ZFS and multi-tiered storage

Boris Protopopov
Nexenta Systems Inc.
at OpenZFS Developers Summit
11.2013
ZFS and multi-tiered storage

• Why multiple storage tiers
• Multi-tiered zpool
  • Data path and storage management
  • Challenges and opportunities
• First steps
• Future work
• Discussion
Why multiple storage tiers

- Meet the needs of varying/mixed workloads
- Allow for efficient use of fast and small storage
- Achieve performance of fast and expensive storage at a fraction of the cost
- Leverage unique aspects of different storage technologies, device types, vendors/products
- Offer uncomplicated policy-based storage management workflows
Multi-tiered zpool: data path

• Differentiated placement of the payload
  – Data vs. metadata, metadata types
  – Dataset based payload differentiation

• Transient and permanent placement

• Interaction between application workloads and background management tasks, e.g. cross-tier payload movement

• Interaction between tiers based on similar storage technologies
Multi-tiered zpool: storage management

- Existing tiers: “normal” tier, logs, and cache
- Tier naming: reserved vs. user-defined names
- Payload placement and migration policies
- Tier-based redundancy and replication
- Tier-based scrubbing and repair
- Tier-based spare management
Multi-tiered zpool: challenges

• General case of payload migration
  – Mutable payload
  – Deduplication
  – Immutable payload

• Opportunities
  – Use cases without migration of immutable payload
  – Payload mobility (block pointer rewrite)
Multi-tiered zpool: first steps

• SSD-based dedicated metadata storage tier
• Differentiated placement of various metadata types in “normal” or “special” tiers
  – Pool level metadata
  – Dataset level metadata
• Managing interaction with cache tier (L2ARC)
  – “special-only”, dual, cache-only placement
• Tier-aware spare management
Multi-tiered zpool: first steps

• Generalized SSD-based tier
  – Use managed per dataset
• Can be used as log, meta device, or write cache
• The latter implies transient payload placement and subsequent migration
• Use cases without immutable payload are considered initially
Multi-tiered zpool: future work

• De-duplicated payload
  – Transient DDT class

• General case of cross-tier payload migration
  – Block pointer rewrite to move immutable payload
Discussion