SDC

STORAGE DEVELOPER CONFERENCE

SNIA ■ SANTA CLARA, 2016

An Enhanced I/O Model for Modern Storage Devices

Martin K. Petersen

ORACLE®

Modern Storage

- Flash / Shingled Magnetic Recording / Storage Arrays
- Existing I/O interfaces need to be enhanced to efficiently drive these devices
- Despite different storage media characteristics there is commonality in the required interface enhancements
- No need to throw away existing I/O model and rewrite legacy applications

OSD, ZAC, ZBC?

Cloud, archival, key/value stores, cat pictures

- Mostly reads
- Data is written once
- Reclaims are rare

Legacy applications

- Mostly random read / write, 70% / 30% mix
- Overwrites frequent and inevitable
- Random writes are not going away!

Media-Specific Command Sets

- Specialized vs. generalized
- Abstraction vs. implementation
  - Cylinders/Heads/Sectors
  - USB key to storage array
- A single I/O request is often going to multiple devices simultaneously

The Read/Write I/O Model

- Ingrained in POSIX/UNIX programming interfaces
- Thousands of applications depend on it, many impossible to change
- OS kernel provides an abstraction but requires cooperation from the storage device

Modern Storage Characteristics

Flash & Friends

- Directing requests into appropriate write channels based on the nature of the data
- Placing related data together on media to reduce write amplification
- Garbage collection

Shingled Magnetic Recording

- Placing often written data in conventional zones or flash
- Placing related data together to facilitate zone management
- De-staging, garbage collection

Storage Arrays

- Identifying performance critical vs. background tasks for tiering and QoS
- Distinguishing individual application I/O streams
- Scrubbing, reclaim, data migration

Common Needs for Modern Devices

1. Identifying the nature of the I/O
2. Identifying distinct, concurrent I/O streams
3. Identifying when to do background operations

1. I/O Classification

I/O Classification

I/O Class	Examples
Transaction	Filesystem or database journals, checkpoints
Metadata	Filesystem metadata
Paging	Swap
Real Time	Audio/video streaming
Data	Normal application I/O
Background	Backup, data migration, RAID resync, scrubbing

Application surveys from 2011 and 2014. 50 hints consolidated into 6 distinct I/O classes.

I/O Classification

I/O Class	Completion Urgency	Desired Future Access Latency	Predicted Future Access Freq.
Transaction	High	Low	High
Metadata	High	Low	Normal
Paging	High	Normal	Normal
Real Time	High	Normal	Low
Data	Normal	Normal	Normal
Background	Low	Normal/High*	Low

I/O Classification: T10 SBC

Table 59 — READ (10) command

Byte \ Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (28h)							
1	RDPROTECT			DPO	FUA	RARC	Obsolete	Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
•••								
5								(LSB)
6	Reserved					GROUP NUMBER		
7	(MSB)	TRANSFER LENGTH						
8								(LSB)
9	CONTROL							

I/O Classification: T10 SBC

M.3 Access Patterns LBMDs

M.3.1 Access Patterns LBMD format for SCSI

The Access Patterns LBMD format processed by SCSI device servers is shown in table M.4.

Table M.4 — Access Patterns LBMD format for SCSI

Bit / Byte	7	6	5	4	3	2	1	0
0	ACDLU	Reserved			LBMD TYPE (0h)			
1	OVERALL FREQUENCY		READ/WRITE FREQUENCY		WRITE SEQUENTIALITY		READ SEQUENTIALITY	
2	Reserved				SUBSEQUENT I/O		OSI PROXIMITY	
3	Reserved							

SDC 16

I/O Classification: NVM Express

Figure 172: Read – Command Dword 13

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:08</td><td>Reserved</td></tr>
<tr><td></td><td>Dataset Management (DSM): This field indicates attributes for the dataset that the LBA(s) being read are associated with.
<table>
<tr><th>Bits</th><th>Attribute</th><th>Definition</th></tr>
<tr><td>07</td><td>Incompressible</td><td>If set to ‘1’, then data is not compressible for the logical blocks indicated. If cleared to ‘0’, then no information on compression is provided.</td></tr>
<tr><td>06</td><td>Sequential Request</td><td>If set to ‘1’, then this command is part of a sequential read that includes multiple Read commands. If cleared to ‘0’, then no information on sequential access is provided.</td></tr>
<tr><td>05:04</td><td>Access Latency</td><td>
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>00b</td><td>None. No latency information provided.</td></tr>
<tr><td>01b</td><td>Idle. Longer latency acceptable.</td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

SDC 16

I/O Classification: NVM Express

07:00	03:00	Access Frequency	(see table below)

Value	Definition
0000b	No frequency information provided.
0001b	Typical number of reads and writes expected for this LBA range.
0010b	Infrequent writes and infrequent reads to the LBA range indicated.
0011b	Infrequent writes and frequent reads to the LBA range indicated.
0100b	Frequent writes and infrequent reads to the LBA range indicated.
0101b	Frequent writes and frequent reads to the LBA range indicated.
0110b	One time read. E.g. command is due to virus scan, backup, file copy, or archive.
0111b	Speculative read. The command is part of a prefetch operation.
1000b	The LBA range is going to be overwritten in the near future.
1001b – 1111b	Reserved

I/O Classification

- Per-I/O property. Static LBA labeling does not work
- Allows the device to distinguish between file data, metadata, transaction logs, etc.
- Communicates the *intent* of why the system is doing I/O
- Tied into posix_fadvise*()*, kernel flags, I/O priority

2. I/O Affinity

I/O Affinity

- Establishes affinity between data submitted in separate I/O requests
- Allows the device to distinguish between different files
- T10 SBC Streams, NVMe Directives used to set an appropriate affinity for every command
- Affinity ID is hashed based on partition/inode #

I/O Affinity

Table 136 — WRITE STREAM (16) command

Byte \ Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (9Ah)							
1	WRPROTECT			DPO	FUA	Reserved		
2	(MSB)	LOGICAL BLOCK ADDRESS						
•••								
9								(LSB)
10	(MSB)	STR_ID						
11								(LSB)
12	(MSB)	TRANSFER LENGTH						
13								(LSB)
14	Reserved		GROUP NUMBER					
15	CONTROL							

I/O Affinity

- 16-bit ID in SCSI and NVMe
- Hash collisions depend on workload
- Number of concurrently available write groups or zones much smaller than 64K
- Number of concurrently open files somewhere between 1K and 100K
- Number of files open for write between 100 and 10K

Combining I/O Classification & Affinity

- Establishes relationship between concurrently issued write requests to facilitate data placement
- Provides separation of distinct I/O streams for cache management and QoS
- Identifies circular logs, metadata, swap
- Identifies high priority (realtime), normal, or low priority (background) requests for prioritization and scheduling purposes

3. Background Operations

Background Operations

- All device types have a need to do background operations
- Device usually initiates when needed or idle
- OS has little insight into application behavior
- OS would like to provide a protocol-agnostic interface much like we have done for TRIM/ UNMAP

SDC 16

Background Operations: T10 SBC

Table 35 — BACKGROUND CONTROL command

Byte \ Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (9Eh)							
1	Reserved				SERVICE ACTION (15h)			
2	BO_CTL		Reserved					
3	BO_TIME							
4								
•••	Reserved							
14								
15	CONTROL							

Results

Results

- Classification via I/O Advice Hints and DSM
- Intelligent data placement via SBC Streams and NVMe LBA Affinity
- Cooperative scheduling of background tasks

Abstract protocol extensions that allow modern storage media to be driven more efficiently without departing from existing application I/O model.

Results

- Flash/SMR/Hybrid
  - Simulated 10 channel/zone target
  - Colocation rate
  - Journal/metadata separation
- Storage Arrays
  - Cache management
  - Backup tasks only use idle time, no business application performance impact

Future Work

- Storage arrays work really well
- Looking to engage with disk drive and flash vendors
- NVM Express Streams vs. LBA Affinity
- T10 SBC READ STREAM and implicit open/close semantics

Questions?