Storage Acceleration, Driven by Autonomic Software

Phillip Clark / JJ Kane, Sr. Software Engineer
The Storage Dilemma

Increasing Performance Gap between Servers and Storage

- Increasing server performance
- Traditional disk performance

Friday, August 27, 2010
The Storage Dilemma

*Increasing Performance Gap between Servers and Storage*

- **2000**
  - Increasing server performance
  - Traditional disk performance

- **2010**
  - 2x performance
  - 25x performance gap

Servers / CPUs

Disk-based storage systems
The Storage Dilemma

Increasing Performance Gap between Servers and Storage

- Increasing server performance
- Traditional disk performance

2000 - 2010

SSD Driven Performance

Performance Gap!
Tiered Storage Architecture
Current Limitations
Tiered Storage Architecture

Current Limitations

- Lack Knowledge to Size Storage Tiers
  - Can cause over-provisioning of costly SSDs
  - Unable to predict/show performance gains
  - No metrics to measure improvement
Tiered Storage Architecture

Current Limitations

- Lack Knowledge to Size Storage Tiers
  - Can cause over-provisioning of costly SSDs
  - Unable to predict/show performance gains
  - No metrics to measure improvement

- Need Scalable High Capacity/Density Arrays
  - Not bandwidth matched to scale capacity
  - Does not leverage HDD=Capacity, SSD=Access
  - Limited support for large scale, semi-random workloads
  - Cannot span from cacheable to pure SSD random access
Tiered Storage Architecture

Current Limitations

- Lack Knowledge to Size Storage Tiers
  - Can cause over-provisioning of costly SSDs
  - Unable to predict/show performance gains
  - No metrics to measure improvement

- Need Scalable High Capacity/Density Arrays
  - Not bandwidth matched to scale capacity
  - Does not leverage HDD=Capacity, SSD=Access
  - Limited support for large scale, semi-random workloads
  - Cannot span from cacheable to pure SSD random access

- Inefficient Management
  - Not adaptive to changing access patterns
  - Requires IT time and resources
  - Inability to scale effectively
New Data Center Storage
ApplicationSmart Self-Optimization

Storage Cache and HSM Limitations: Cache is limited in scale/scope, HSM is slowly activated

- Too Small – GB, need TB
  - Locality and set association limited
  - Limited memory scaling

Traditional Storage Arrays

- Bandwidth Mismatch w/ Cache & HDD
  - Missed Penalty, Too High
    - Over-provisioning of cache
    - Too much rack space for HDD
    - HSM migration with SSD is slow and steals bandwidth

Time Linear Cache

- Imprecise, Not Adaptive
  - RAM cost/GB high
  - Cache hit rate efficiency with LRU/MRU low

ApplicationSmart Provides Data Access Acceleration: Manages cacheable data in real-time

- If Data is Completely Random, Need Spindles or Random Access Storage
  - SSD
  - SAID

Profile Monitors Access Changes

Identifies Cacheability by Application and Computes Speed-up

Terascale
Petascale

Replication (TME)

Friday, August 27, 2010
## Autonomic Storage Tiering

<table>
<thead>
<tr>
<th>Customer Benefit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sizes SSD requirements</td>
<td>• Analyzes and recommends amount of SSD prior to purchase</td>
</tr>
<tr>
<td></td>
<td>• Only what is needed for applications, based on profile</td>
</tr>
<tr>
<td></td>
<td>• No over-buying or over-provisioning</td>
</tr>
<tr>
<td>No added management</td>
<td>• Enables autonomic data tiering, no policies to set</td>
</tr>
<tr>
<td></td>
<td>• Anticipates SSD needs based on data access patterns</td>
</tr>
<tr>
<td>Eliminates overhead</td>
<td>• Data is replicated but remains resident on HDDs</td>
</tr>
<tr>
<td></td>
<td>• Avoids migration to and from HDD and SSD</td>
</tr>
<tr>
<td></td>
<td>• No unnecessary IO, all tiering is opportunistic</td>
</tr>
</tbody>
</table>
# Autonomic Storage Tiering

**Hybrid VLUN spans SSDs and HDDs**

<table>
<thead>
<tr>
<th>Customer Benefit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sizes SSD requirements</td>
<td>• Analyzes and recommends amount of SSD <em>prior</em> to purchase</td>
</tr>
<tr>
<td></td>
<td>• Only what is needed for applications, based on profile</td>
</tr>
<tr>
<td></td>
<td>• No over-buying or over-provisioning</td>
</tr>
<tr>
<td>No added management</td>
<td>• Enables autonomic data tiering, no policies to set</td>
</tr>
<tr>
<td></td>
<td>• Anticipates SSD needs based on data access patterns</td>
</tr>
<tr>
<td>Eliminates overhead</td>
<td>• Data is replicated but remains resident on HDDs</td>
</tr>
<tr>
<td></td>
<td>• Avoids migration to and from HDD and SSD</td>
</tr>
<tr>
<td></td>
<td>• No unnecessary IO, all tiering is opportunistic</td>
</tr>
</tbody>
</table>
The Bottom Line - Hybrid Storage Delivers the Flexibility to Solve Problems

Fundamental Storage Customer Requirements

<table>
<thead>
<tr>
<th>Performance</th>
<th>Capacity Scalability</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM Scaling</td>
<td>SSD Scaling</td>
<td>HDD Scaling</td>
</tr>
</tbody>
</table>

Performance vs. Capacity

RAM Scaling: $$$$$$

SSD Scaling: $$$

HDD Scaling: $
The Bottom Line - Hybrid Storage Delivers the Flexibility to Solve Problems

Fundamental Storage Customer Requirements

<table>
<thead>
<tr>
<th>Performance</th>
<th>Capacity Scalability</th>
<th>Cost</th>
</tr>
</thead>
</table>

- **Performance**
  - RAM Scaling $$$$$
  - SSD Scaling $$$
  - HDD Scaling $

- **Capacity Scalability**
  - RAM Scaling $$$$$
  - SSD Scaling $$$
  - HDD Scaling $
The Bottom Line - Hybrid Storage Delivers the Flexibility to Solve Problems

Fundamental Storage Customer Requirements

<table>
<thead>
<tr>
<th>Performance</th>
<th>Capacity Scalability</th>
<th>Cost</th>
</tr>
</thead>
</table>

RAM Scaling $$$$$

SSD Scaling $$$

SSD + HDD Scaling $$

HDD Scaling $

Add HDD Back-end

Add SSD TME / IA
Add RAM TME / EA
Performance Increase with SSDs

IO Request Size

512b 4K 32K 64K 128K 512K 1MB

IOPs (IO/sec)

HDD IOPs

SSD IOPs

Bandwidth

20 K

400 K

2 GB/s
Performance Increase with SSDs

- HDD IOPs
- SSD IOPs
- Bandwidth

IO Request Size:
- 512b
- 4K
- 32K
- 64K
- 128K
- 512K
- 1MB

IOPs (IO/sec):
- 20K
- 400K

Bandwidth (MB/sec):
- 2 GB/s

Friday, August 27, 2010
Multi-Tiered Management Software

ApplicationSmart™

Access Profiler
- Adaptive histogram, highly compressed, scales to PB
- Drives TME to accelerate IO for high access content

TME (Tiered Management Engine)
- Dynamic block replication with access pattern changes
- Optimal FBR (or plug-in heuristic) set replacement
- Mapped to LUNs or pools of LUNs

Ingest Accelerator
- Tuned for RAID access (FIFO, back-end IO reforming)
- Lower latency, higher throughput with log-structured FIFO

Egress Accelerator
- Detector for sequential/random initiator streams
- Read-ahead cache with auto enable/disable

SLM (SSD LUN Manager)
- Full AVS VLUN creation and management
- SSD storage pool, data lifetime protection options

Friday, August 27, 2010
Data Access Profiler

Provides real-time application storage access patterns

Histogram Analysis
- Identifies access hot-spots
- Notes when access changes are statistically significant
- Mapping integrates with virtualization engine

Histogram Groupings
- Drives TME IO acceleration
- Replicates blocks when statistically significant
- Provides continuous opportunistic updates
- Uses access visualization
Works w/ Wide Spectrum of Workloads

Sequential

Hot-Spots

Egress IO read-ahead

Ingest IO reforming

Fully Predictable
(Solid State FIFOs)

Semi-Predictable
(Scalable Hybrid Flash/Disk)

Random

Non-Cacheable
(Solved by Spindle Density or Random Access Storage Devices)
Tiered Management Engine (TME)

- Uses output from Access Profiler to drive TME
- Dynamic block replication
- As patterns change, new blocks are replicated
- Overwrites less active data
Ingest Acceleration

- Log Structured IO Reforming
  - Handles Sub-Optimal Writes
  - Optimizes for RAID Backend

**Diagram:****

- Tier 0
- Tier 1
- RAID0 SATA ROC
- AVE
- 2115K Single IO
- 16x128K Threaded IOs
- Single IO Completion
- 33rd Single 67K IO
- 16x128K Threaded IOs
- Single IO Completion

---

**Friday, August 27, 2010**
# Performance Tiering Checklist

<table>
<thead>
<tr>
<th>Performance Optimization</th>
<th>Key Benchmarks:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Integrates high velocity storage tiers (<em>Tier 0, Tier 1</em>)</td>
</tr>
<tr>
<td></td>
<td>• Block level movement for increased granularity</td>
</tr>
<tr>
<td></td>
<td>• Works across multiple application workloads</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dynamic Management</th>
<th>Key Benchmarks:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Understands access patterns and changes</td>
</tr>
<tr>
<td></td>
<td>• Moves data in real-time</td>
</tr>
<tr>
<td></td>
<td>• Supports multiple VLUN configuration options</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cost Efficiency</th>
<th>Key Benchmarks:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Integrates efficient storage architectures (<em>Tier, 0, Tier 1</em>)</td>
</tr>
<tr>
<td></td>
<td>• Recommends SSDs only when needed</td>
</tr>
<tr>
<td></td>
<td>• Fully autonomic, minimizes human intervention</td>
</tr>
</tbody>
</table>
Thank You!
Questions?