STORAGE DEVELOPER CONFERENCE
SDC 22
Fremont, CA
September 12-15, 2022
BY Developers FOR Developers
A SNIA Event

Direct Drive

Azure's Next-Generation Block Storage Architecture

Greg Kramer
Partner Software Architect
Microsoft Azure Storage

Agenda

- Introduction
- Architectural overview
- Notable design elements
- Questions

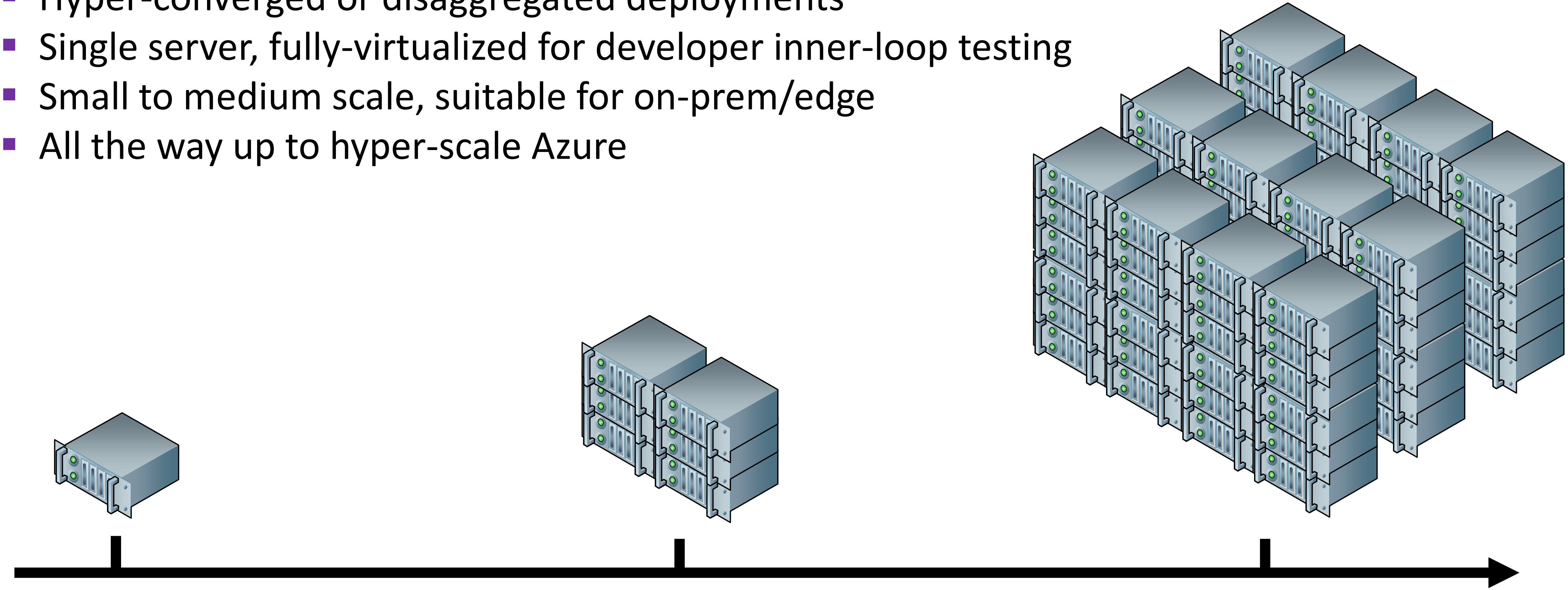
Introduction

What and why?

- Direct Drive is the internal code name for Azure’s next-generation block storage architecture
  - The foundation for a new family of disk offerings
    - Summer 2019: Azure Ultra Disk
    - Summer 2022: Azure Premium Disk v2 (in preview)
- Motivation
  - Microsoft has decades of storage experience
    - On-premises (Windows / Windows Server)
    - Public cloud (Azure).
  - New storage workloads and new technologies are constantly emerging
  - How would we use our experience to reimagine block storage for the next decade’s worth of growth?

A Spectrum of Deployment Options

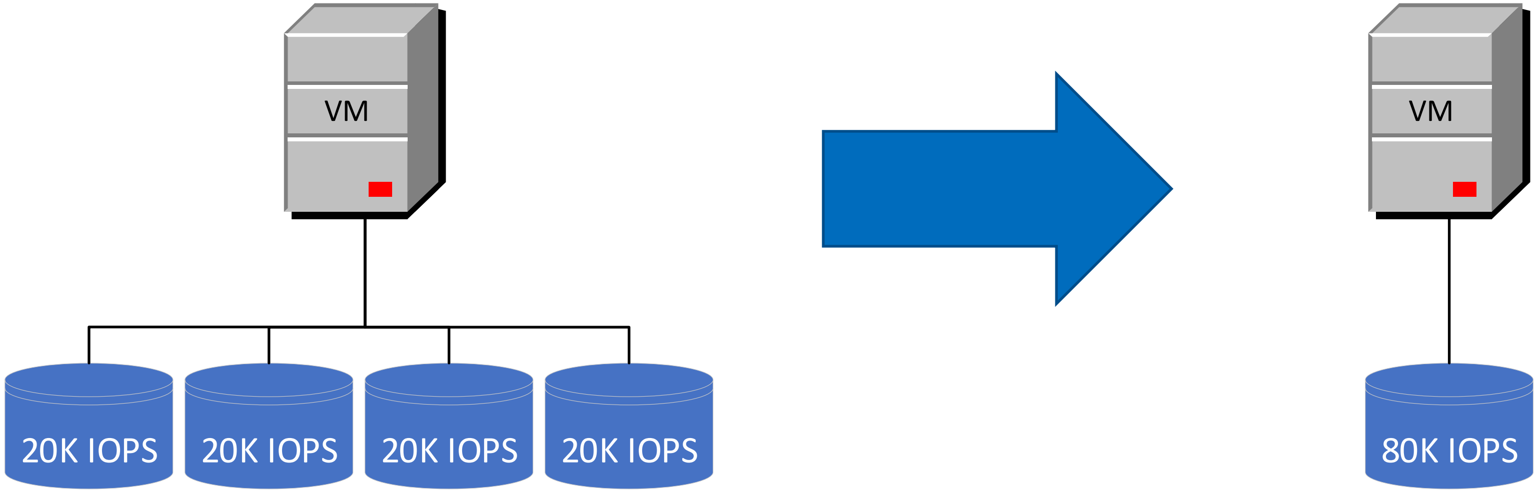
- Hyper-converged or disaggregated deployments
- Single server, fully-virtualized for developer inner-loop testing
- Small to medium scale, suitable for on-prem/edge
- All the way up to hyper-scale Azure



STORAGE DEVELOPER CONFERENCE
SDC 22

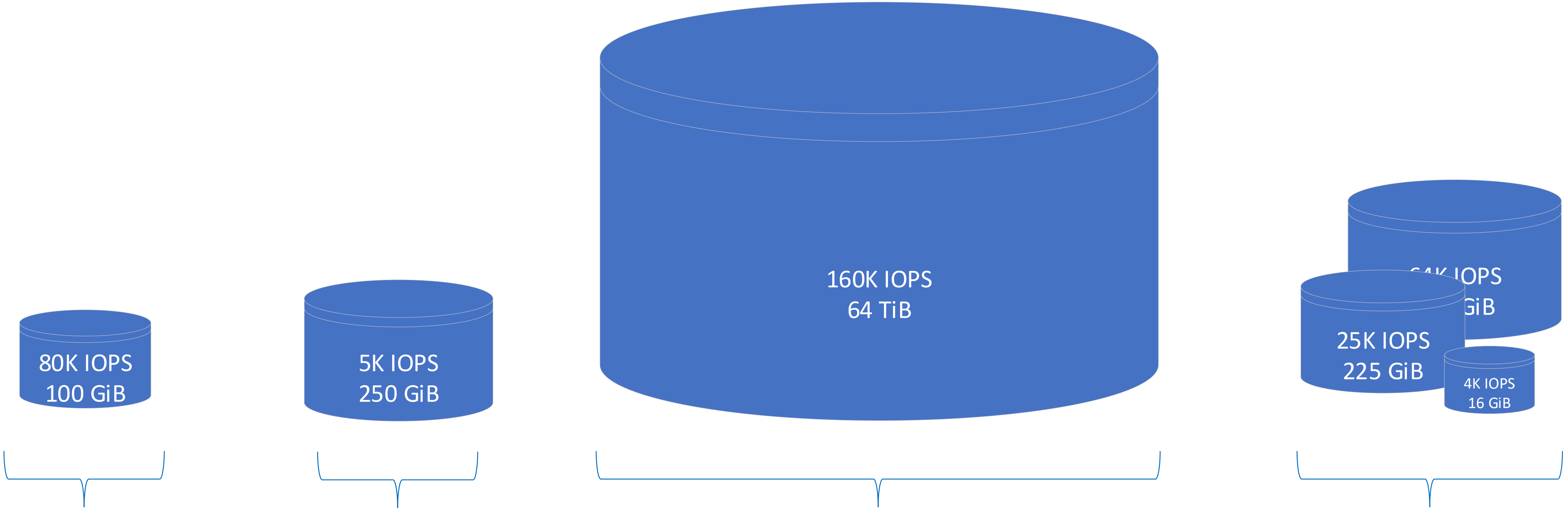
One disk for the job

Reduce the need to group multiple disks to achieve desired performance



One disk for the job…

Allow IOPS, throughput, and disk size to be independently configured.



Fast, small disk?
No problem!

Or a slower, larger disk?
Sure thing!

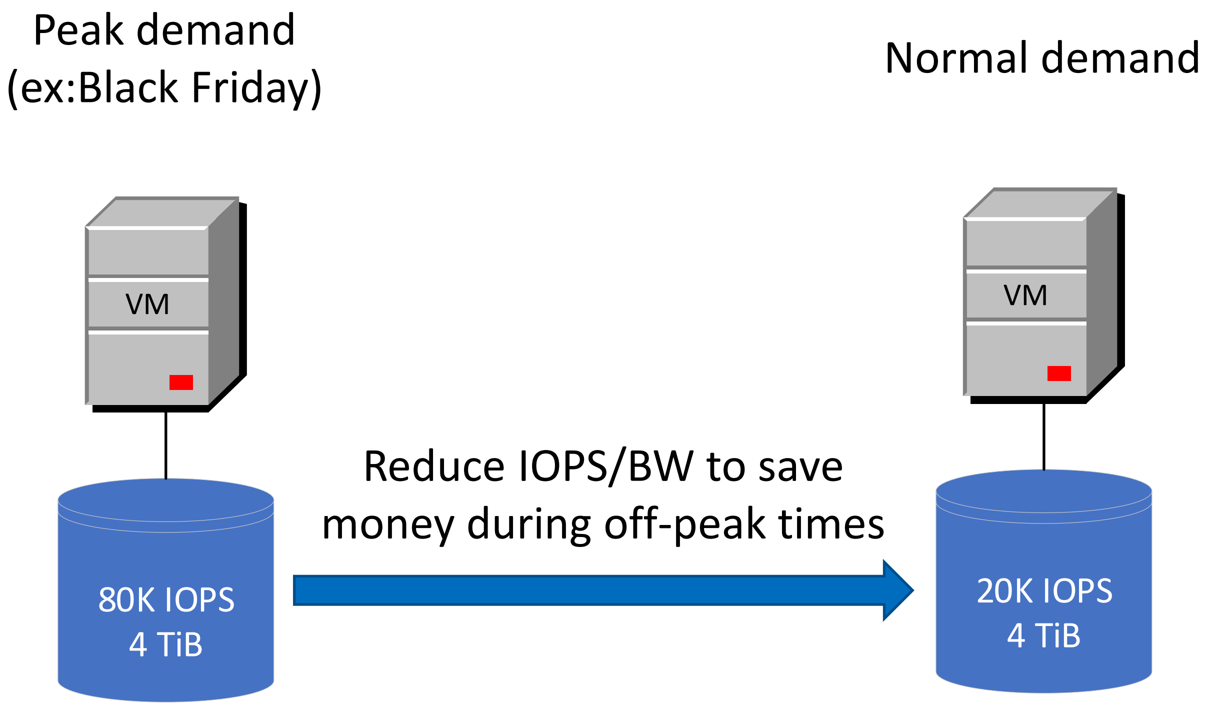
More of everything?
We've got you covered!

And everything in between…

STORAGE DEVELOPER CONFERENCE
SDC22

No need to provision for worst case

Allow performance to be changed dynamically to match workload needs



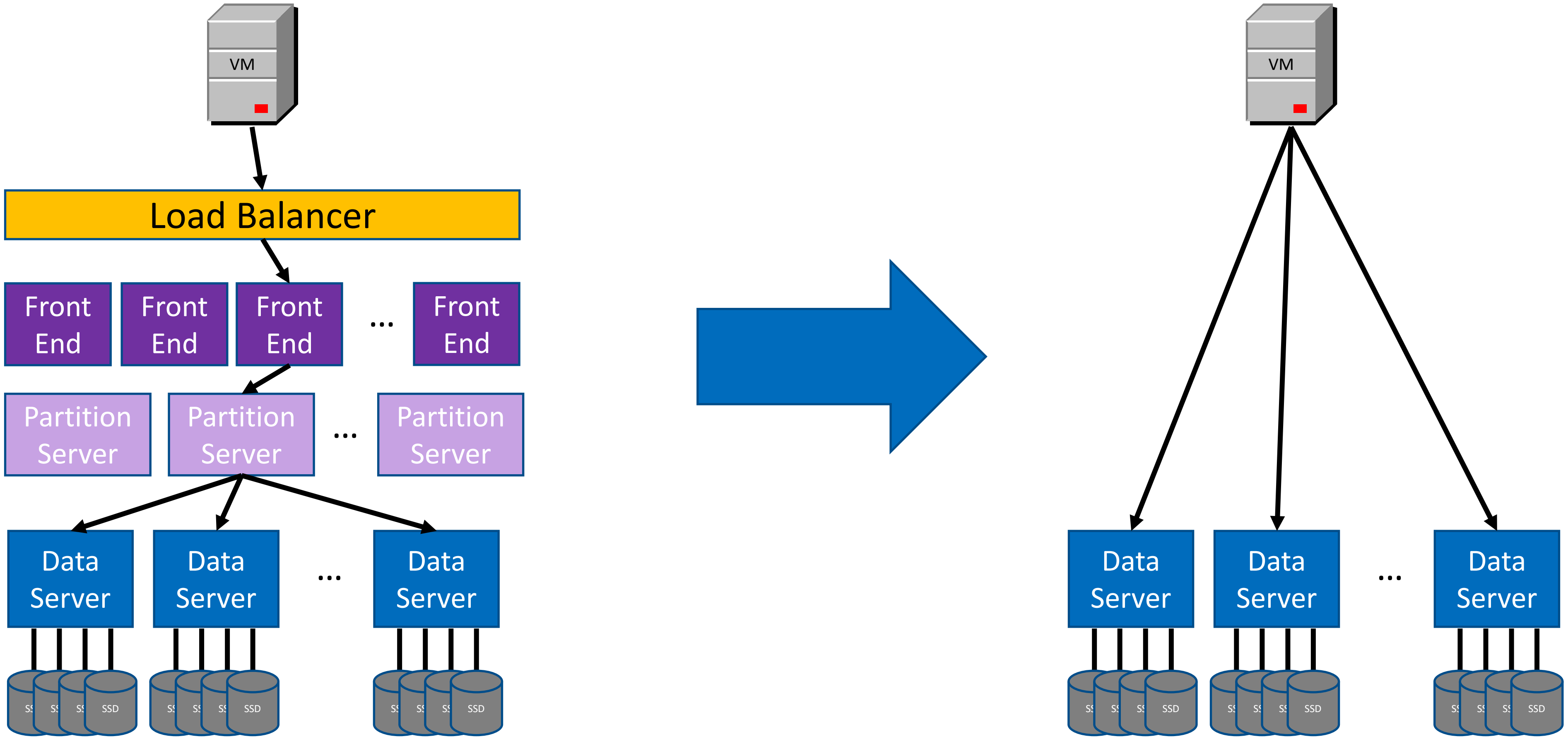
Positioned to take advantage of new technologies

- High speed, low latency networks
- New storage network protocols (more on this later)
- Storage class memory (SCM) and other innovations in storage media
- Hardware offloads (network, crypto, CRC, etc.)

Leverage these to provide consistently high-performance
while maintaining desired durability guarantees

Simplify the I/O path

Fewer layers means better, more consistent performance. Clients should be able to access data directly (the “Direct” in Direct Drive), avoiding load-balancers, front ends, partitioning layers, etc.



STORAGE DEVELOPER CONFERENCE SDC 22

Architectural Overview

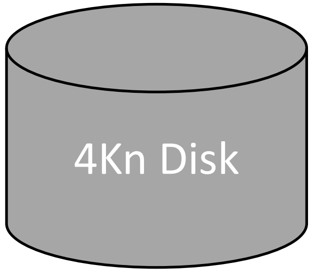
Before we get started...

- The Direct Drive architecture is very flexible
- Many options in terms of:
  - Form factors and deployment scale
  - Performance
  - Data durability guarantees
  - Feature set
- This talk is about the architecture, not specific products built on it.
  - Just because the architecture allows it, doesn't mean it's available in a product.
  - Always consult official docs for product capabilities, limitations, guarantees, etc.

Disks

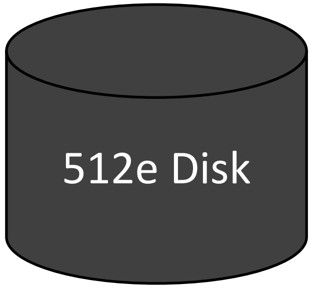
Two types of disks:

4 KiB native



4 KiB logical and 4 KiB physical sector

512 B emulated

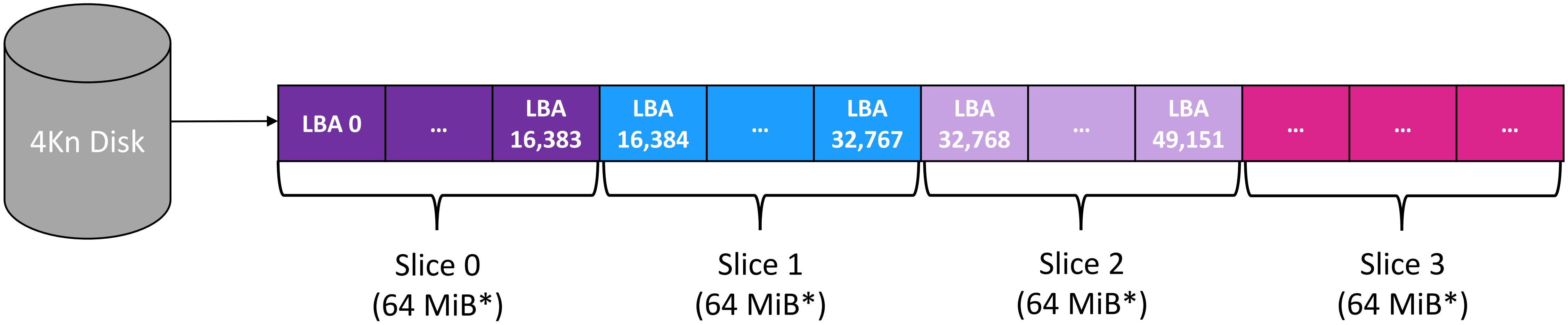


512 B logical and 4 KiB physical sector (best performance with 4K-aligned I/O to avoid read-modify-write penalty)

Core feature set includes:

- Shared disks (single-writer/multi-reader and multi-writer/multi-reader)
- Crash consistent, distributed snapshots
- Disk migration (move disk between storage clusters while mounted and taking I/O)

Disk Layout (slices)

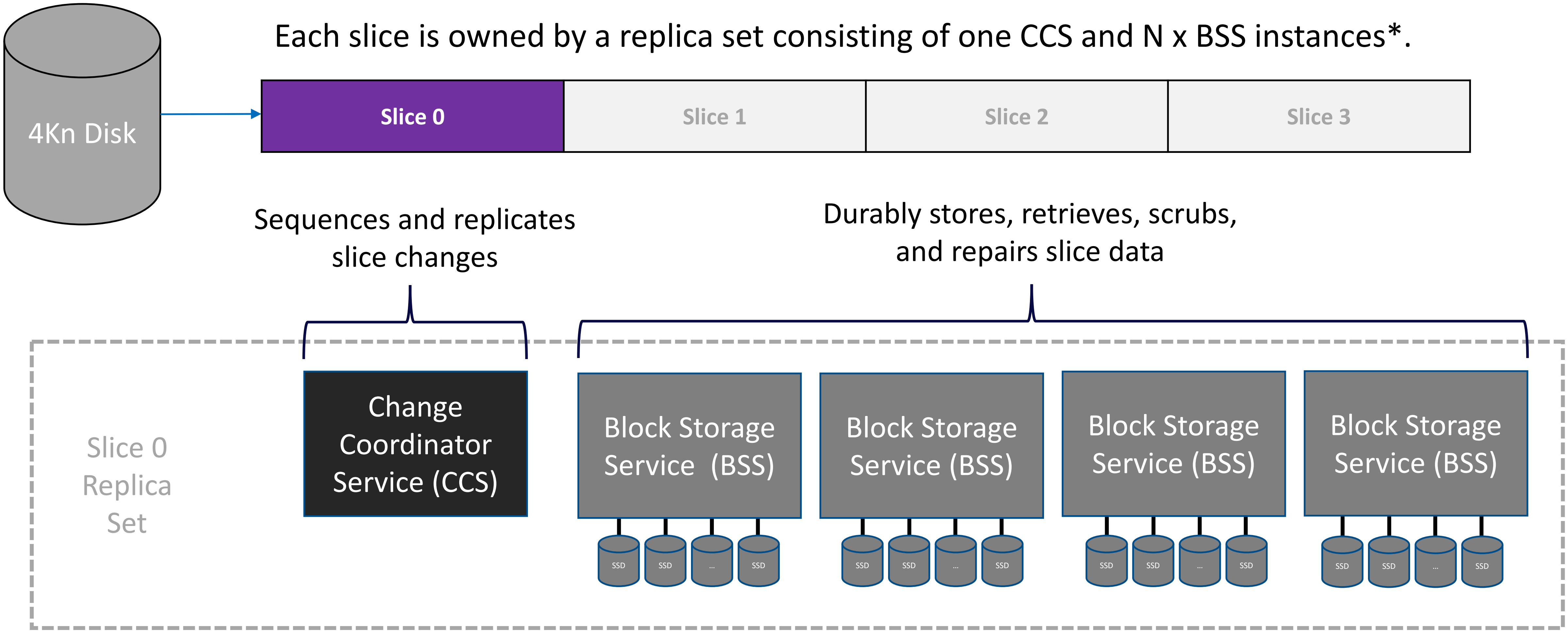


Disks are managed in fixed-size chunks called slices

* Slice size is a configurable property of a disk, 64 MiB is just an example.

Slice Replica Sets

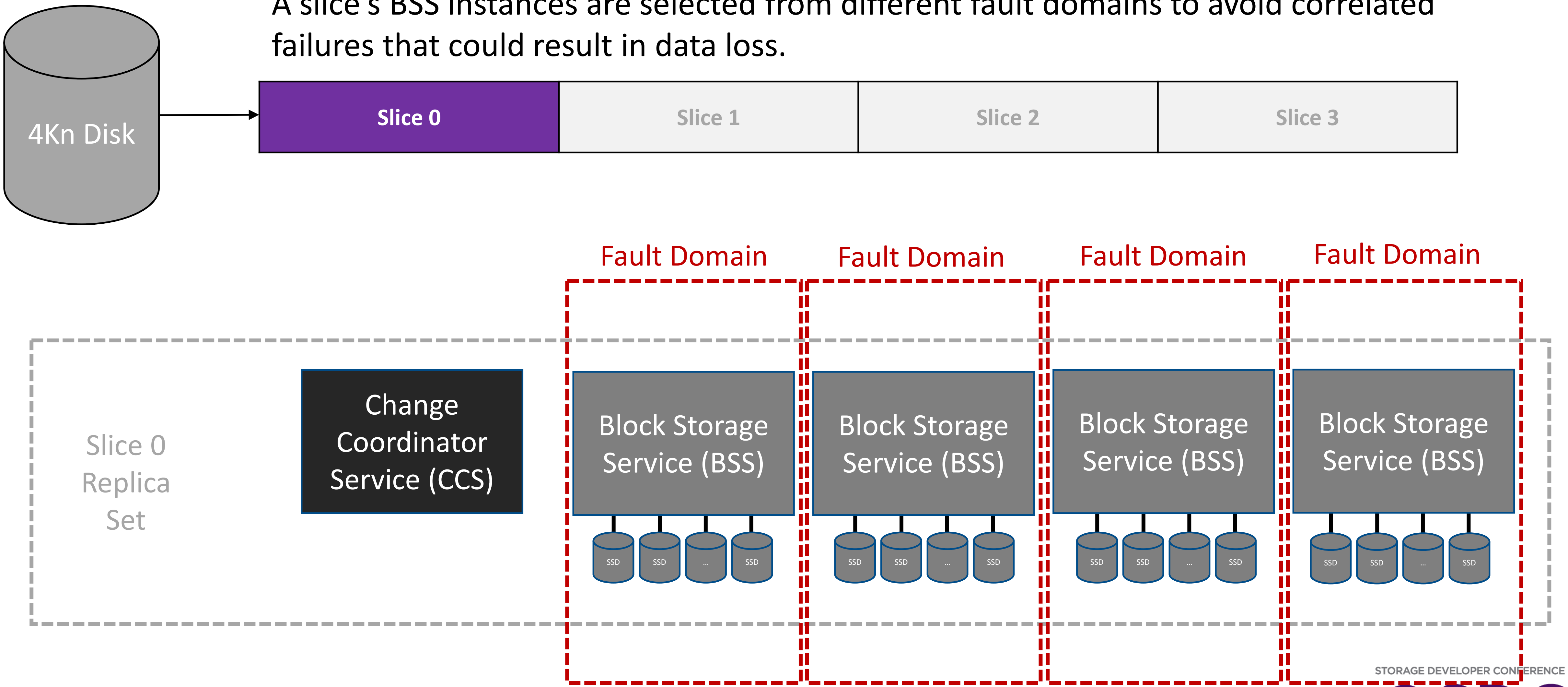
Each slice is owned by a replica set consisting of one CCS and N x BSS instances*.



* Number of BSS instances in a replica set is determined by disk durability requirements. Four is just an example.

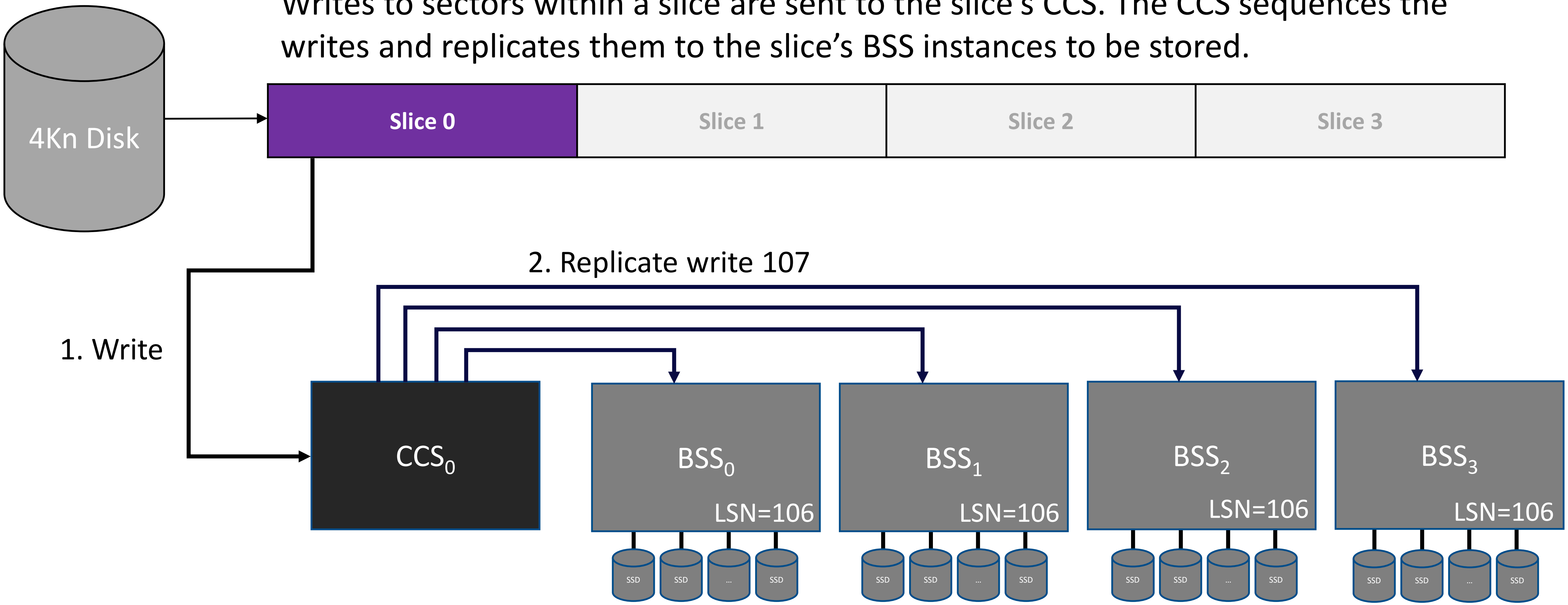
Slice Replica Sets…

A slice's BSS instances are selected from different fault domains to avoid correlated failures that could result in data loss.



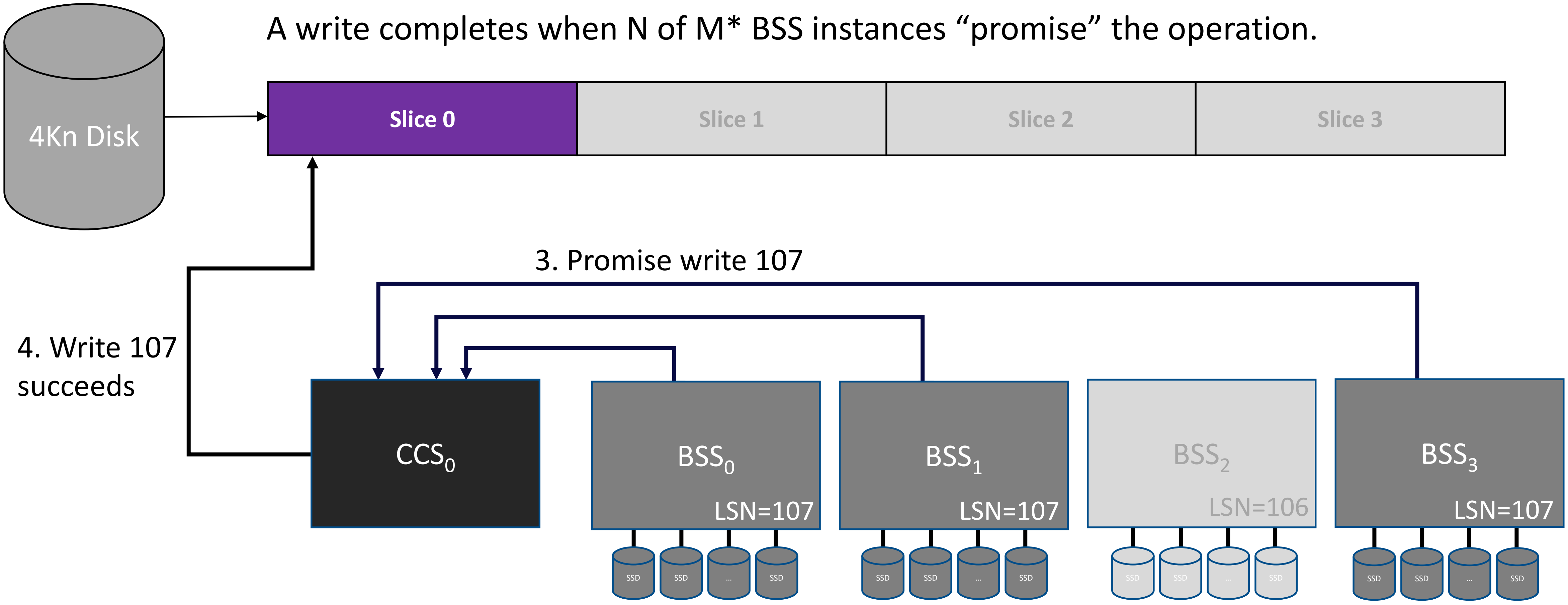
Writing to a slice

Writes to sectors within a slice are sent to the slice's CCS. The CCS sequences the writes and replicates them to the slice's BSS instances to be stored.



Writing to a slice…

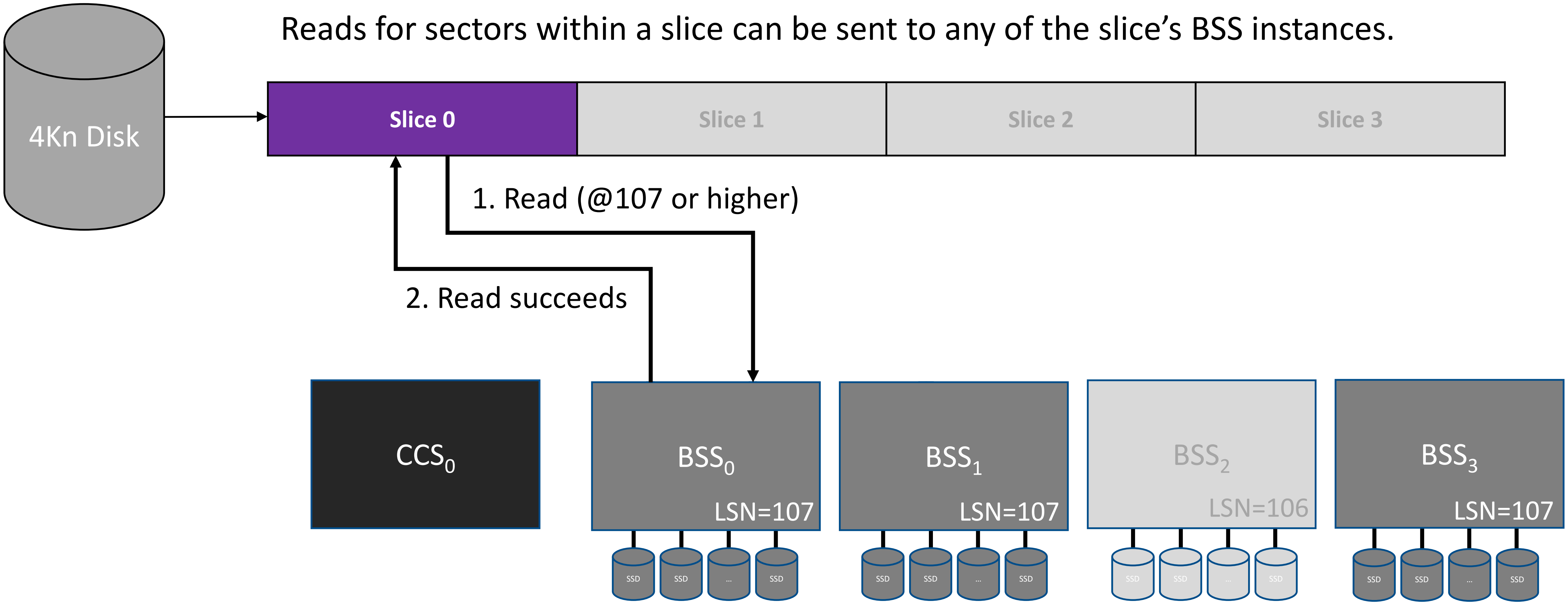
A write completes when N of M* BSS instances "promise" the operation.



* N and M are configurable. In this example, the slice is configured for a quorum of 3 out of 4 BSS instances.

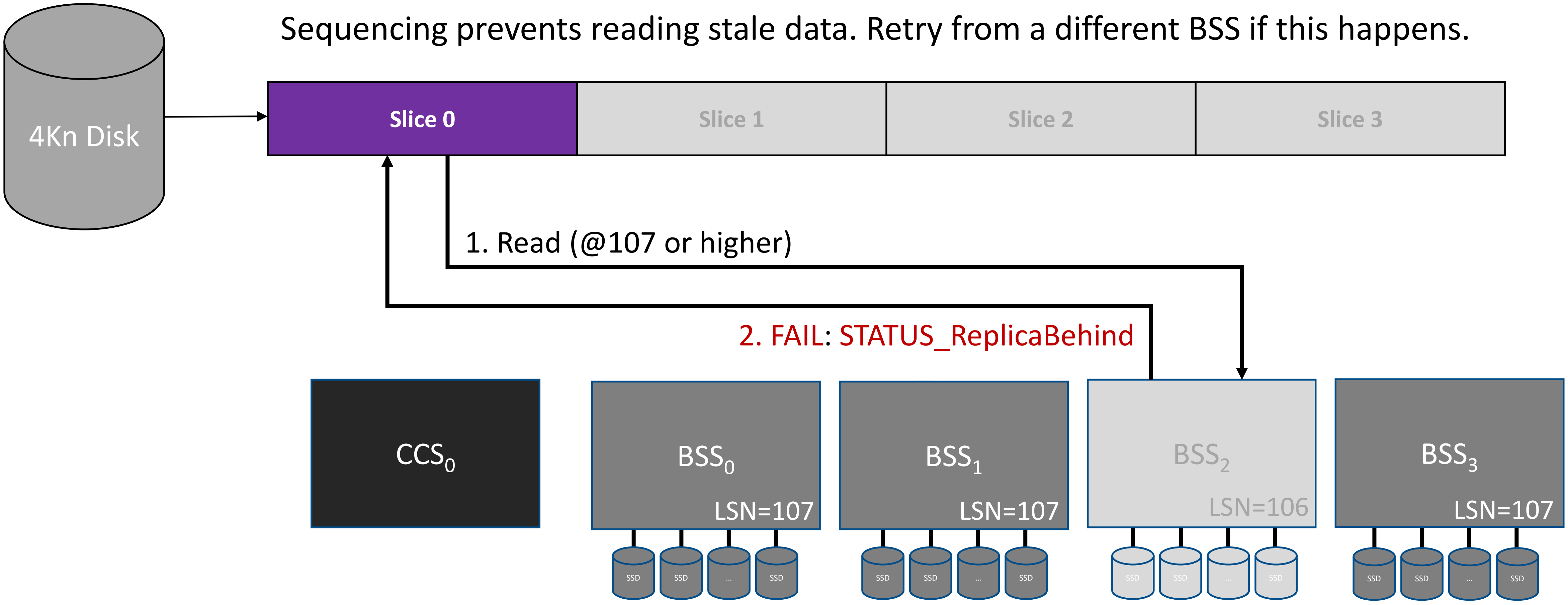
Reading from a slice

Reads for sectors within a slice can be sent to any of the slice's BSS instances.



Reading from a slice…

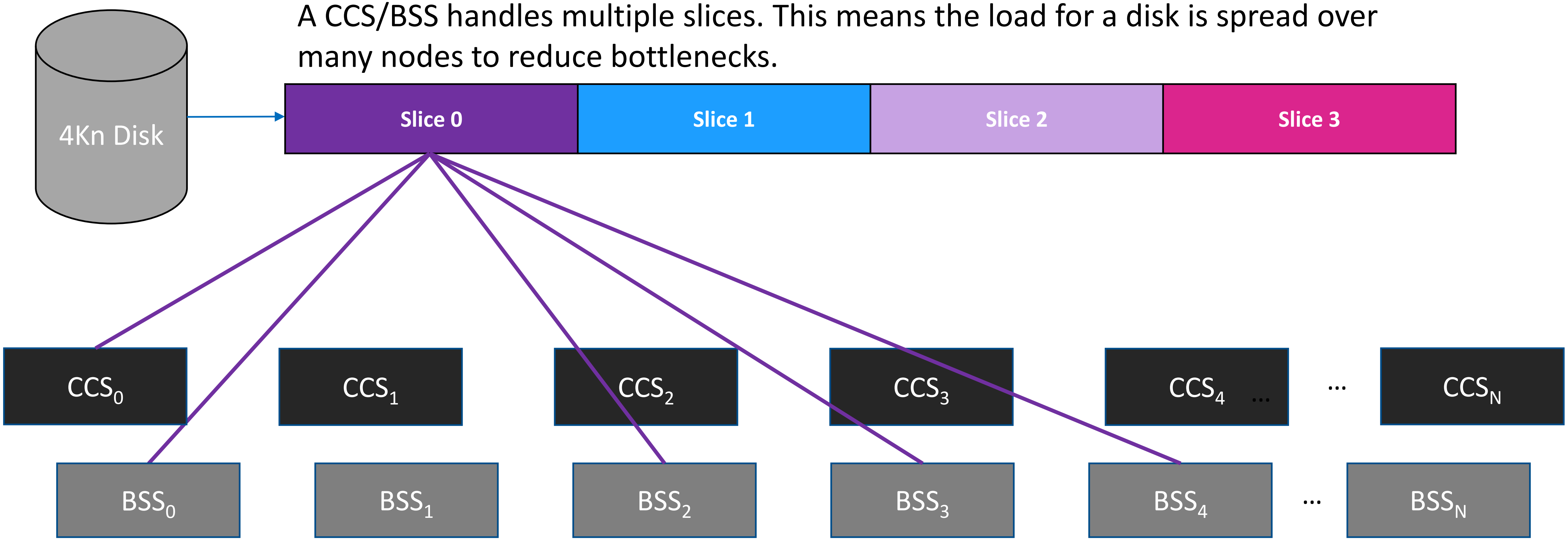
Sequencing prevents reading stale data. Retry from a different BSS if this happens.



Retries due to replica behind are rare in most workloads. Replica set can be configured with N == M to avoid if necessary.

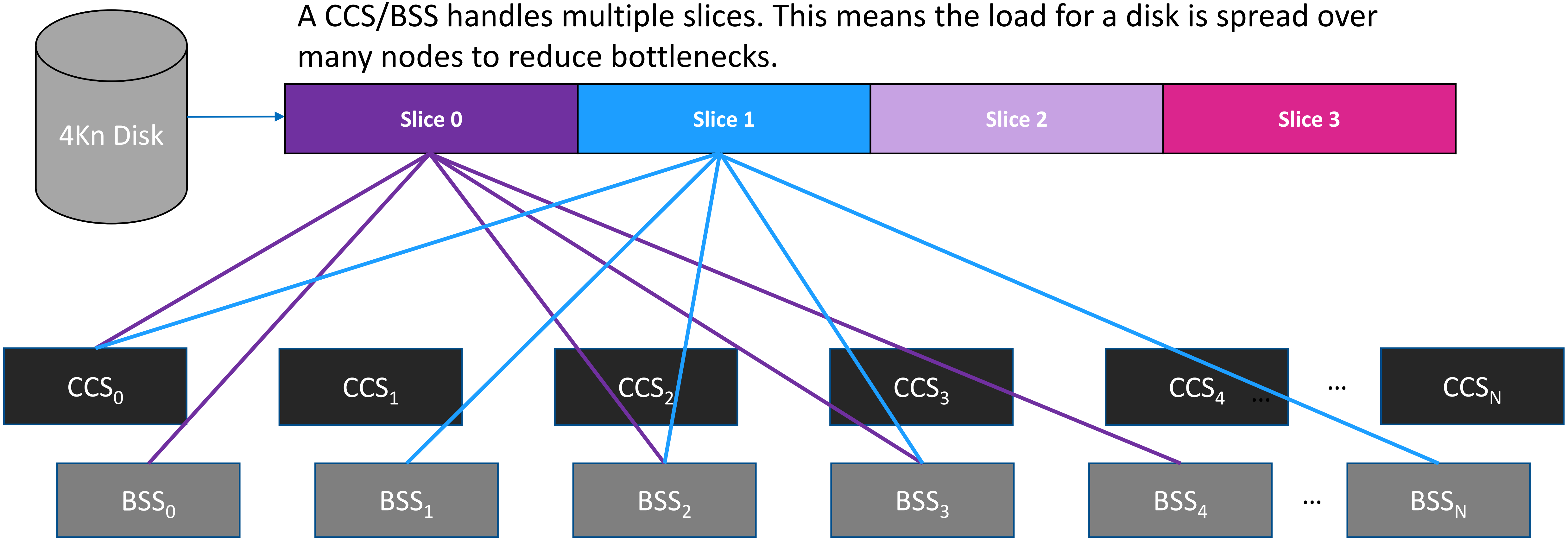
Slice Replica Sets…

A CCS/BSS handles multiple slices. This means the load for a disk is spread over many nodes to reduce bottlenecks.



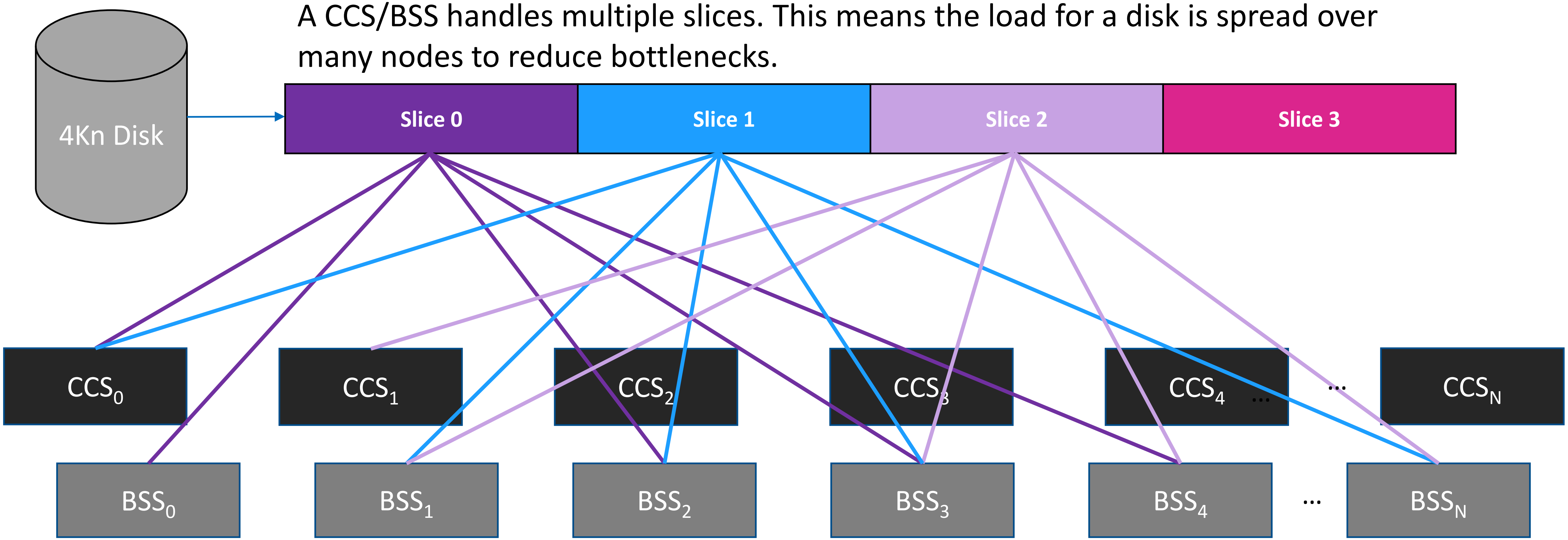
Slice Replica Sets…

A CCS/BSS handles multiple slices. This means the load for a disk is spread over many nodes to reduce bottlenecks.



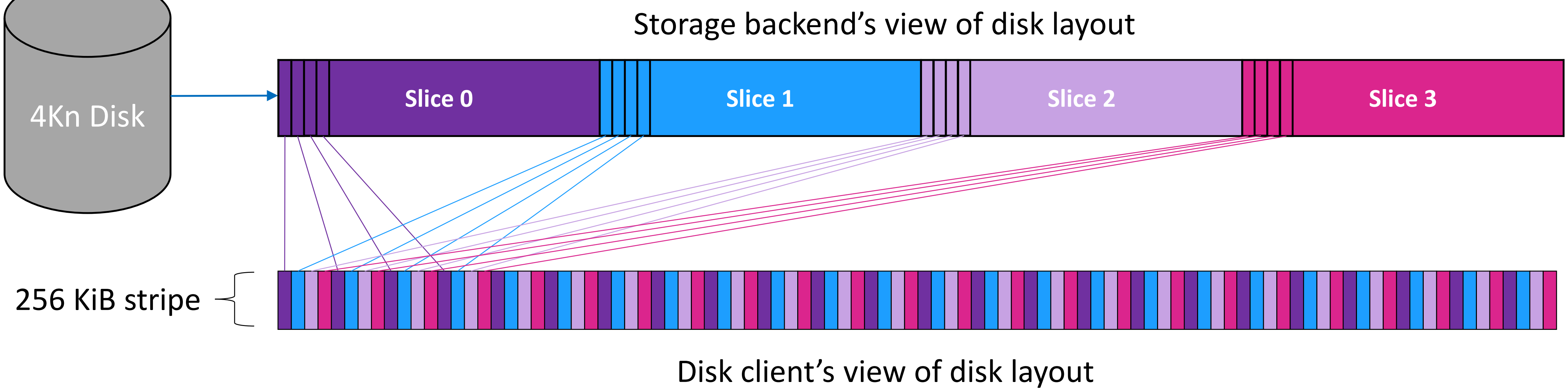
Slice Replica Sets…

A CCS/BSS handles multiple slices. This means the load for a disk is spread over many nodes to reduce bottlenecks.



STORAGE DEVELOPER CONFERENCE
SDC 22

Disk Layout – The Client's View



The disk client stripes data to help avoid replica set hot spots and improve performance.

VM LBA	Slice #	Slice Sector #
0	0	0
1	0	1
64	1	0
65	1	1
128	2	0

Example shows 4-way stripe set with 256 KiB stripe width, but is configurable per disk

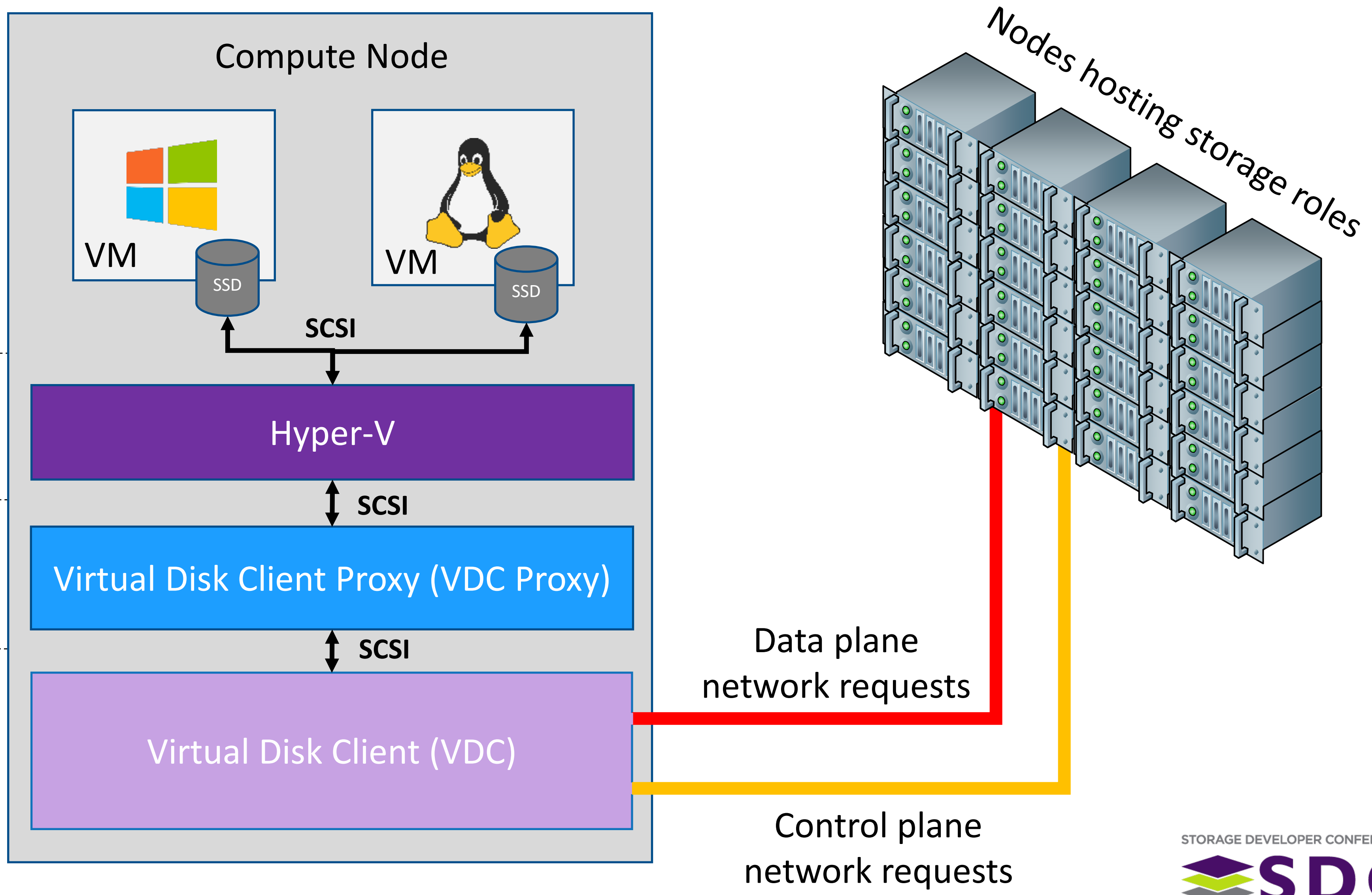
Disk Client

Virtual machines with attached Ultra / Premium v2 disks

Hypervisor

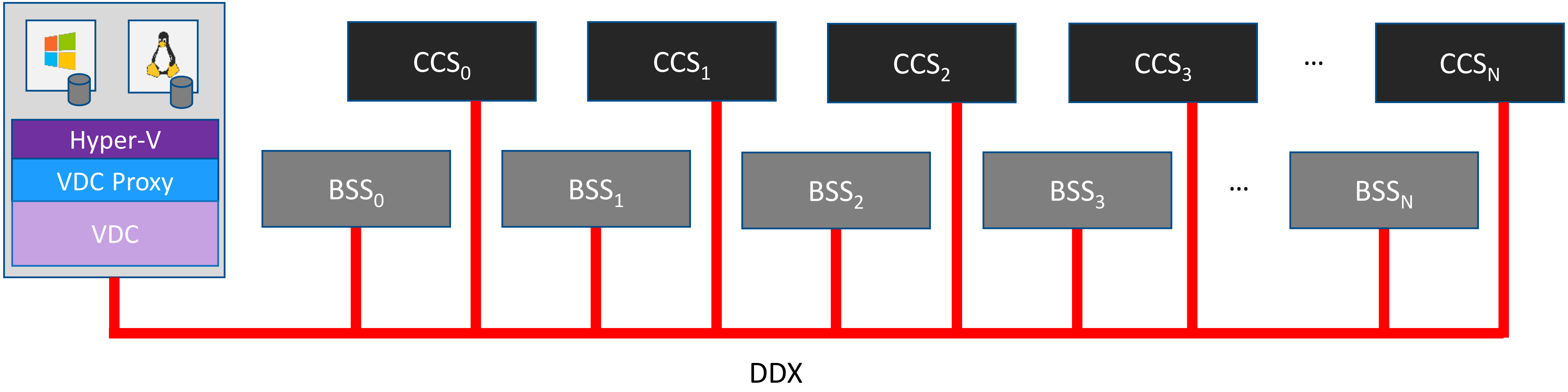
Allows disk client stack to be updated while disks are mounted and performing I/O.

Mount / dismount, report / handle errors, issue / throttle disk I/O requests, disk striping logic.



Direct Drive Data Plane

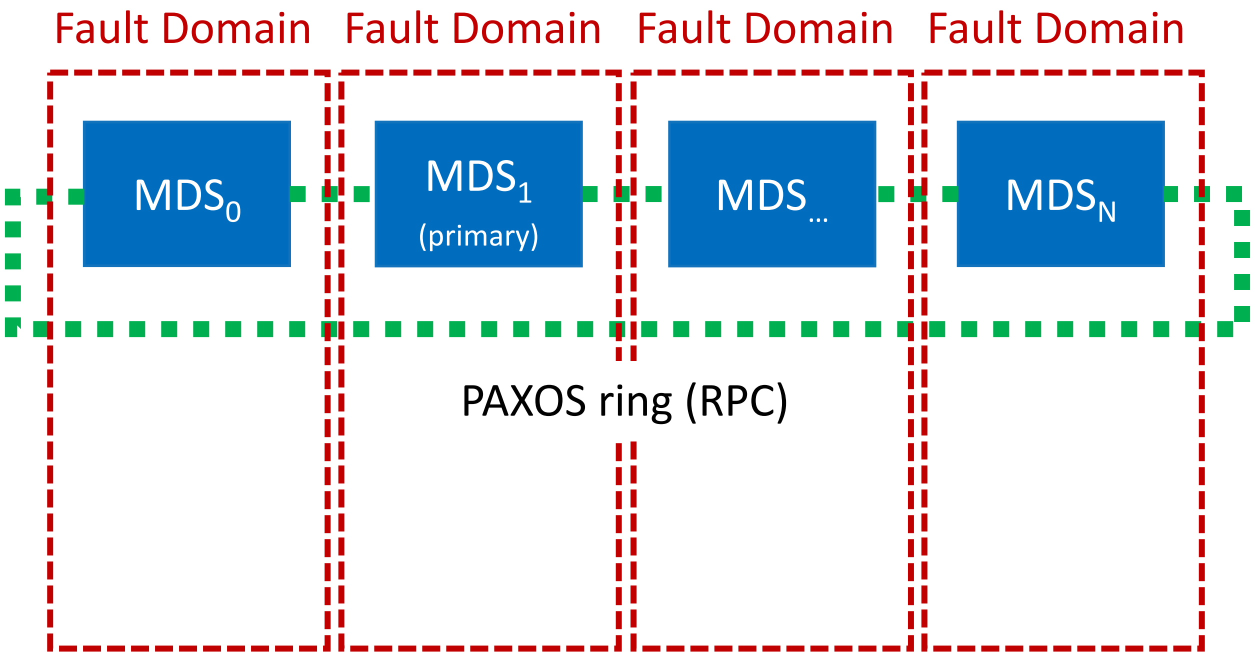
Data plane consists of the components that issue and perform disk I/O



Data plane requests use a custom storage/network protocol named DDX (more on this later).

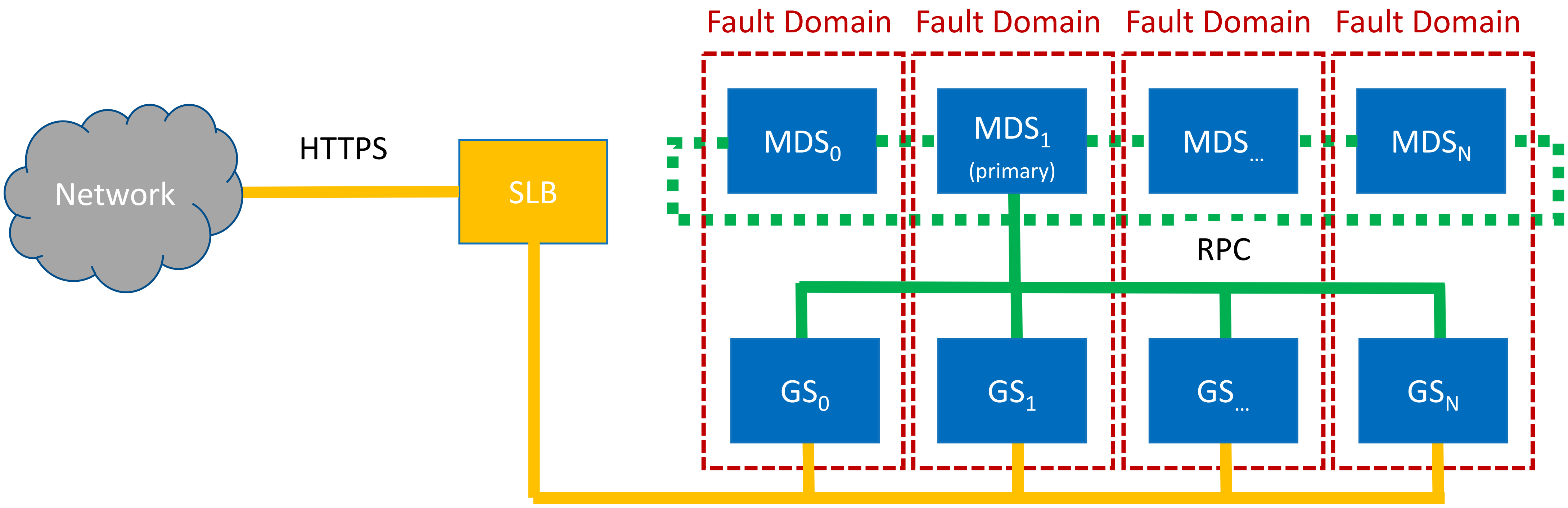
Metadata Service (MDS)

- Responsible for assigning CCS/BSS instances to slice replica sets
- Handles disk control requests: create, delete, mount, grow, etc.
- Handles error reports and issues corrective actions to data plane
- Spread across fault domains to prevent correlated failures
- PAXOS state replication
  - Primary
  - Secondaries
- RPC used for intra-cluster control traffic

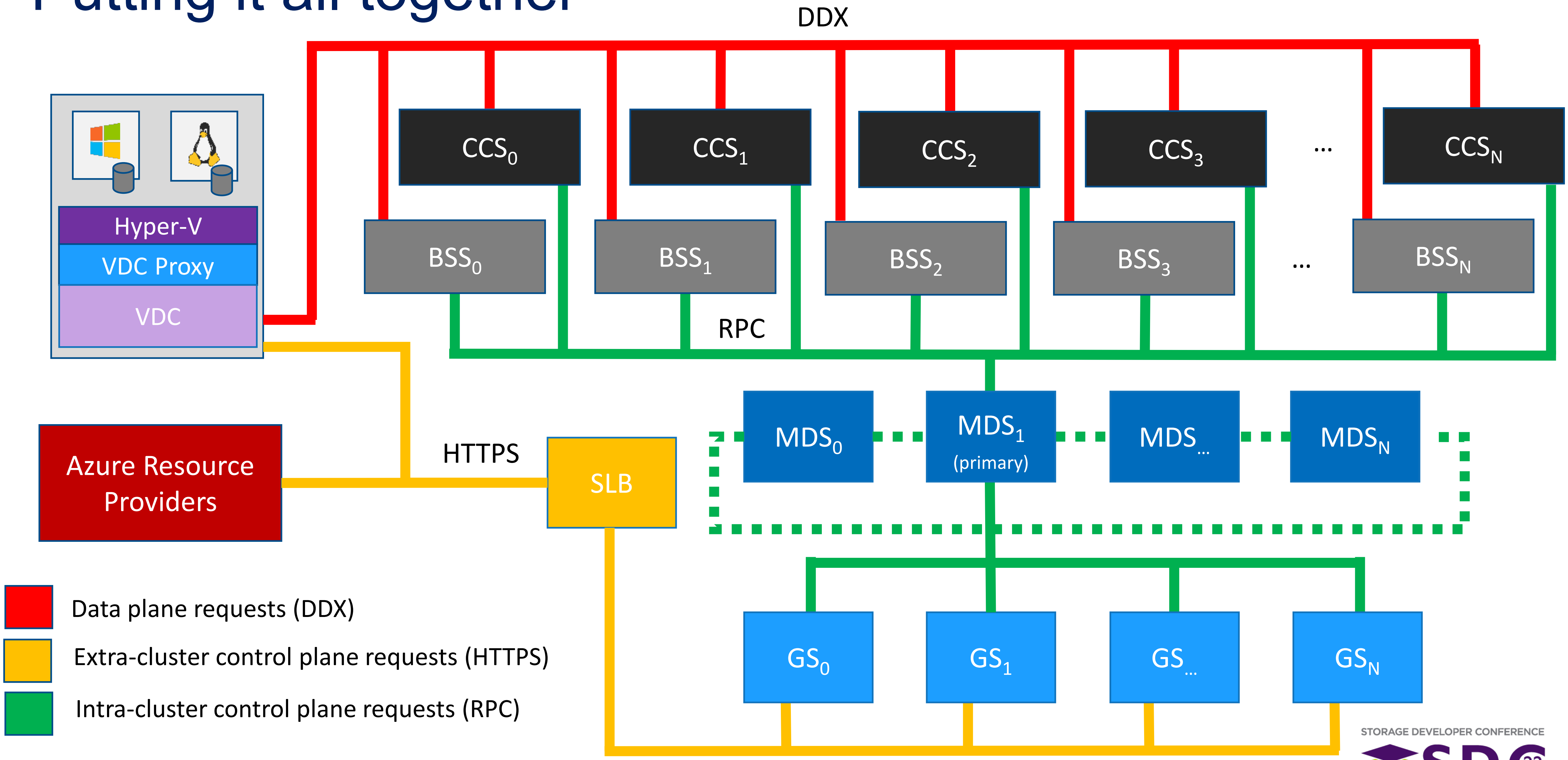


Gateway Service (GS)

- HTTPS front end that sits behind a Software Load Balancer (SLB)
- Spread across fault domains
- Authenticates and forwards extra-cluster control requests to Primary MDS



Putting it all together



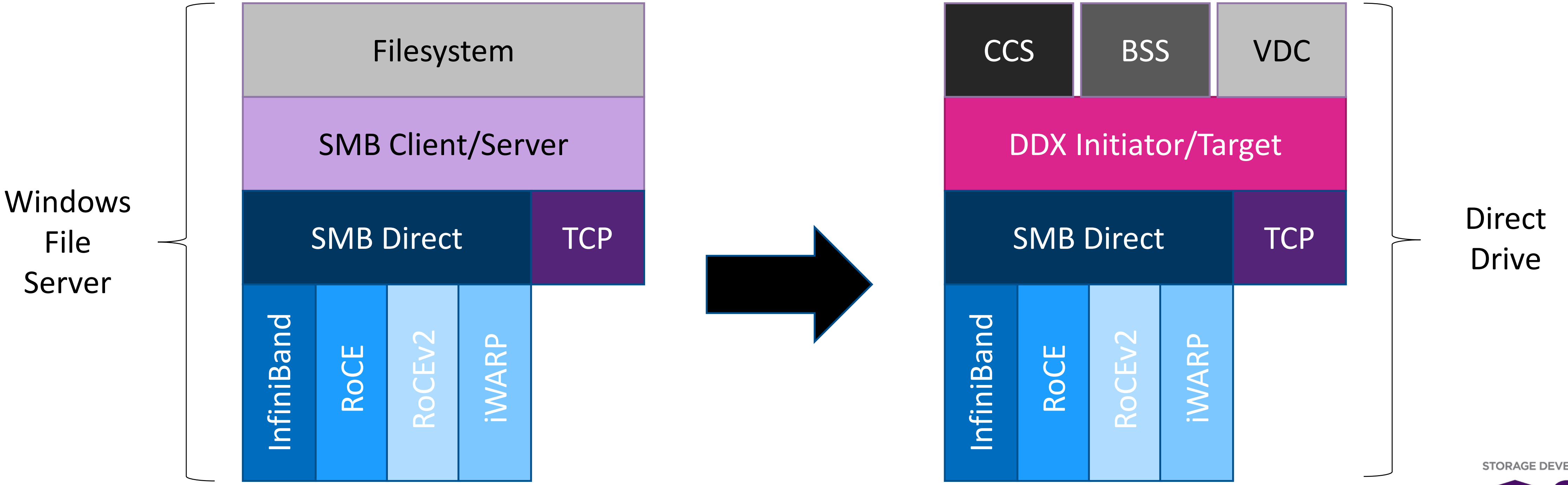
Notable Design Elements

DDX Protocol

- A purpose-built storage network protocol for Direct Drive data plane
- Why not NVMEoF, iSCSI, or other off-the-shelf block protocols?
  - Distributed nature of disk slices cannot be hidden from client to efficiently support:
    - Consistent reads/writes across slice replica sets
    - Shared disks (single-writer/multi-reader and multi-writer/multi-reader)
    - Crash consistent distributed snapshots
    - Disk migration
  - End-to-end diagnostic support baked directly into protocol
    - Activity IDs attached to requests and sub-requests to allow distributed log search
    - Request completions carry diagnostic data from responder needed to automatically identify problems
      - Time spent in queue
      - Time spent on network
      - Time spent waiting on storage media
      - Etc.
  - Agility
    - Need to be able to rapidly evolve protocol to address issues, implement new features, and take advantage of new opportunities

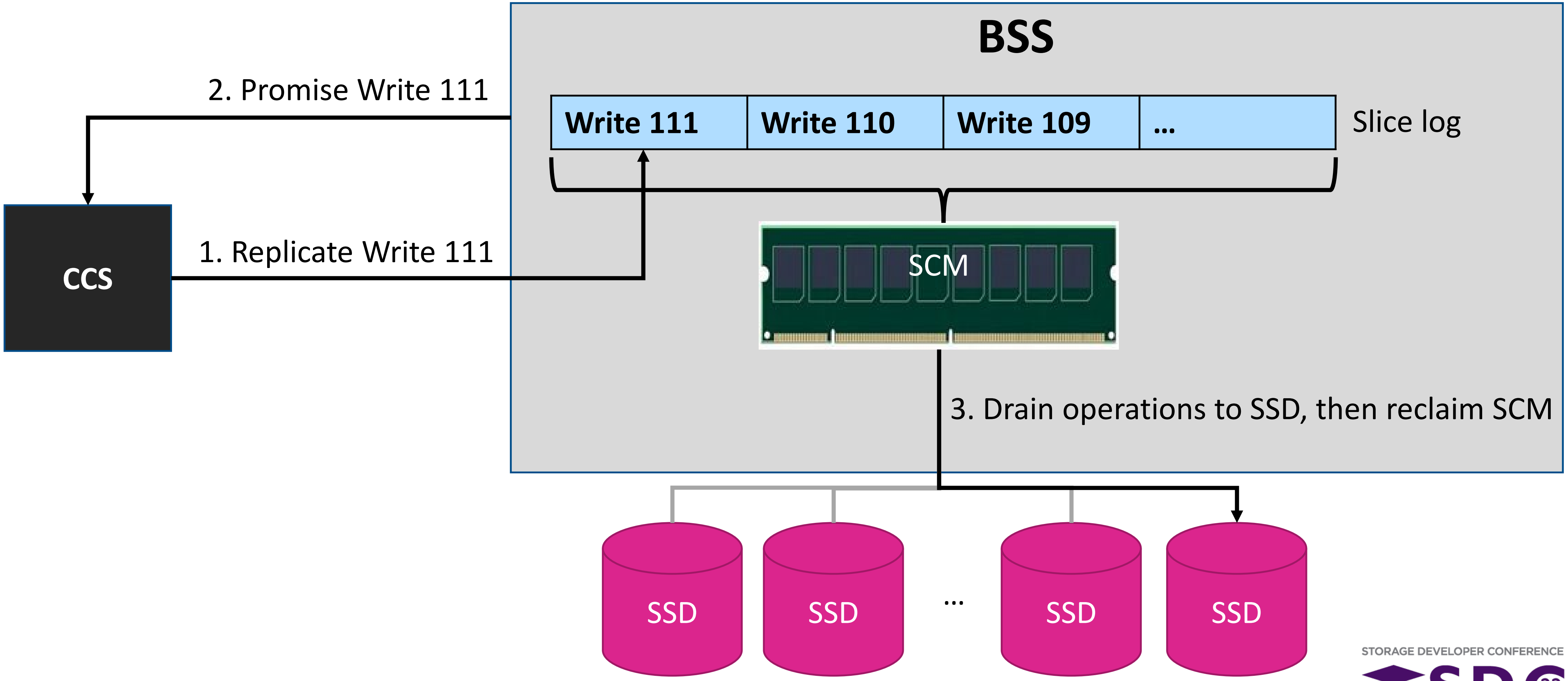
Remote Direct Memory Access (RDMA)

- Steal a page from Window Server 2012's playbook!
- SMB Direct (MS-SMBD) always meant to be a generic RDMA transport
  - Now transports the vast majority of Azure Disk traffic



SCM Writeback Cache

Hide SSD latency from disk client by using storage class memory (SCM)

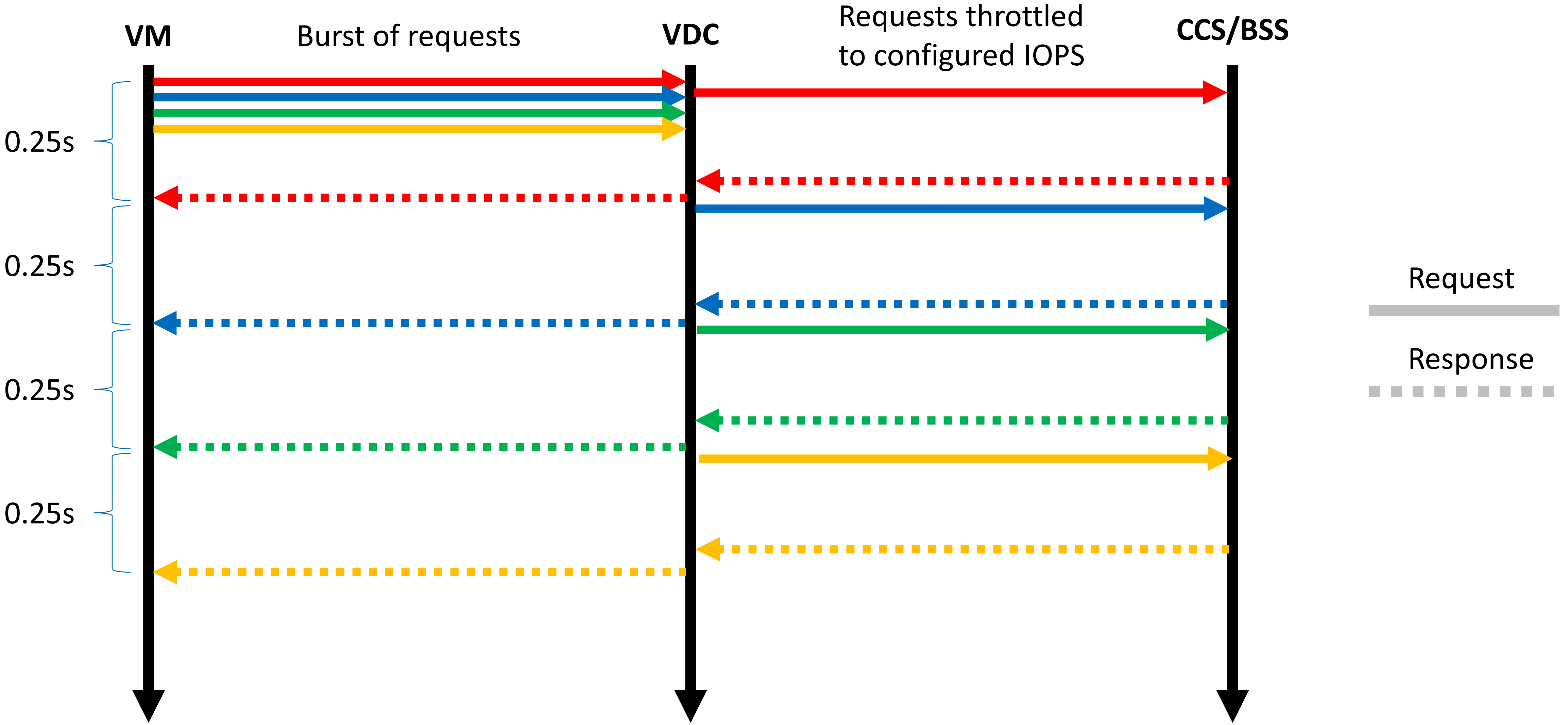


STORAGE DEVELOPER CONFERENCE SDC 22

Traditional I/O Throttling

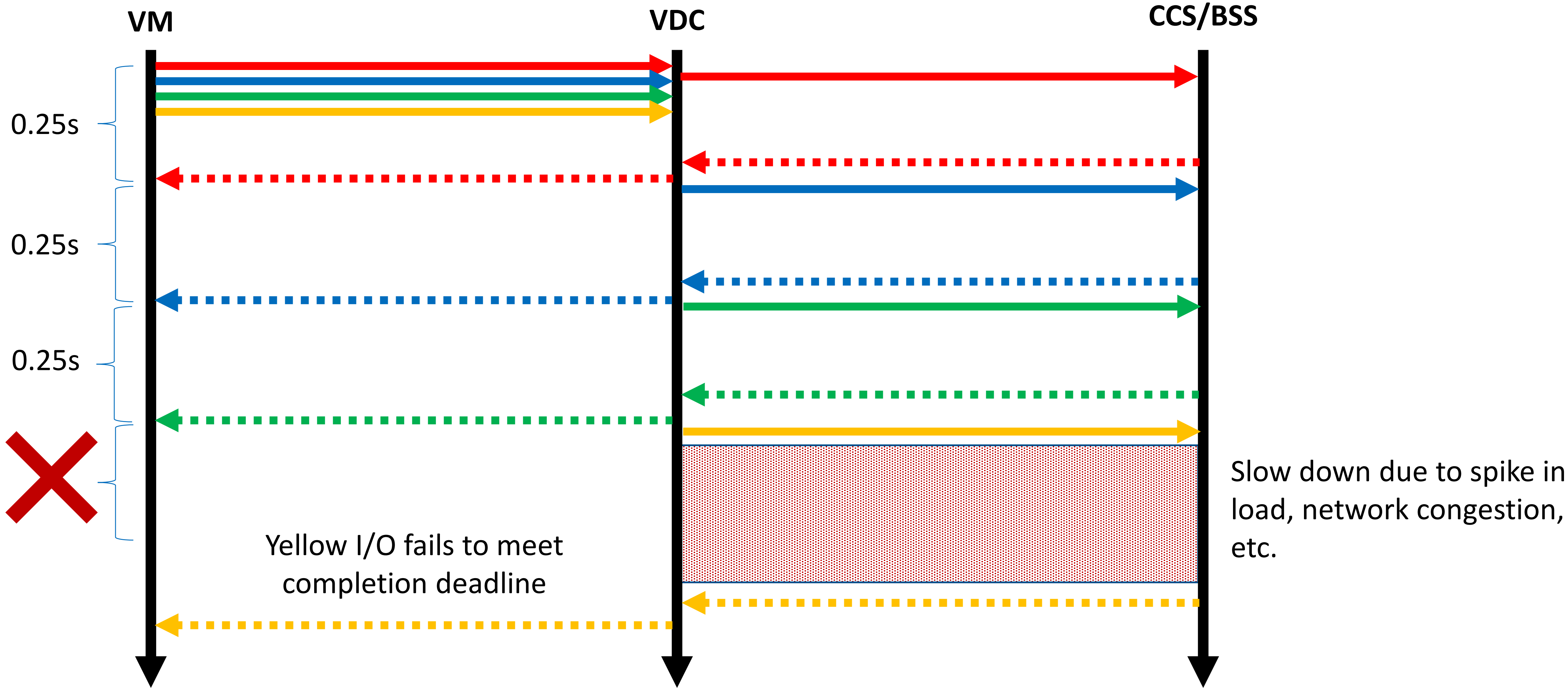
- Clients typically implement throttles in the initiation path:
  - Receive I/Os from VM
  - Delay issuing I/Os to backend to match disk’s configured IOPS
  - Complete I/Os back to VM as soon as they are completed by backend
- By the time the I/Os are issued, they must be processed ASAP to meet completion deadline.
- Anything that slows down transmission or processing of I/O risks violating the completion deadline, increasing disk’s latency distribution tail.

Traditional I/O Throttling



Assume IOPS configured to 4 for simplicity

Traditional I/O Throttling Disadvantage



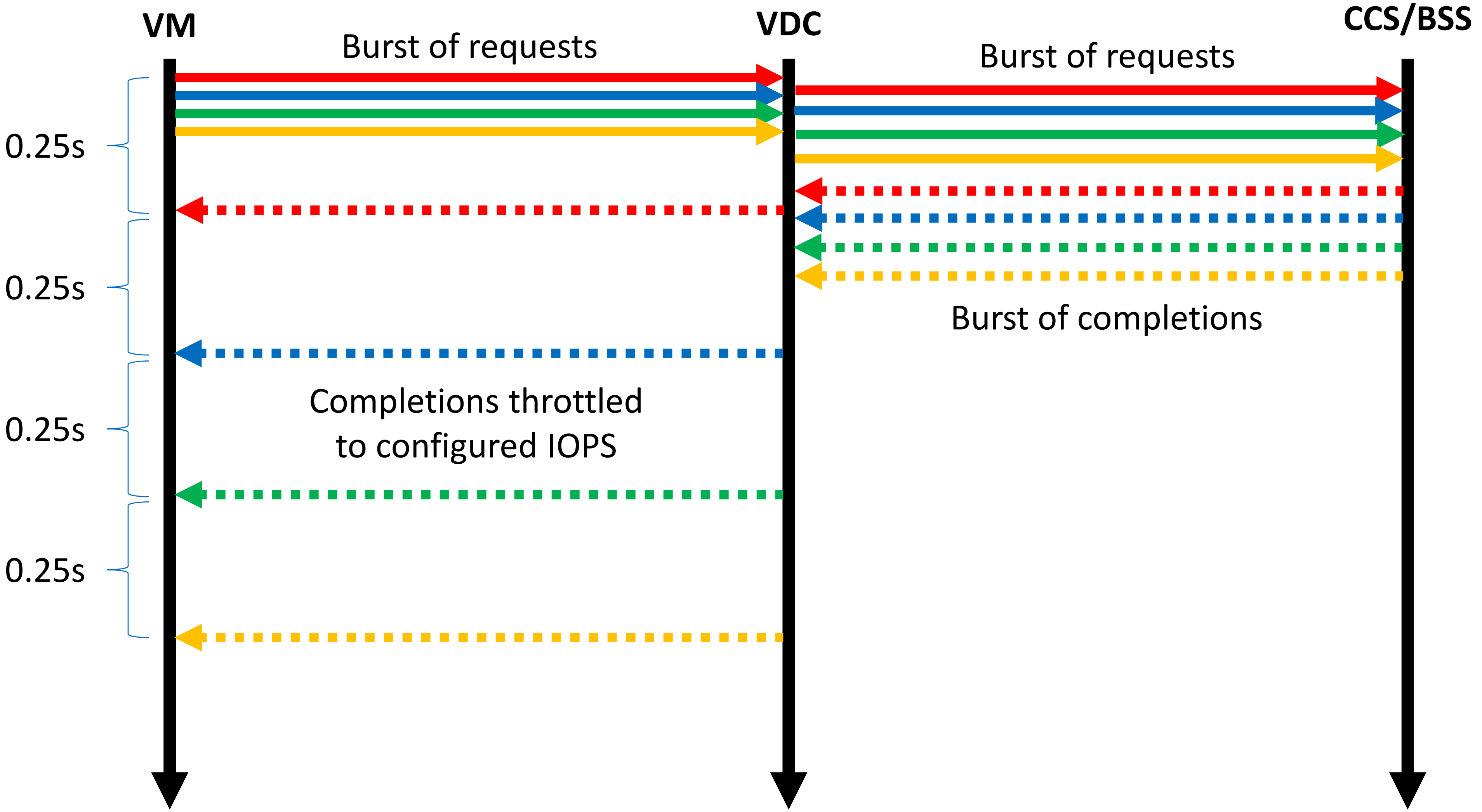
Assume IOPS configured to 4 for simplicity

Completion Side I/O Throttling

- VDC implements disk throttle in the completion path
  - Receive I/Os from VM
  - Issue I/Os received from VM to backend immediately*
  - Delay completion of I/Os to VM to match the disk's configured IOPS
- By allowing I/Os to be processed ahead of time, the system is more resilient to slow downs induced by load spikes, network congestion, etc.

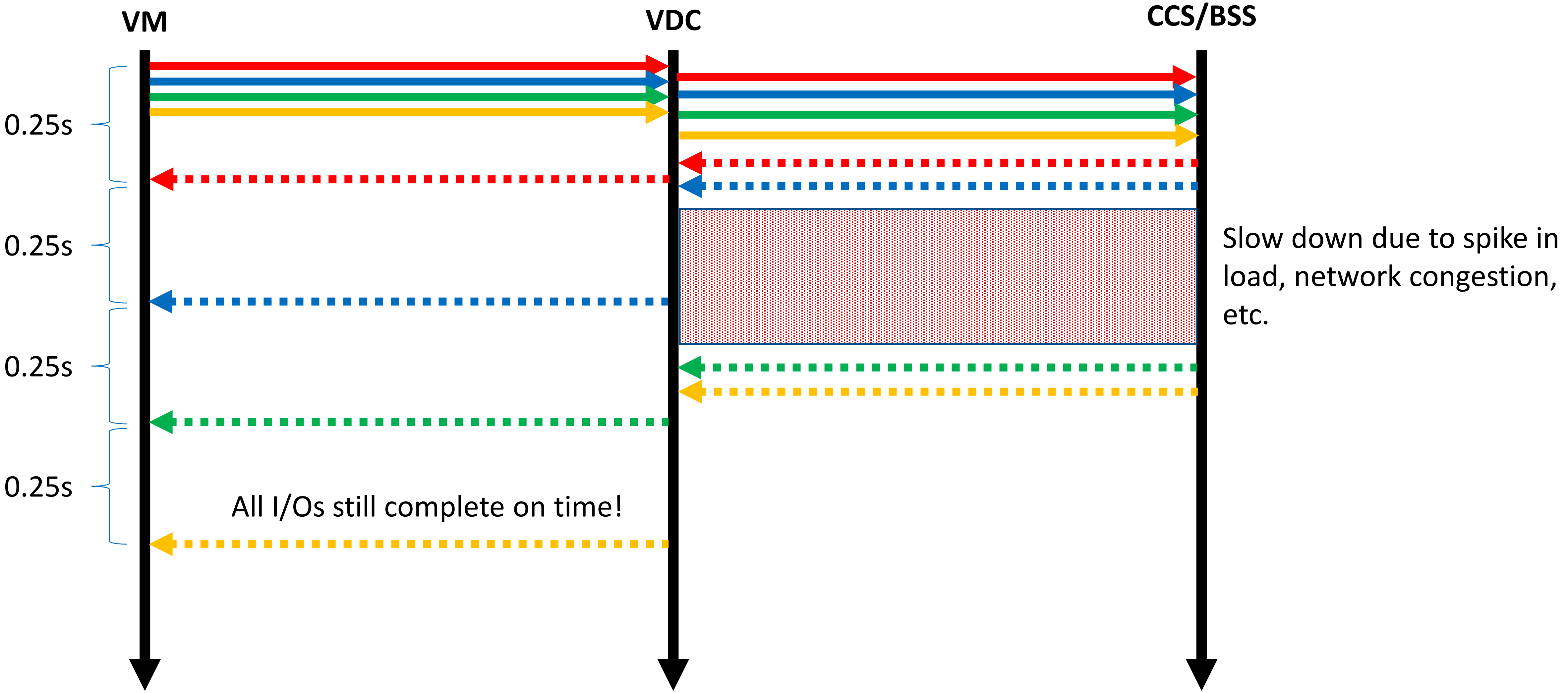
* VDC can still apply traditional initiator-side throttle if necessary to prevent overloading backend.

Completion-Side I/O Throttling



Assume IOPS configured to 4 for simplicity

Completion-Side I/O Throttling Advantage



Assume IOPS configured to 4 for simplicity

Questions?

Please take a moment to rate this session.

Your feedback is important to us.

