Preparing Your Storage for Handling Even More Capacity - Again

Block Storage

By Yosef Shatsky
Storage Security & Software Architect
2019
Introduction

- Capacity has always been increasing – what’s new?
- What is the influence on the storage controller?
- Capacity is *one* of many factors
  - Workload
  - Performance
  - Feature set (disaster recovery, high availability, security, data reduction)
Worldwide Enterprise Storage Systems Capacity Shipped (PB)

© 2019 IBM Corporation
Workload

Capacity

Controllers

Workload
Workload

Capacity

Controllers

RAM

Workload

RAM

CPU

Bus

Interconnect
RAM Consumers in the Controller

- Physical capacity management
  - Free space mapping
  - Garbage collection metrics
- Read/write cache
- Virtual to physical lookup
- Deduplication database
RAM Consumers in the Controller

- Physical capacity management
  - Free space mapping
  - Garbage collection metrics
- Read/write cache
- Virtual to physical lookup
- Deduplication database
RAM Supply

“The DRAM shortage will continue through 3Q19”

“While the DRAM supply shortage continues, there is a limit on capacity for the entire market”
The Metadata Challenge

Handle more data with the same amount of RAM

Increase Storage-to-RAM ratio
## Swappable Structures

<table>
<thead>
<tr>
<th>Metadata</th>
<th>Swap Mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deduplication database</td>
<td>Content based</td>
</tr>
<tr>
<td>Virtual to physical lookup</td>
<td>Location based</td>
</tr>
<tr>
<td>Physical capacity management</td>
<td>Freed areas</td>
</tr>
</tbody>
</table>

- Typically, most of the data is cold
- Tailor each type of metadata to its workload  
  - Read / write  
  - Random / sequential
## Workload Optimized Metadata

<table>
<thead>
<tr>
<th>Structure</th>
<th>Read Workload</th>
<th>Write Workload</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deduplication database</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Read/write cache</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Virtual to physical lookup</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Physical capacity management</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

Optimize at a fine granularity (e.g. volume), not system-wide
Storage Class Memory / NVRAM

- Metadata persistency
- Read / write cache
  - Even without utilizing persistency
- Memory structure tiering
  - Similarly to data tiering with HDD and SDD
Metadata Structure Efficiency

- Use larger chunk sizes
  - Increases the amount of data per metadata entry
  - Fewer entries are required

- Use a sparse deduplication database
  - Reduces database size
  - Implement supplemental lookup algorithms
Proliferation of Storage Objects

- More: Volumes, Pools, Mirrors
- Caused by:
  - Increasing capacity
  - VVols
    - Adoption rate was slow
    - Support is increasing
  - Vvols 2 – replication
  - CDP - Continuous Data Protection
    - More snapshots
Capacity Cap on Failure Domains

- How much data would you place in a single system?
- Can we keep increasing the capacity of a single system?

- Clients prefer not to put all their eggs in the same basket
- Clients want to limit the size of a failure domain
- Concern involves failures that lead to:
  - Data loss
  - Offline time
- Failures may also include security breaches
Failure Domain = a System
Failure Domain < a System

If due to increased capacity density even small systems have too much capacity for a failure domain?
Failure Domain < a System

If due to increased capacity density even small systems have too much capacity for a failure domain?
Failure Model

- Single failure → redundant components
- System failure → DR site
  - DR sites are expensive
  - May not be immediate failover
- Software failure domain
  - Same error in all nodes
  - Triggered by state
  - Cascading
Domains by Capacity Growth

Number of Failure Domains vs. Capacity per Failure Domain

Capacity Growth
Conclusions

Storage to RAM ratio
- Swappable memory structures
- Workload optimized structures
- Sparse deduplication database

Object Proliferation
- Support
- Management

Failure domain
- Many physical systems
- Software/logical failure domains