Ultra Low Latency NVMe-oF Controller Design

Swati Chawdhary, Samsung
Agenda

- **Overview**
  - Storage Disaggregation and its Challenge
  - NVMe-oF Protocol
  - Existing NVMe-oF JBOF Solutions

- Native NVMe-oF design

- Design comparison of Bridge vs Native NVMe-oF

- Experimental Results

- Key Takeaways
Disclaimer

This presentation and/or accompanying oral statements by Samsung representatives collectively, the “Presentation”) is intended to provide information concerning the SSD and memory industry and Samsung Electronics Co., Ltd. and certain affiliates (collectively, “Samsung”). While Samsung strives to provide information that is accurate and up-to-date, this Presentation may nonetheless contain inaccuracies or omissions. As a consequence, Samsung does not in any way guarantee the accuracy or completeness of the information provided in this Presentation.

This Presentation may include forward-looking statements, including, but not limited to, statements about any matter that is not a historical fact; statements regarding Samsung’s intentions, beliefs or current expectations concerning, among other things, market prospects, technological developments, growth, strategies, and the industry in which Samsung operates; and statements regarding products or features that are still in development. By their nature, forward-looking statements involve risks and uncertainties, because they relate to events and depend on circumstances that may or may not occur in the future. Samsung cautions you that forward looking statements are not guarantees of future performance and that the actual developments of Samsung, the market, or industry in which Samsung operates may differ materially from those made or suggested by the forward-looking statements in this Presentation. In addition, even if such forward-looking statements are shown to be accurate, those developments may not be indicative of developments in future periods.
Storage Disaggregation

- Fixed ratio of Compute and Storage resources
- Resources scaled and managed together
- Upfront decision of resources for future needs
- Resources underutilized

- Compute and Storage resources separated to create resource pools
- Resources scaled and managed independently
- Flexibility to size resources for different needs
- Improved resource utilization
Disaggregation challenge

- Remote access overheads
  - Additional Interconnect latencies
  - Added Network and Protocol processing

- Disaggregation of Disk Storage (HDD) Common in Data Centers
  - Network overheads are small compared to HDD’s millisecond access latency and low IOPs

- Disaggregation of NVMe Flash SSD is challenging
  - Network and protocol overheads are more pronounced compared to NVMe SSD’s microsecond latency and high IOPS (~MIOPs)
  - Disaggregation of NVMe with iSCSI introduces performance drop

* “Flash storage disaggregation,” EuroSys’16
* “NVMe-over-Fabrics Performance Characterization and the Path to Low-Overhead Flash Disaggregation”, Systor’17
NVMe over Fabric Protocol

- Builds on top of NVMe Protocol and extends it to support various Network Transports (RDMA, FC, TCP)
- Enables scale-out of NVMe devices with DAS like performance and low latency (less than ~10μs)
- Avoids unnecessary protocol translation and offers an End-to-End NVMe model
- Architected from the ground up for current and Next-Generation NVM
Disaggregated storage with NVMe-oF JBOF

- Enables disaggregation of NVMe
- Provide end-to-end NVMe scale out
- Low latency
- Low cost
- High bandwidth
- High density
Existing NVMe-oF JBOF Solutions

- NVMe-oF Target implemented either in Software or Hardware
- Require protocol conversion (NVMe-oF to NVMe)
- Latency offered meets NVMe-oF protocol goal of ~10μs and could vary based on the implementation type
Remote access Latency Impact

- Impact of Network and Protocol much more pronounced with Faster Storage like Z-NAND SSD
- Next Generation memory technologies may have read access latency under a microsecond
- Protocol implementation needs to be improved with optimized design for Faster Storage
Bridge NVMe-oF Design

- Implemented using 2 SoCs:
  - NVMe-oF Bridge (Bridge SoC)
  - NVMe SSD

- Bridge SoC implements the Fabric transport layer and the NVMe-oF to NVMe protocol translation (Conversion logic)

- Delivers performance of NVMe SSD, but protocol translation adds latency overheads
Bridge NVMe-oF – Read Flow

Command submission
1. Host issues NVMe-oF Read Command
2. RoCE receives NVMe-oF Read Command encapsulated in RDMA Send packet
3. RoCE reserves Data Buffers and prepares PRP/SGLs
4a. RoCE updates the SQ entry in Submission Queue
4b. RoCE sends the command info to Conversion logic
5. Conversion logic rings the doorbell

Command processing
6. Controller fetches the command from submission queue
7. Controller processes the Read command
8. Controller fetches the data from NAND Flash
9. Controller writes the data into Data Buffer
10. Controller posts the CQ entry in Completion Queue
11. Controller Posts Interrupt
12. Firmware triggers conversion logic to read data from Data buffer
13. Conversion logic converts Read data as payload for RDMA Write
14. RoCE transmits RDMA Write packet to Host

Command completion
15. Conversion logic converts CQ entry in Completion Queue to RDMA send Packet
16. RoCE Transmits NVMe completion encapsulated in RDMA Send Packet
17. Conversion logic updates the Completion Queue head doorbell
Bridge NVMe-oF – Overhead

- PCIe memory write for doorbell ring (step 5)
- PCIe memory read for command fetch (step 6)
- PCIe memory write for data buffer update (step 9)
- PCIe memory write for Completion posting (step 10)
- PCIe memory write for MSIX posting (step 11)
- PCIe memory write for doorbell ring (step 17)

Each and every command processing incurs the above PCIe overheads!!!
Native NVMe-oF Design

- Integrates the Fabric transport layer and the Flash controller on a single SoC
- Natively supports NVMe-oF protocol with an optimized design
- No PCIe transactions required for Data transfer
- Eliminates NVMe-oF to NVMe Conversion
- Ultra Low Latency
- Targeted for faster flash like Z-NAND
Native NVMe-oF – Read Flow

**Command submission**
1. Host issues NVMe-oF Read Command
2. RoCE receives NVMe-oF Read Command encapsulated in RDMA send Command
3. RoCE posts SQ entry in Submission Queue
4. RoCE notifies the controller about the arrival of new SQ entry

**Command Processing**
5. Controller Fetches the command from submission queue
6. Controller processes the Read command
7. Controller Fetches data from Flash
8. Controller Sends data to RoCE
9. RoCE encapsulates Read data as payload for RDMA Write and sends to Host

**Command completion**
10. Controller posts the CQ entry to RoCE
11. RoCE encapsulates CQ entry into a RDMA Send packet and sends to Host
Bridge vs Native - Results and Analysis

- Additional cost, power, footprint of the Bridge SoC
- Protocol Conversion
- Bridge SoC may become bottleneck for faster SSDs in the future

- Low cost, power, footprint due to single SoC
- No Protocol conversion
- DAS like performance and Latency

Bridge NVMe-oF: 20% more latency than DAS
Native NVMe-oF: Equivalent to DAS

Latency
Bridge NVMe-oF Z-NAND SSD  Native NVMe-oF Z-NAND SSD  Directly Attached Z-NAND SSD
Key Takeaways

- NVMe-oF protocol enables disaggregation of NVMe devices without any additional overheads.
- Existing solutions with a NVMe-oF bridge design cannot harness the low latency benefits of faster storage, as it adds overheads.
- Native NVMe-oF design removes the latency overheads of Bridge NVMe-oF design.
- A Native NVMe-oF SSD can be a key enabler for building Next-Gen NVMe-oF JBOF type of solutions.
References

- https://nvmexpress.org/
Acknowledgements

- Venkataratnam Nimmagadda
- Navia Vishnu Rugmini
- Sandeep Kumar Ananthapalli

THANK YOU!