Optimizing Software-Defined Storage for Flash Memory

Avraham Meir

Chief Scientist, Elastifile

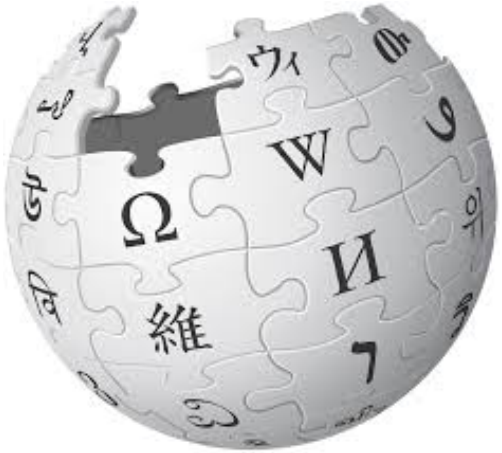
Agenda

- What is SDS?
- SDS Media Selection
- Flash-Native SDS
- Non-Flash SDS
- File system benefits

Wikipedia Says….

"…when implemented as software only in conjunction with commodity servers with internal disks, it may suggest software such as a virtual or global

file system…"



SDS Characteristics

In simple of words:

- Easy to use
- Easy to deploy
- Flexible and easy to management
- Scalable
- Hides hardware (HW Agnostic)

Media Options for SDS

Feature	HDD	FLASH	SCM (e.g. 3D XPoint)
Cost	x1	x5	~x20
Performance	x1	x1000	X10,000
Latency	x10,000	x100	x1
Availability	high	high	limited

- Optimal usage in SDS
  - FLASH – Primary storage for most applications
  - HDD – For cold data
  - SCM – For caching or front end tier (if available)

Flash-Native SDS

Key Elements:

- **Performance optimization** – different from HDD-centric approach
- **Cost optimization** – as flash is relatively expensive, cost reduction techniques should be used
- **Flash health & life cycle optimization**

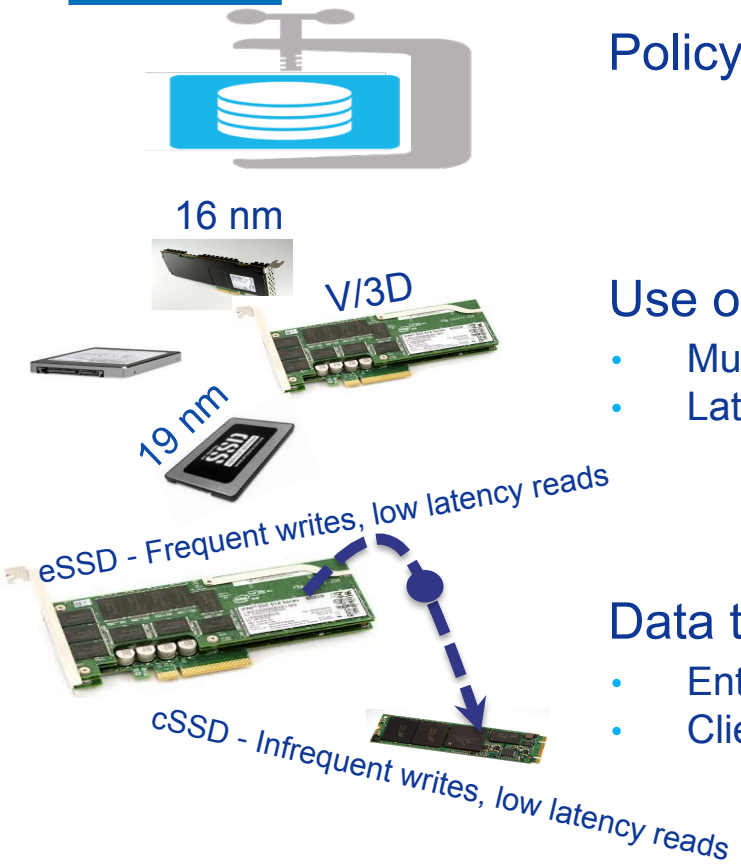
Performance Optimization

- **Prefer read over write**
  - HDD management - data segments re-written to enable efficient retrieval
  - SSD management – data is not re-written as write is expensive
- **Read caching**
  - HDD management - must have caching (huge media latency)
  - SSD management - does not require caching

Performance Optimization

- **De-allocate (TRIM) usage**
  - De-allocate significantly improves performance and SSD life-expectancy
  - When using de-allocate, I/O cost overhead should be considered
- **Erasure code management**
  - I/O amplification – classical implementations lead to significant I/O amplification
  - Reduced I/O amplification methodologies leads to improved performance

Cost Reduction



Policy-based compression and de-duplication

Use of Heterogeneous SSD hardware

- Multiple Vendors → lower cost
- Latest technology → better $/GB

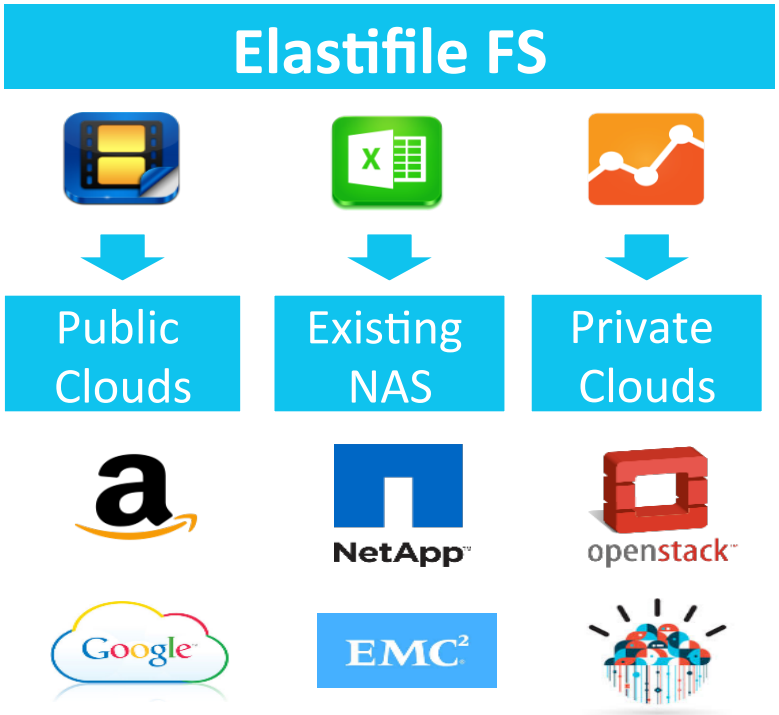
Data tiering between SSD without performance degradation[2]

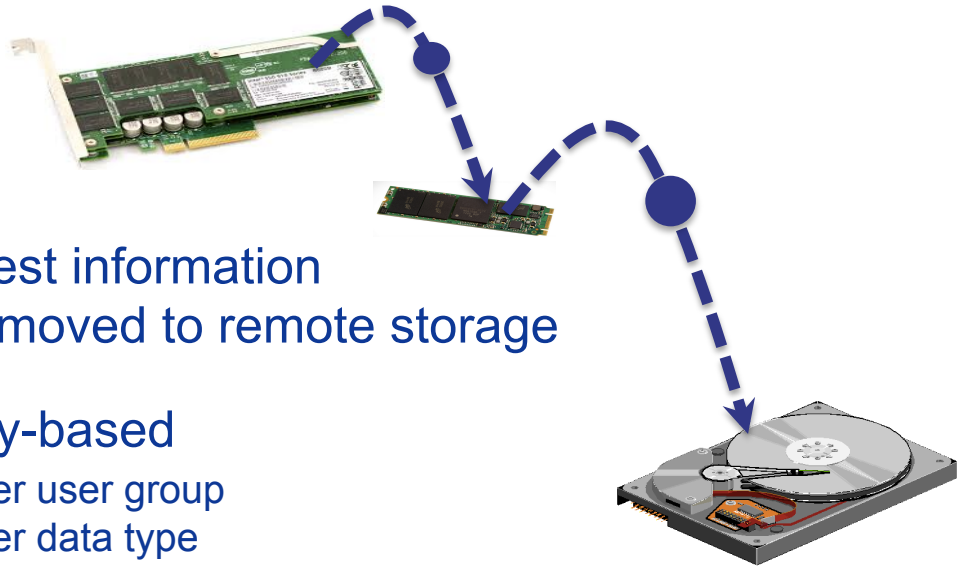
- Enterprise SSD – low latency reads, frequent writes
- Client SSD – 1/3 of the cost with low latency reads, infrequent writes

Comments:

1. System wide endurance leveling required (between SSDs)
2. Reliability management of each tier is required

Cost Reduction Through ILM[1]





Coldest information
gets moved to remote storage

Policy-based

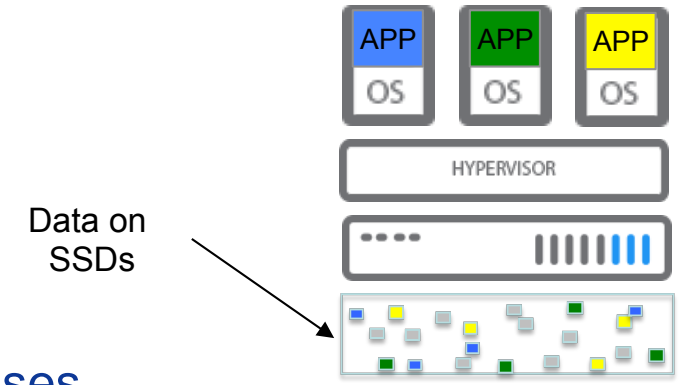
- Per user group
- Per data type

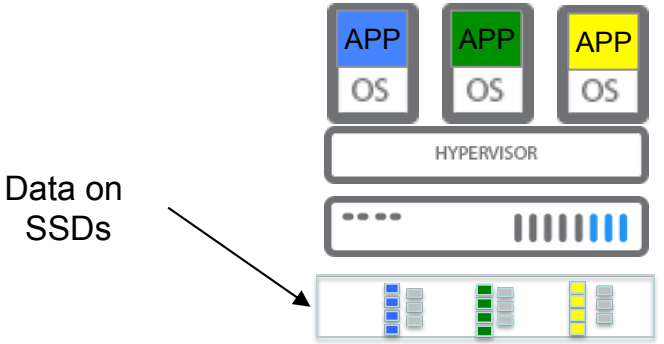
Comments:

1. ILM – Information Lifecycle Management

Life Cycle Extension Examples

- Locality
  - Virtualization causes all data to look random
  - A centralized file system can arrange the data[1]
- Hot/Cold separation
  - Mixing hot and cold (e.g. metadata & media) causes unnecessary garbage collection and reduces drive life
  - File system is capable separating the data by temperature[1]





1. Using the "streams" interface is an example of how a file-system can manage data on SSDs

Health Management

- **SSD Endurance**
  - Endurance monitoring - each SSD's endurance should be monitored separately
  - Endurance balancing - different SSD usage and endurance levels should be balanced

- **SSD Endurance rate**
  - SSD Endurance sensitivity - SSDs are sensitive to write rates[1]
  - Endurance rate should be managed

Comments:

1. Endurance rate doesn't appear in an SSD specification cycle

Health Management

- **Mixed enterprise & client drive usage**
  - Enterprise & client drives have different (retention) specifications
  - Mixed use requires different health management per type

"HW Agnostic" SDS

- SDS should provide a "HW agnostic" interface to the user

- Does this mean that the SDS is oblivious to HW? ***NO!!!!***
  - Must appropriately manage different SSD types
  - Must consider where to place data due to HW constraints
  - …. plus many any other considerations

Non-Native Flash SDS

Since legacy implementations were based on HDDs, there are many intermediate solutions:

- SSD as HDD replacement
  - Expensive
  - Limited life expectancy
- SSD as cache
  - Very low performance (when outside cache)
- SSD for metadata, HDD for data
  - Low performance

File System Benefits

***THE* General Interface** – File is a superset of other interfaces (object, block) and can support their use cases…

- File can deliver the simplicity of object
- A *good* file system can deliver the performance of block

…but only File delivers everything:

- **Data type awareness:**
  - File systems are aware of data types (mp3, mp4 etc.)
- **"Data type"-specific policies:**
  - File systems can assign different policies to different data types (different redundancy levels, different scheduling)
- **"Data type"-specific data placement:**
  - File systems can place different data types in the most suitable media (client, enterprise, external)
- **Simultaneous data sharing:**
  - File systems enable multiple users to access and process the same file simultaneously

Summary

- SDS can deliver many advantages relative to pre-existing solutions
- Flash should be used as the primary storage media
- The SDS must be optimized to effectively leverage flash
- A file system is the most efficient SDS platform

Thank You