SNIA Tutorial 3
EVERYTHING YOU WANTED TO KNOW ABOUT STORAGE:
Part Teal —
Queues, Caches and Buffers

John Kim, Mellanox - @Tier1Storage
J Metz, Cisco - @drjmetz

2018 Flash Memory Summit
Welcome to SNIA Education Afternoon at Flash Memory Summit 2018
<table>
<thead>
<tr>
<th>Time</th>
<th>Tutorial</th>
</tr>
</thead>
</table>
| 1:00 pm – 1:50 pm | SNIA Tutorial 1  
A Case for Flash Storage  
Dejan Kocic, NetApp |
| 1:50 pm – 2:45 pm | SNIA Tutorial 2  
What if Programming and Networking Had a Storage Baby Pod?  
John Kim, Mellanox Technologies and J Metz, Cisco Systems |
| 2:45 pm – 3:00 pm | Break                                                                   |
| 3:00 pm – 3:50 pm | SNIA Tutorial 3  
Buffers, Queues, and Caches  
John Kim, Mellanox Technologies and J Metz, Cisco Systems |
| 4:00 pm – 5:00 pm | SNIA Tutorial 4  
Birds-of-a-Feather – Persistent Memory Futures  
Jeff Chang, SNIA Persistent Memory and NVDIMM SIG Co-Chair |
170 industry leading organizations

2,500 active contributing members

50,000 IT end users & storage pros worldwide
Join SNIA at These Upcoming Events

Storage Developer Conference 2018

9/24-9/27 Santa Clara, CA

SDC discount registration cards in FMS bags & at SNIA booth 820

Complimentary registration now open at snia.org/pm-summit
The material contained in this presentation is copyrighted by the SNIA unless otherwise noted.

Member companies and individual members may use this material in presentations and literature under the following conditions:
- Any slide or slides used must be reproduced in their entirety without modification
- The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.

This presentation is a project of the SNIA.

Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.

The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.
Special Thanks

Mark Rogov
Dell EMC

David Minturn
Intel

Rob Davis
Mellanox
Today’s Agenda

- Queuing
- Buffering
- Caching
- Flow Control
QUEUING
**Definitions- Queue Depth**

**IO Operation (aka “IOP”)**
Storage operation issued by a host (initiator) to a storage device/system (target)
*Example: Host issues a READ Operation of 100 blocks from a storage device*

**IO Queue**
A queue which holds one or more outstanding **IO Operations**

**IO Queue Depth**
Maximum number of outstanding **IO Operations** that the **IO Queue** can hold
Example: IO Queue with a IO Queue Depth of three
Larger Queue Depth allows IO Operations to be serviced in parallel or batched resulting in higher total IOPS and bandwidth.

Example Results:
@Queue Depth = 1;
  IOPS are 10K (1/100usec)
@Queue Depth = 3;
  IOPS are 25K (3/120usec)
IO Queue Depth Considerations (End to End Queue Depth)

- Host systems typically have multiple Autonomous Processes simultaneously issuing IO Operations (Application Queue Depth)
- Host O/S storage stacks have internal queues to accommodate oversubscribed Storage Device IO Queues (O/S Stack Queue Depth)

Queue Depth must be looked at End to End; App->O/S->Storage Device->Media
Modern Storage Devices use a multi IO queue model for efficiency, typically one IO Queue per host CPU

Effective Storage Device IO Queue Depth equals:
# of IO Queues * individual IOQ Depth
Memory buffers (termed Transfer Buffers) are used to exchange IO Operation data between the Host System and Storage Device.

- Transfer buffer resources are committed until the IO Operation completes.
  - Resources may be large; example 64K IO requires 64K of memory.

\[
\text{IO Queue Memory} = \text{IO Queue Depth} \times \text{size of (IOP Descriptor + Transfer Buffer)}
\]
CACHE
Cache in the US
Cache (aka Cache Memory)
/'kaSH/

- An auxiliary memory from which high-speed retrieval is possible
Buffer

Use once and throw it away + allows blocks re-arranging
Cache implies multiple use of blocks
Reads with Cache

Read Hit: First reason for Caches to exist
Reads with Cache

Read Miss

CACHE

REQUEST

REPLY

READ FROM DISK
Writes with Cache

Write-through: Write data to Cache and Disk, then confirm completion
 Writes with Cache

Write-around: Bypass cache
Writes with Cache

Write-back: Write data to Cache and confirm completion; write to disk later
Listing the contents of a directory

"Uncached ‘ls’" had a USB unmount just prior to the ‘ls’ command execution
Where Do Caches Exist?

- PHYSICAL STORAGE
- BACK-END CONNECT
- STORAGE CONTROLLER
- FRONT-END CONNECT
- CLIENTS / HOSTS

CACHE
FLOW CONTROL
What is Flow Control?

Flow control is a mechanism for temporarily stopping the transmission of data on computer network to avoid buffer overflows.
What is Flow Control?

No Flow Control
What is Flow Control?

No Flow Control

Flow Control
Buffers are Everywhere

All computer networking devices have some buffers to facilitate speed matching.
Buffers are Everywhere

But these buffers seem to never be big enough.
Buffers are Everywhere

This can lead to a Buffer Overflow resulting in Data Packet Drops forcing error recovery delays.
Buffer Overflows are Bad…
Buffer Overflows happen when more data is coming into a networking device then is going out.
IEEE 802.3x standard defines a flow control mechanism for Ethernet called the pause frame.
Flow Control Prevents Overflows

When the data in a buffer gets to a certain level the pause frame is sent causing the upstream device to stop sending data for a specified amount of time.
With credit based flow control the sending device knows how much buffer space the receiving device has eliminating buffer overflows.
Explicit Congestion Notification (ECN) slows down a explicit device’s data rate that is believed to be overflowing another devices buffer.
Explicit Congestion Notification

The data rate of the device slowed down then increases in increments over time based on preset parameters.

RFC 3168 - Explicit Congestion Notification (ECN)
Priority Flow Control
Priority Flow Control
Priority Flow Control (PFC) is similar to 802.3x Pause, except eight priority levels are added. When the data in any of the eight buffers gets to a certain level a pause is sent causing the upstream device to stop sending data only for that priority level for a specified amount of time.

802.1Qbb - Priority-based Flow Control
Overall Summary

- Queues Line Up Work Processes or Requests
- Buffers absorb traffic bursts and smooth out data flow
- Caches store data closer to the user to accelerate access
- Flow Control Modules the Rate of Data or Requests to prevent buffer overflow
Check out previously recorded webcasts:
- [http://sniaesfblog.org/everything-you-wanted-to-know-about-storage-but-were-too-proud-to-ask/](http://sniaesfblog.org/everything-you-wanted-to-know-about-storage-but-were-too-proud-to-ask/)

- **Teal** – Buffers, Queues and Caches
- **Rosé** - All things iSCSI
- **Chartreuse** – The Basics: Initiator, Target, Storage Controller, RAID, Volume Manager and more
- **Mauve** – Architecture: Channel vs. Bus, Control Plane vs. Data Plane, Fabric vs. Network
- **Sepia** – Getting from Here to There
- **Turquoise** – Where Does My Data Go?
- **Cyan** – Storage Management
- **Aqua** – Storage Controllers
Storage Performance Benchmarking:

1. Introduction and Fundamentals
2. Solution under Test
3. Block Components
4. File Components

Watch them all on-demand at:
http://www.snia.org/forums/esf/knowledge/webcasts-topics
Thank You!

Visit snia.org/education for more webcasts, videos, and educational materials