SSD Implementation of Live Migration

Dan Helmick
Principal Architect

SAMSUNG

Motivation for Live Migration

- **Why Migrate a workload?**
  - Load Balancing
    - Example:
    - AI training is long running without user interactions
    - Data Center (DC) load may vary as a function of the local time zone
    - Migrate the AI training to a Data Center experiencing reduced load due to night time
  - Data Center down time, errors, or other access anomalies
- **Why Live Migrate?**
  - Workload can continue to run without awareness of migration event
  - Minimizes downtime
- **Why enable Live Migration at the SSD?**
  - Allows the removal of SW shim layers on the IO queues
  - Reduces Host SW load
  - Improved storage access latencies

SAMSUNG

Overview of Live Migration

Host IOs are continuous

Pre-Copy Phase

- Initial copy of NS
- Iterative copies of NS data that is changed by Host

Suspend and Copy Phase

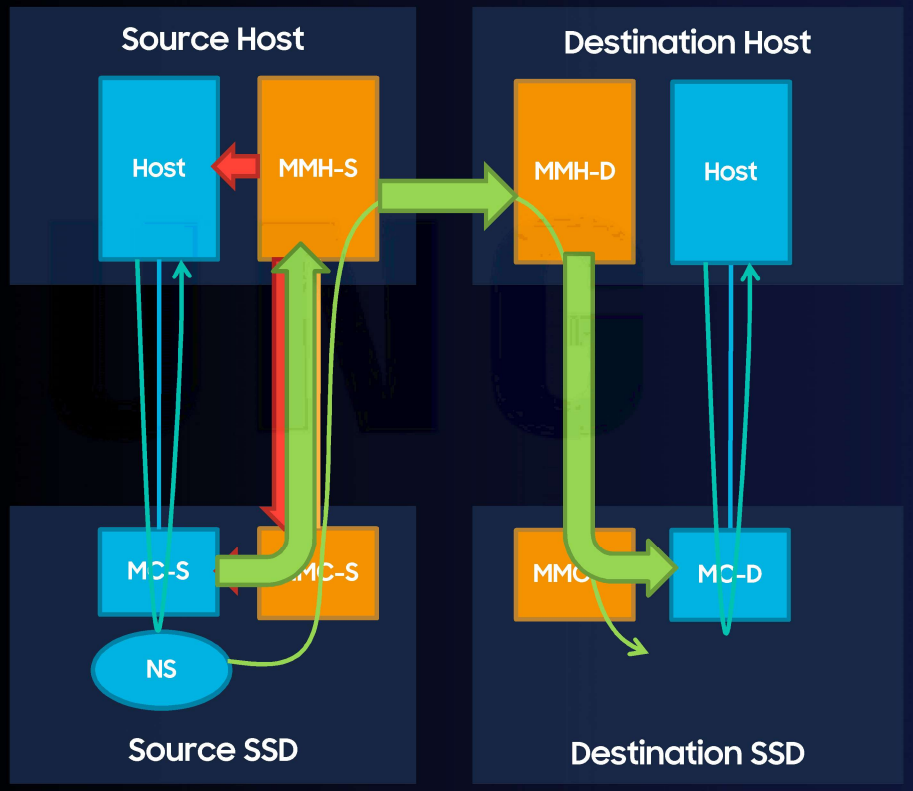
- Suspend MC-S
- Final copies of NS data that has been changed by Host

Post-Copy Phase

- Move Host and copy MC State
- Resume

Acronyms

- Migrating Management Host - MMH
- Migration Management Controllers - MMC
- Migratable Controller - MC



SAMSUNG

An Example System Set-up

Virtual Machines (VMs) and VM Monitor (VMM)

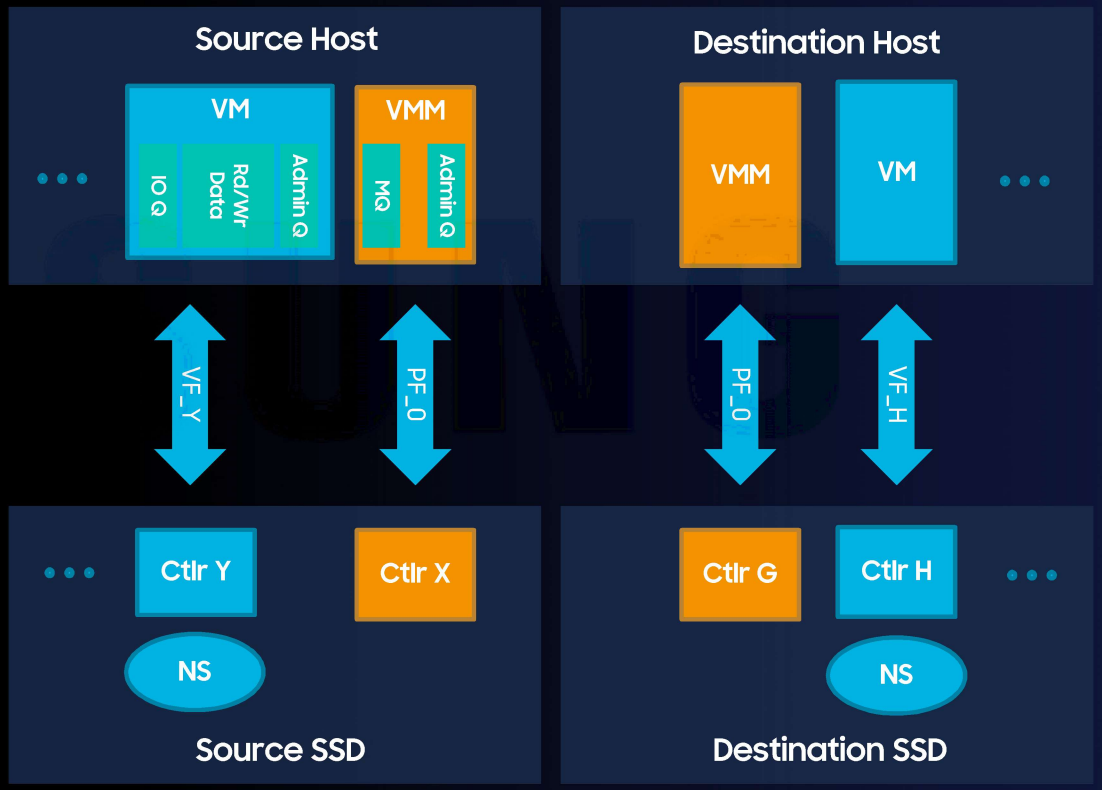
- 1 VMM to many VMs
- VMM = Migration Management Host
- All Live Migration (LM) commands come through VMM
- VM is unaware LM is happening
- Logging in the MQ in the form of Migration Queue Entries (MQE)

SSD example with SR-IOV

- Migration Management Controller = Primary Controller (Ctlr) on PF_0
- Migratable Controller = Secondary Ctlr per VM on VF_Y and VF_H

Destination vs Source

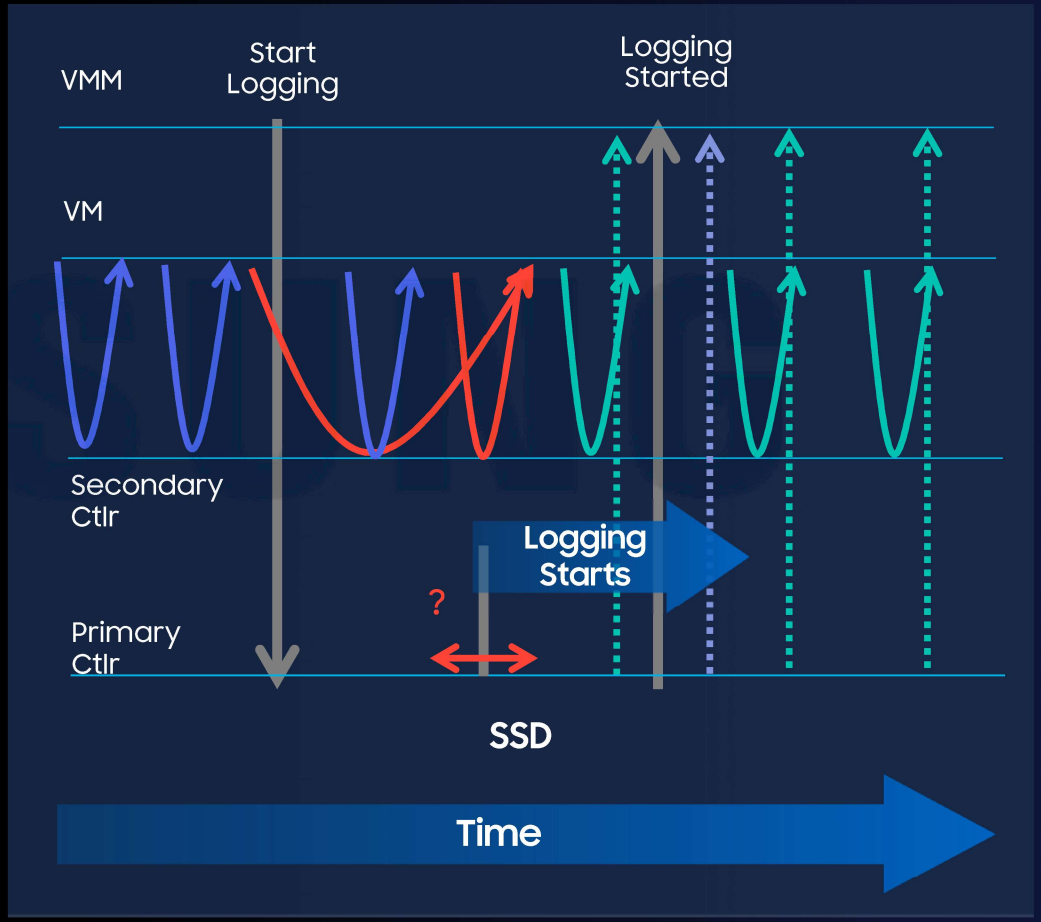
- Similar setups
- Destination VM may send writes/reads to Ctlr H prior to "start"
  - Destination VM's commands may be generated by VMM prior to migration



SAMSUNG

Pre-Copy Phase: Start Logging

- **VM continues to interact with Secondary Ctlr on SSD (Rd/Wr)**
  - Race Conditions are a concern
  - Solved by ordering of "Start Logging", "Logging Started", and then "Initial NS Migration"
- **"Start Logging" Command Flow**
  - VMM sends "Track Send" with "Log User Data Changes" option while IOs continue
  - Primary Ctlr begins tracking all requested MQ events occurring in VM's Ctlr (Secondary Ctlr)
  - Primary Ctlr completes "Start Logging" Command
    - SSD Promise: All potentially log-able commands will now be logged
- **VMM has successfully started logging in MQ**
  - Relationship of Logging Start and some commands is unknown
  - Unknown timing of where Logging Start occurred with respect to Completion of Start Logging command
  - "Logging Started" ensures
    - All prior commands in flight have finished
    - All future commands in flight will be logged



SAMSUNG

Pre-Copy Phase: Destination Preparation

Destination Precondition

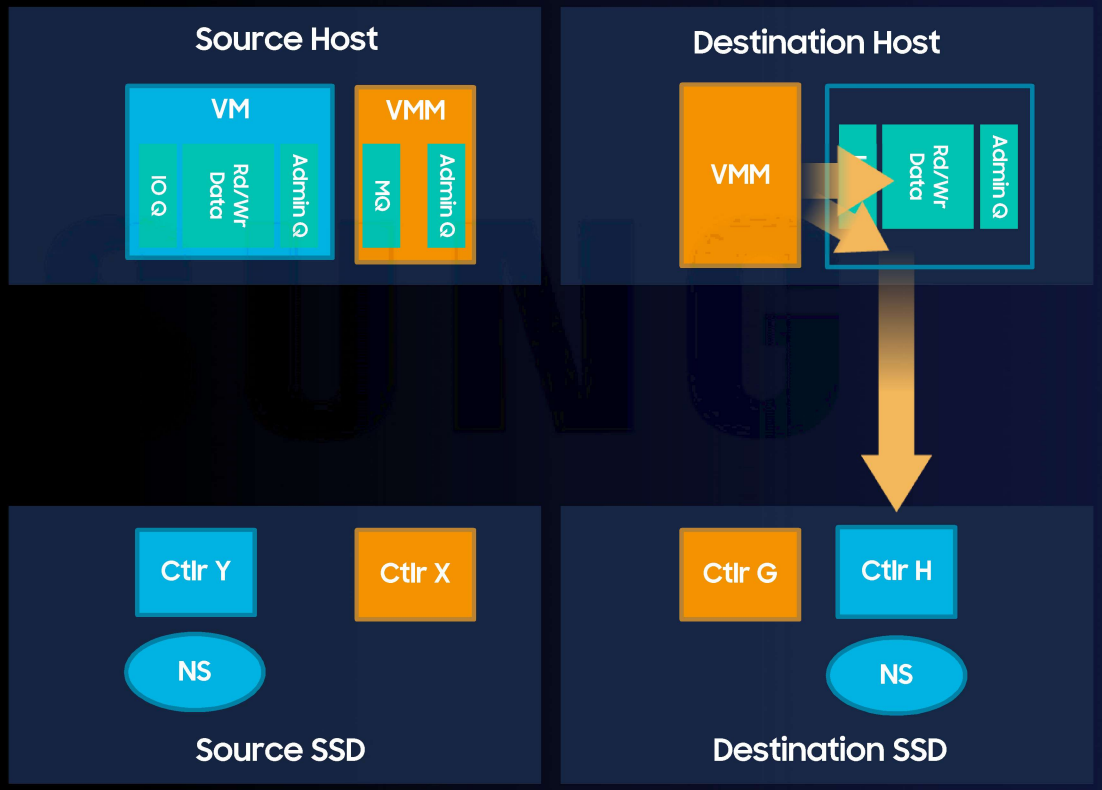
- Available Secondary Controller
- Available Host side VM resources

Standard NVMe commands for initializing Destination SSD

- Initialize any Queue and IO command structures needed
- Create NS

Above illustrates one potential flow, but other options exist

- Ex: Shared NS created by VMM on Ctlr G



SAMSUNG

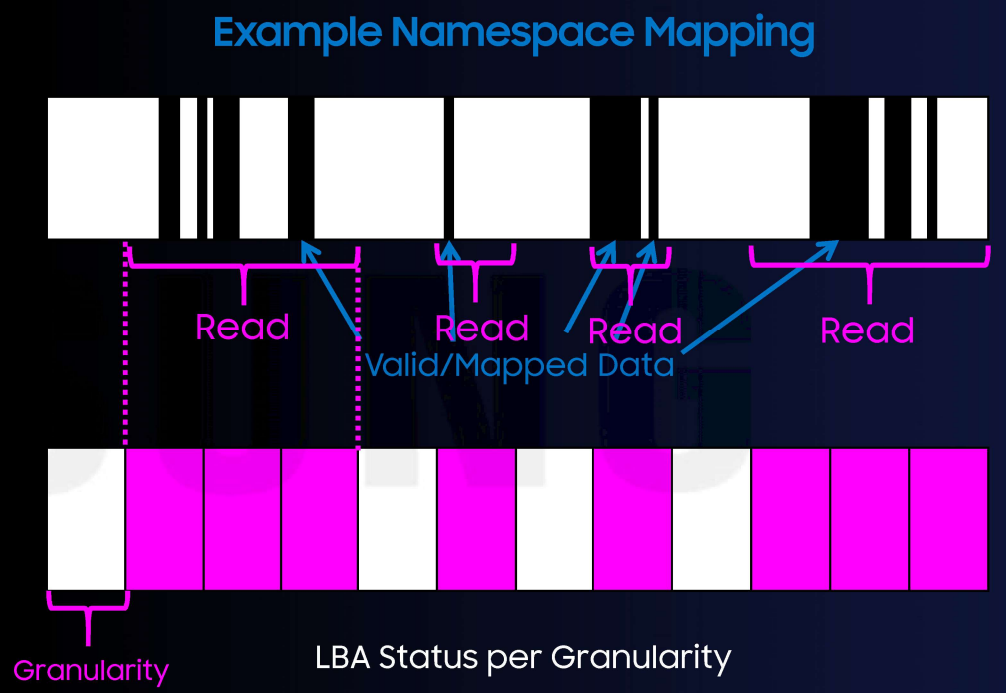
Pre-Copy Phase: Initial NS Migration

Option 1:

- VMM copies entire VM NS
  - Not optimal for sparsely written data
  - <See example on right>

Option 2:

- VMM sends Primary Ctlr: Get LBA Status
  - Granularity: Set by SSD
    - Customer requirements discussion
- Primary Ctlr
  - Returns results with granularity restrictions
  - Any data state other than deallocated is returned as mapped
    - Ex: Read Uncorrectable
- VMM
  - For each mapped LBA status
  - Submitted as Read of Secondary Ctlr's NS

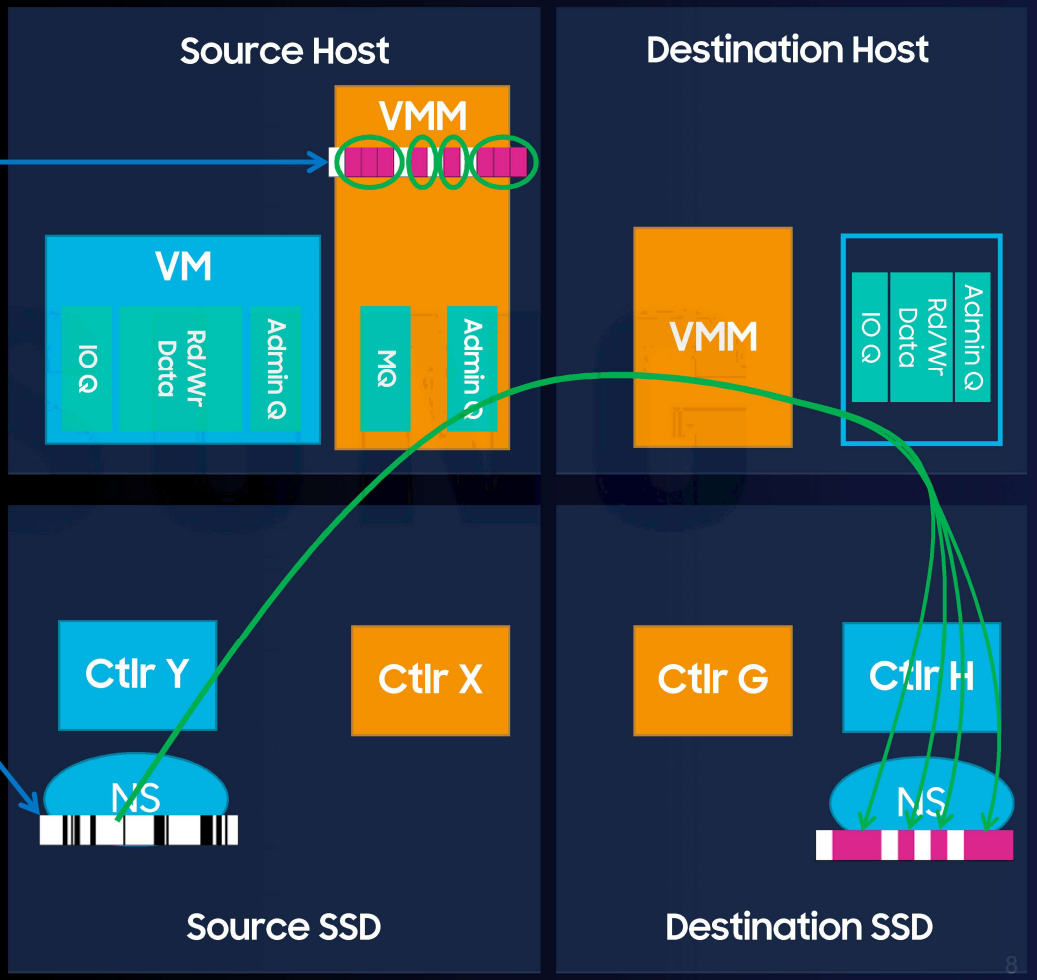


For more info: TP4165 Tracking LBA Allocation with Granularity

SAMSUNG

Pre-Copy Phase: Initial NS Migration to Destination

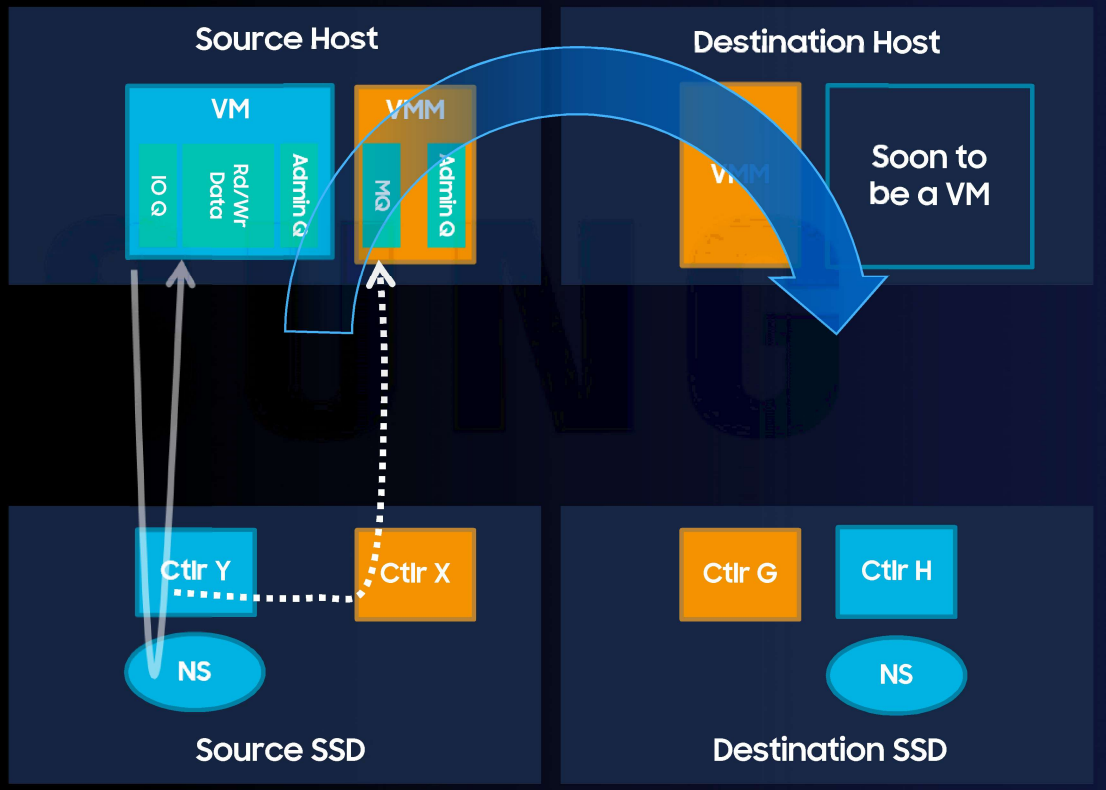
- VM's NS Mapping
- Returned LBA Status per Granularity
- VMM submits Read to Secondary Ctlr's NS for each contiguous mapped LBA range
- New NS is populated with no dependence of Source SSD's granularities



SAMSUNG

Pre-Copy Phase: Iterative Data Copy

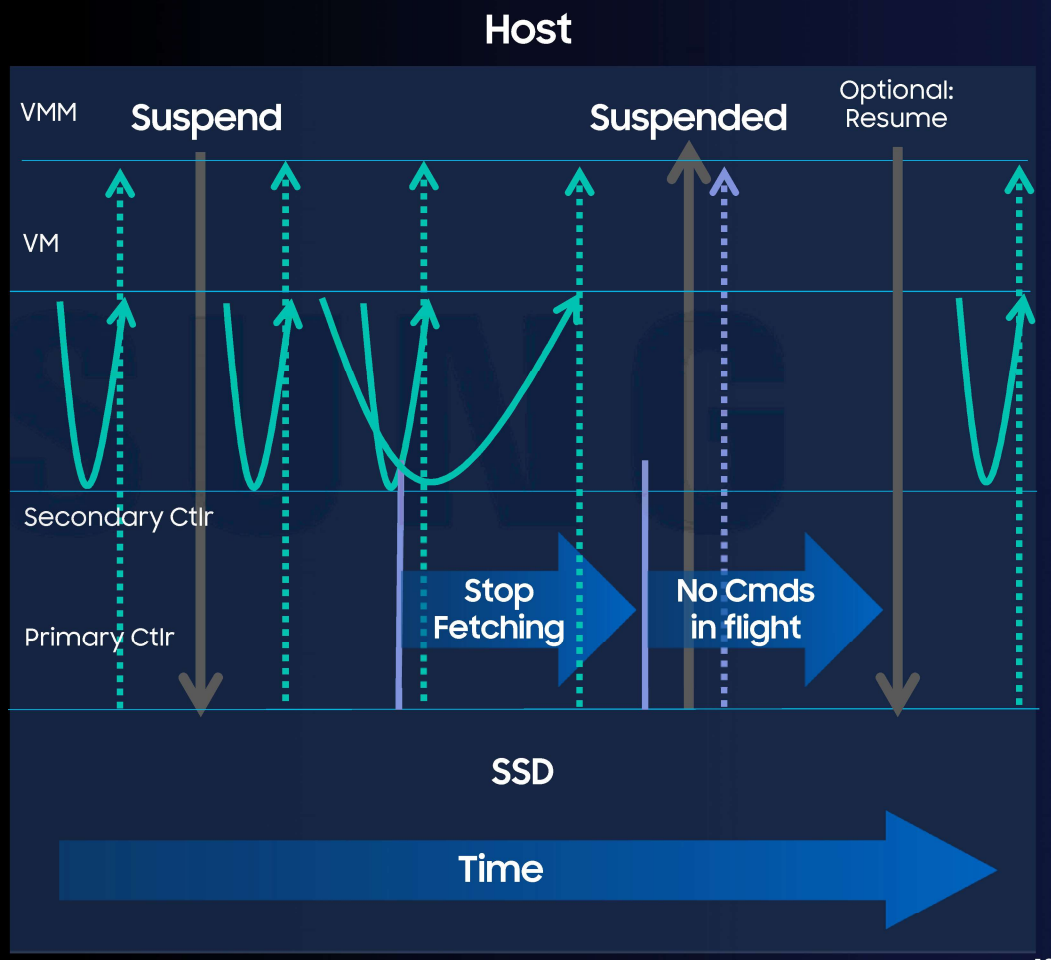
- Ongoing
  - VM has continued to Rd/Wr to Source NS
  - Source Primary Ctlr X has continued to log all writes to VMM
  - Copying from Source SSD to Destination SSD takes time
- Source Drive View
  - Has experienced Reads from initial copy of Source NS to Destination NS
  - Continues to experience additional Reads from VMM parsing the posted MQEs
    - VMM is continuing to catch up to the VM's activity
  - Data is written to Destination Secondary Ctlr NS
- Note: No Memory Tracking Discussed
  - NVMe-oF focused feature
  - Enterprise Processors have this capability in their IOMMU
  - Recommend full memory copy for those systems lacking this capability



SAMSUNG

Suspend and Copy Phase: Suspend

- **VMM decides to complete/execute the migration**
  - VMM issues Suspend Command to Primary Ctlr
- **Suspend Command Flow**
  - Secondary controller stops fetching new commands
  - Secondary controller completes all commands in flight
    - Success vs Error are both acceptable
  - All CQEs are properly returned to VM
    - With any MQEs for logging
  - Primary Ctlr completes the Suspend command to VMM
    - Log this successful Suspend in the MQ
- **Suspended status Summary**
  - SQE/CQEs may be on the SQ/CQs of the VM
- **Source SSD**
  - Must be prepared for potential Resume Command
    - Perhaps due to a system error
    - Conceptually Resume/Start should behave the same on both Source and Destination
    - Except: Source SSD would continue logging
  - If not resumed, expect Secondary Ctlr to be reset.
- **VMM will**
  - Parse all remaining MQEs
  - Copy any remaining data to Destination Secondary Ctlr NS



SAMSUNG

Post-Copy Phase: Copy Final Data and Migrate Controller State

Final Data Copy Iterations from MQ Parsing

Get/Set Controller State

- Reads Ctlr Y out to the VMM
- VMM Writes Ctlr H into the Destination SSD

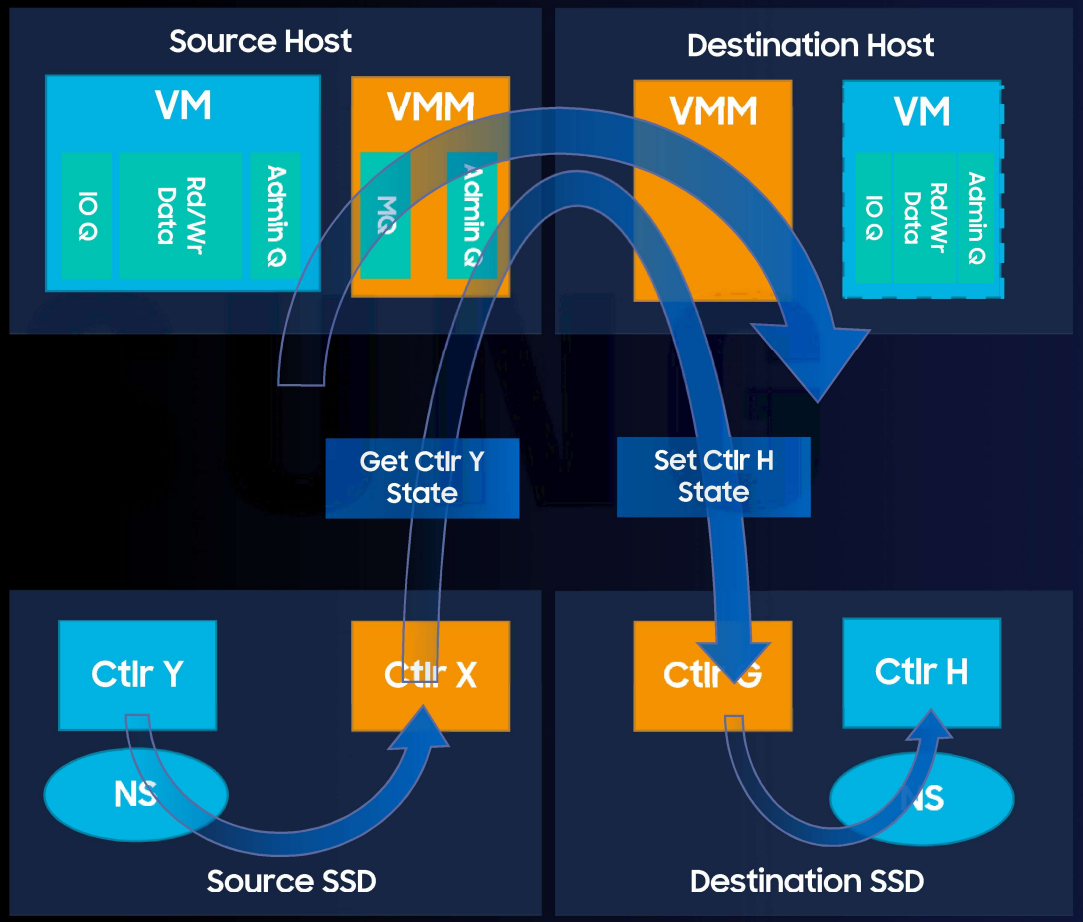
VMM will migrate the VM

From SSD's view of Resume

- Same behavior:
  - Resume Ctlr Y sent to Ctlr X
  - Resume Ctlr H sent to Ctlr G
- One difference: unlikely Ctlr G has enabled logging on Ctlr H

Nominal NVMe Flows

- Source VMM will clean up and reset Ctlr Y and NS



SAMSUNG

Learn More about Live Migration!

Samsung's Booth: NVMe Live Migration Proof of Concept

SPOS-203-1: NVMe® Live Migration, High Availability & Event Notification

- Wednesday 8/7 at 3PM-4:05PM
- Chairperson
  - Mike Allison - Samsung
- Panelists
  - Nicolae Mogoreanu - Google
  - Chaitanya Kulkarni - NVIDIA
  - Myron Loewen - Solidigm
  - Klaus Jensen - Samsung
  - Lee Prewitt - Microsoft

Thank You