Erasure Code Offload for Distributed Software Defined Storage

Dror Goldenberg
VP Software Architecture
Mellanox Technologies
Software Defined Storage – Why?

**Scale out**
- More capacity
- More performance

**Cheaper**
- Resource utilization
- Software only

**Flexible**
- Multiple architectures
- Upgradeable

Flash Memory Summit 2017
Santa Clara, CA
Software Defined Storage: Scaling and Performing

- Network BW: Efficient Scale Out
- Low latency: Efficient Storage Disaggregation
- Offloading: Efficiency, Lower TCO

**NVMe Drives Speed (MB/s)**

- Sustained Bandwidth (MB/s)
- 1 NVMe: 2000 MB/s
- 2 NVMe: 4000 MB/s
- 4 NVMe: 6000 MB/s

**Storage Access Latency (ns)**

- Latency (ns)
- HDD: 100,000,000 ns
- SAS SSD: 10,000,000 ns
- NVMe: 1,000,000 ns
- NVDIMM: 100,000 ns

Flash Memory Summit 2017
Santa Clara, CA
To make storage cheaper we use lots more network!

How do we make Azure Storage scale? RoCE (RDMA over Converged Ethernet) enabled at 40GbE for Windows Azure Storage, achieving massive COGS savings.”
Ensuring Data Availability

Flash Memory Summit 2017
Santa Clara, CA

https://www.nextofwindows.com/raid-terms-explained-in-water-cooler
Ensuring Data Availability

Data Replication

Erasure Coding

D0
D0
D0
D0
D1
D1
D1
D1
D2
D2
D2
D2
D3
D3
D3
D3
D0
P0
D1
P1
D2
P2
D3
P3
Ensuring Data Availability

Data Replication

Erasure Coding
Ensuring Data Availability - Summary

**Data Replication**
- Capacity: 3x data (typical)
- Resilient to 2 failures

**Erasure Coding**
- Capacity: 1.4x data (typical)
- Better failure resilience
  - e.g. 10+4: 1.4x capacity, 4 failures
- CPU & network hungry
- Longer & data intensive rebuild
Erasure Coding Calculation

Calculating parity – Encoding
Matrix multiplication in GF(8):

\[ B \times [D] = [S] \]

Failure recovery (rebuild) – Decoding
Matrix multiplication in GF(8):

\[ [B']^{-1} \times [S] = [D] \]
Offload APIs - 101

- **Erasure Coding Onload**
  - Computation all done on CPU (CPU at 100%)
  - Cache/TLB pollution
  - Example: ISA-L

- **Erasure Coding Offload**
  - Computation all done in accelerator (CPU at 0%)
  - Cache/TLB unaffected
  - Example: ec_offload APIs

Efficient data movement - critical element to enable distributed scale out erasure coding
RDMA - asynchronous offload networking API
Encoding Performance (Single Core) – x86 ISA-L vs Offload
Encoding Performance (Single Core) – ARM vs Offload

K=10, M=4 Encoding Performance - ARM

Flash Memory Summit 2017
Santa Clara, CA
Network Traffic – Rebuild (Erasure Coding)

- Example - Time to rebuild (10+4)
  - Net networking time to move data
  - 20TB system @40GE 14.4hrs
  - 200TB system @40GE 144.4hrs

- Similar flows for scrubbing

Data recovery for erasure coding can be 10x more network intensive
Efficient data movement (RDMA) and efficient network are critical elements
Summary

- Cloud infrastructure requires efficient and scalable storage
- Software Defined Storage drives scale out storage
  - Performance, capacity, flexibility, TCO
- Erasure Codes enable data availability at lower capacity
  - Tradeoff: CPU & Network intensive
- Erasure Codes offload offers
  - Better performance (per I/O, efficient rebuild)
  - Lower cost
- Network efficiency (bandwidth, latency, offload) – an important enabler
- Library available today
  - Integration to scale out storage systems underway (Reed Solomon, LRC)
  - Can be used for local (host based) erasure coding, e.g. RAID5 codes
Thank You!