Why Your Next Server Will Have a Solid-State Cache

Ted Sanford
CEO FlashSoft
Why Caching in the Server?

- **Performance**
  - Active data on flash in the server
  - Near the speed of flash as primary storage
  - 3x to 10x application acceleration

- **Economics**
  - ROI of caching hot data only – spend far less
  - Server consolidation + storage efficiency
  - Leverage existing storage investments
Flash in the server vs. storage

Tier -1 ("tier minus one")

10 – 20 µs

High-Performance IO

Tier 0 (SSD)

Tier 1 (SAS)

Tier 2 (SATA)
Design Goals: Caching for the Server

- Optimized for flash
- Software-only solution
- Any type of SSD: PCIe, SAS, SATA
- Turns SSD into persistent read-write cache
  - Read-only caching also supported
- Transparent to existing SW layers
- Minimal server resource utilization
  - Memory, CPU
- Cross-platform caching engine
  - Windows, Linux, ESX
Enabling the SSD Cache

Log Structured Cache
- Circular buffer
- Write variable size blocks
- Minimize amount of metadata
- Reduce data fragmentation

Multi-level Metadata
- Minimal resource utilization
- 150MB memory for 1TB cache
- Low CPU utilization – 3% - 5%
- Instant recovery after crash

Software Flexibility
- Tuning for specific SSDs
- Tuning for specific apps
- Read-write-flush optimization
Design Goals: Caching for the Cluster

- Enable flash to accelerate applications in clusters
  - Read-write and read only cache
- Full High Availability support
  - Server-to-Server Horizontal Replication
    - Leverage existing LAN
  - Server-to-Storage Vertical Replication
    - Integration with underlying storage arrays
    - Tier -1 to Tier 0 API for integration with storage arrays
- Leverage existing cluster services
  - MSCS, VCS
Horizontal Replication

Server A

VM
Hypervisor
SSD (Tier -1)

Caching SW

Write transaction : SSD & Replication

Server B

VM
Hypervisor
SSD (Tier -1)

Caching SW

Storage Network

Shared Storage
Vertical Replication

Caching SW Tier-1

Storage Network

Tier 0

SSD Slice SSD Slice

Tier 1

Tier 2

SCSI Copy

Replication

SSD

Unused Write Cache Unused

Flush
Design Goals: Caching for Virtualization

- Support major virtualization platforms
- Install caching in the host
  - No agent in guest VMs required
  - Guest OS independent
- Low-latency SSD access
- Support all virtualization platform capabilities
  - Write-through caching mode
  - Write-back caching mode
Leverage Horizontal Replication

- High Availability
- Live Migration
- Snapshot & Clones
- Storage Motion, etc.
Technology Pyramid

- N-Node Cluster Support
- VM-specific Optimizations
- Integration with Cluster Services
- Flush Coordination
- Replication for HA
- Snapshots Capabilities
- Multi-level Metadata Management
- Log Structured Object Cache
- Caching for Virtualization
- Caching for Clusters
- Caching for Servers
Thank You

ted@flashsoft.com

http://www.flashsoft.com