



Flash Memory Summit

Quick Prototyping of ZNS Protocol Using a Conventional SSD

Ganesh Guddanti

- ZNS (Zoned Namespace): Genesis
- A conventional SSD
- Prototyping ZNS on a conventional SSD
- Benefits of this quick prototyping
 - For an SSD Vendor
 - For an OEM



ZNS (Zoned Namespace): Genesis

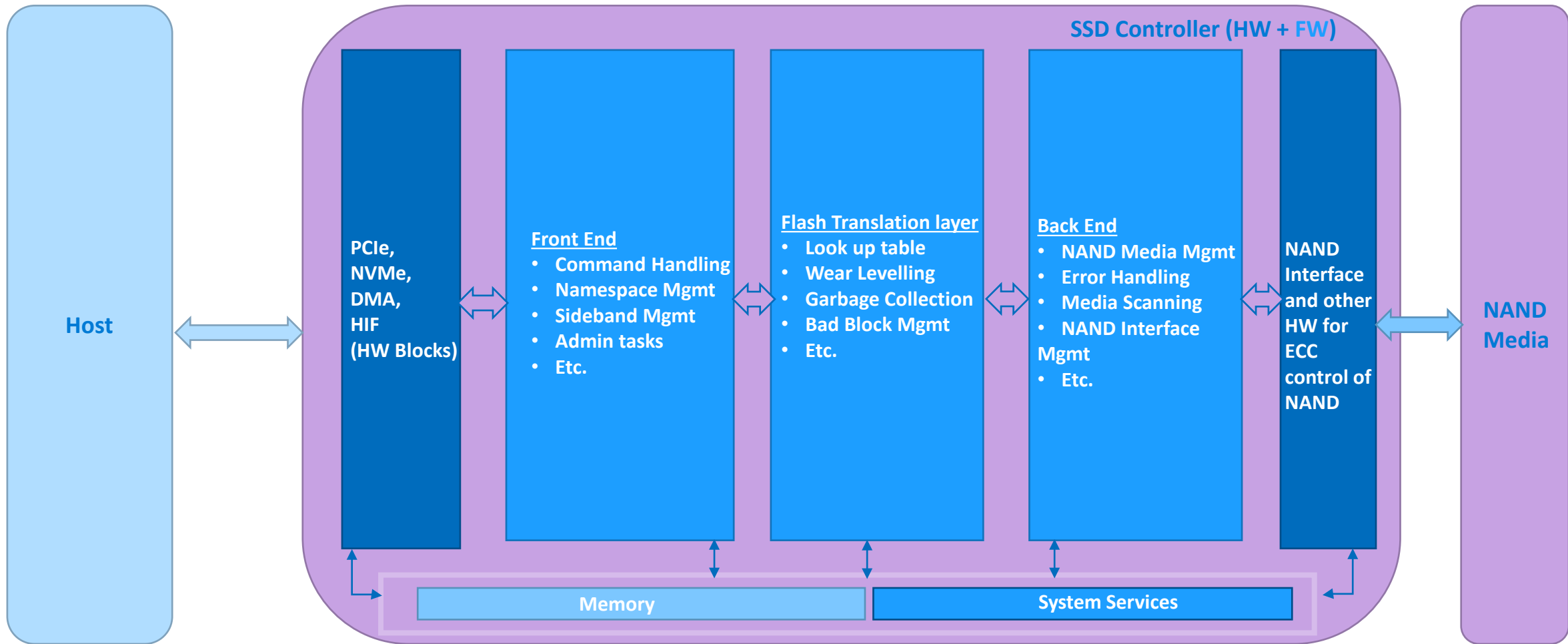
- SSDs are mostly abstracted to host software as block storage devices
 - When underlying persistent media is NAND –complex controller FW and downsides of Write Amplification (WA), latency outliers and overprovisioning (OP)
- Alternatives to reduce WA, latency outliers, OP
 - Open Channel SSDs
 - Hints to the host (e.g., NVMe : Directives, NVM Sets, etc.)
 - But both these approaches have their downsides too!
- **ZNS is a co-operative flash management approach to solve these issues** (inspired from SMR HDDs!)

ZNS deployment scenarios and target market segments

- Where reduced Total Cost of Ownership (TCO) matters
 - File hosting service, Personal Cloud Storage, etc
- When I/O predictability is needed
 - Hyper scalars/Data centers, with appropriate configuration of ZNS
- When SMR Eco-system can be leveraged
 - Replace SMR HDDs with ZNS SSDs
 - Reduced TCO
- Challenge
 - Host SW eco system adoption

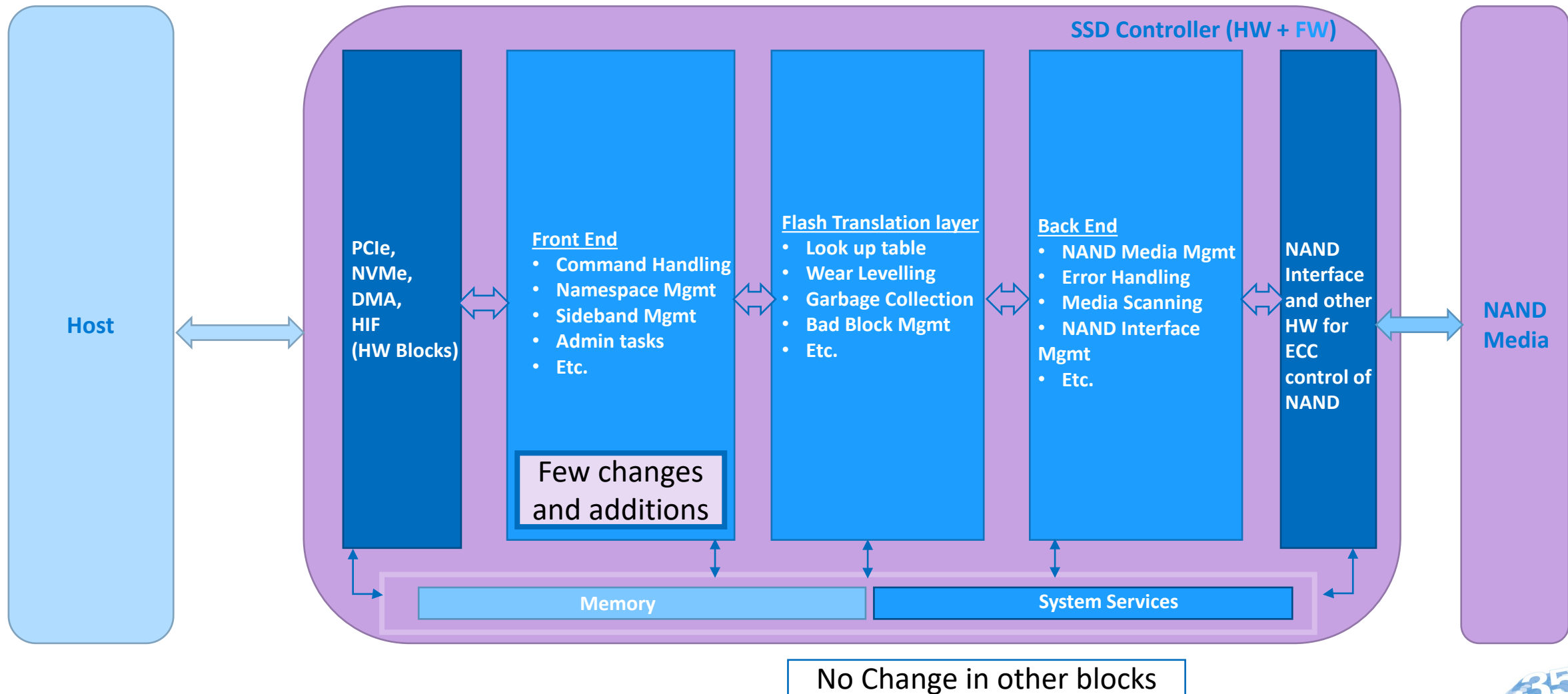


Conventional SSD architecture





Prototyping ZNS on a conventional SSD



Prototyping ZNS on a conventional SSD

- At a high level, changes needed in the Front End sub-system:
 - Add support for ZNS Command set
 - Write commands – validate (WP) and proceed as before
 - Read commands – mostly no change if Read Across Zone Boundaries bit is '1'.
 - Zone Management Send
 - Zone Management Receive
 - Changes related to support Zone Append command
 - Addition of Command Specific Status Values, Zoned Namespace Command Set

(Continued on the next page)

Prototyping ZNS on a conventional SSD

- At a high level, changes needed in the Front End sub-system (continued):
 - Include support for “TP 4056d Namespace Types”
 - Identify Namespace changes
 - Minor Namespace management changes

Prototyping ZNS on a conventional SSD

- Initially support bare minimal features of ZNS
 - Read across zone boundaries bit '1'
 - Only one LBA Format Extension
 - Zone Size equal to Zone Capacity
 - Minimal Protection Information settings
 - No Reset Zone Recommended or Finish Zone Recommended support
- Add-on additional features later
 - Zone Descriptor Extension
 - Changed Zone List log page
 - Etc.

Prototyping ZNS on a conventional SSD

- Host sees the drive as a ZNS SSD but internally it is a conventional SSD
 - Flash translation remains same – LUT at a granularity as before
 - Garbage Collection, Wear levelling
 - WA is still high as before
 - IO predictability matches conventional SSD
- All the above limitations of the prototype are underplayed by the benefits this approach brings...



Benefits for SSD Vendor

- Enable partners in end-to-end SSD product development **very early!**
 - De-risks schedule! → **Time to market (TTM) is in control!**
 - Quickly enable infrastructure and internal partners to handle intricacies of ZNS SSD with confidence!
 - Quickly check if host interface stack, software, tests, etc. are in place
 - E.g., is host able to recognize the drive as a ZNS SSD?
- Otherwise, ZNS FW is new, and its verification also is new!
 - Integration issues or bugs are in multiple moving targets
 - Quick Prototype → verification starts early and is more thorough. This helps the actual product development → **Less TTM!**



Benefits for OEM

- Reduce unknowns and intricate issues on the host interface/ software stack by validating the host applications scenarios early on with this prototype
 - Better control on TTM for the OEM/Customer who needs to deploy ZNS SSDs eventually in their products/services
 - Negotiate certain features for the better deployment and cost-efficiency
- Limitations
 - No early insights into tuning power and performance of the whole application deployment

References

- <https://nvmexpress.org/wp-content/uploads/NVM-Zoned-Namespace-Command-Set-Specification-1.1b-2022.01.05-Ratified.pdf>