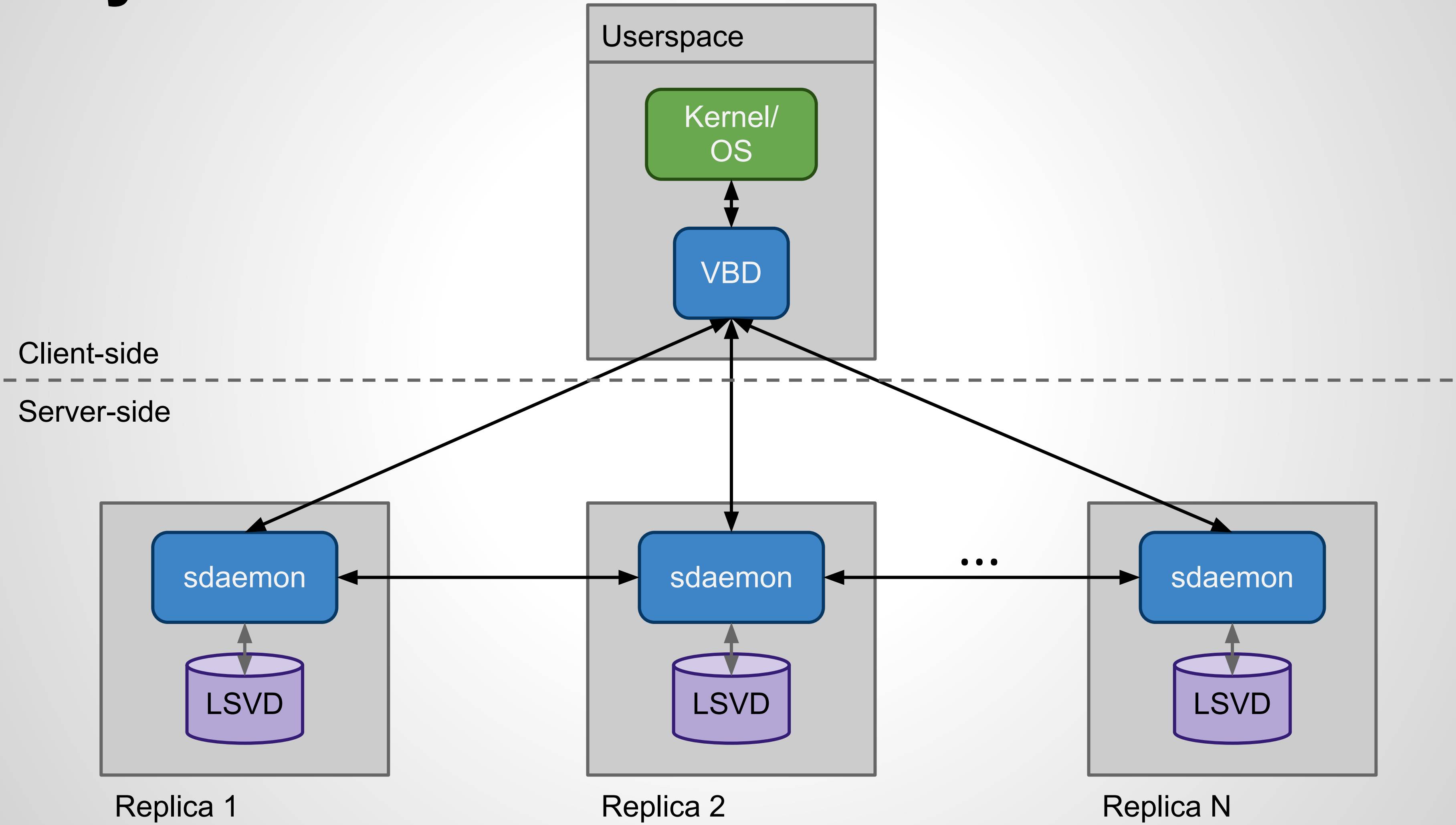
FreEBS

- Move as much as possible into userspace
- Use log-structured disk image file format as backing store
- Still provide:
  - Availability
  - Durability
  - Snapshots
  - Active, deterministic, virtualized enterprise reliability

Outline

- System Architecture
- Implementation Details
- Methodology
- Results
- Conclusion

System Architecture

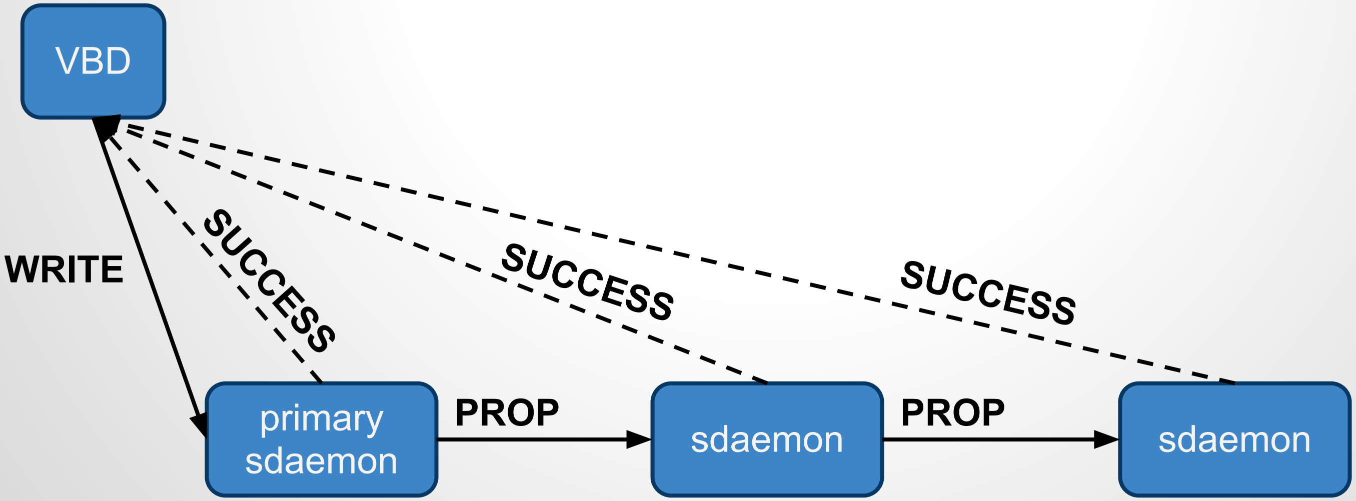


Outline

○ System Architecture
- Implementation Details
- Methodology
- Results
- Conclusion

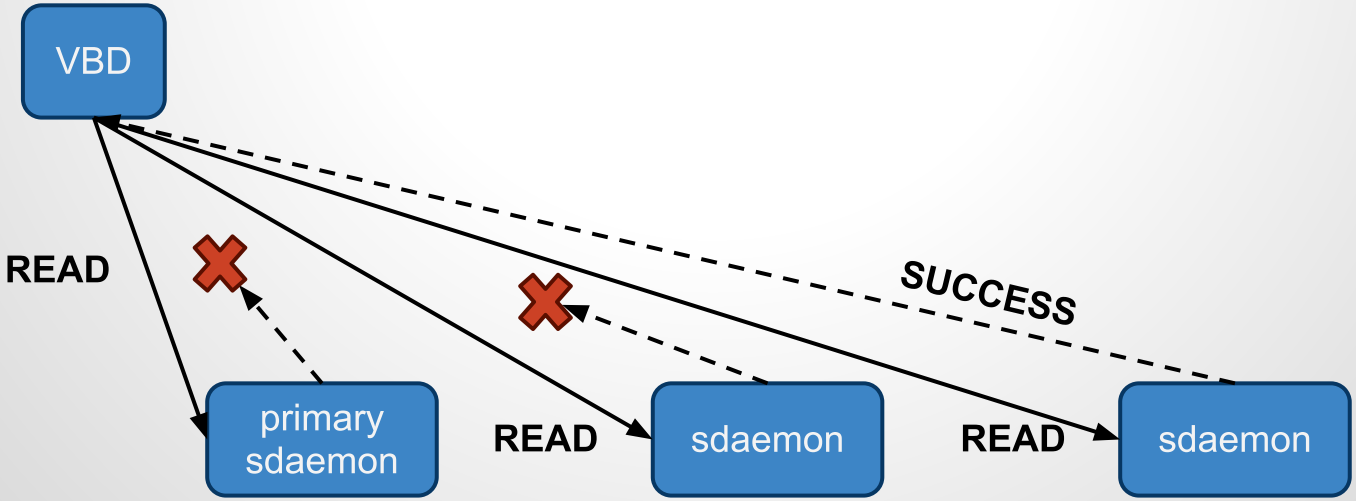
Write Operation

- Chained Replication
  - VBD issues WRITE request to primary
  - Primary sends PROP request to next replica, etc.
  - Replicas send response to VBD
  - Success if quorum reached, else fail



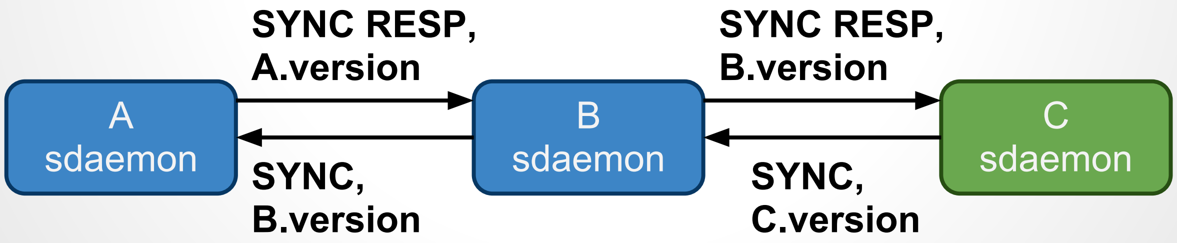
Read Operation

- Procedure:
  - VBD sends READ request to primary
  - If replica responds, serve client
  - Else, send read to next replica w/ most recent version
  - If none, then fail



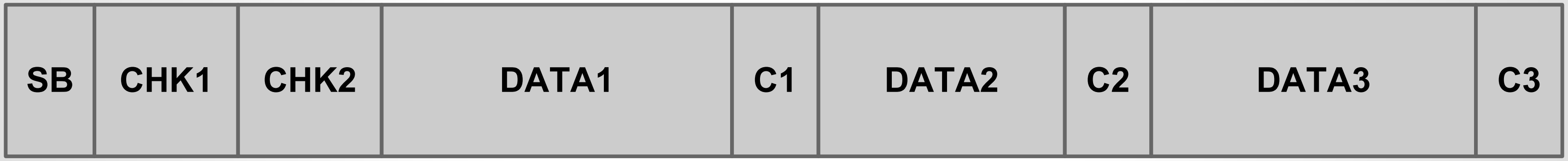
Synchronization

- Procedure
  - Send SYNC request to previous replica + version
  - Previous replica sends back writes since version



LSVD implementation

- Dynamically growing file format
- Versioning
- Data integrity



SB | CHK1 | CHK2 | DATA1 | C1 | DATA2 | C2 | DATA3 | C3

- Sector to offset map - Checkpointing (impl.)
- Cleanup (impl.)
- Snapshots (not impl.)

Outline

○ System Architecture
○ Implementation Details
● Methodology
● Results
● Conclusion

Methodology

- Setup
  - Driver on VM on mumble
    - VirtualBox
    - 2-core, 2.66GHz
  - Replicas on mumble machines
    - 4-core, 2.66GHz
    - 1 gbps network

Benchmarks

- Microbenchmarks
    - dd
- Benchmarks
    - fio - ioserver

Outline

○ System Architecture
○ Implementation Details
○ Methodology
● Results
● Conclusion

dd if=/dev/zero ... conv=fsync

- Local: ~60 MB/s
- FreeEBS: ~30 MB/s (w/ 2 replicas)

fio - iometer

- Mixed random reads (80%) and writes
- Non-buffered IO
- iodepth = 64
- 400 MB file

fio - iometer

		Avg throughput	Avg latency
FreEBS (2 replicas)	read	2423 KB/s	64 ms
	write	613 KB/s	68 ms
Local	read	1023 KB/s	301 ms
	write	269 KB/s	296 ms

Status

- One or two replicas (more coming)
- Checkpointing
- Segment cleaner

Conclusion

- Anything meaningful we can conclude about the data we got? Yes.
- Were we able to meet our goals? Totally.

Questions?