Architected for Performance

NVMe Over Fabrics
Real World Use Cases and Applications

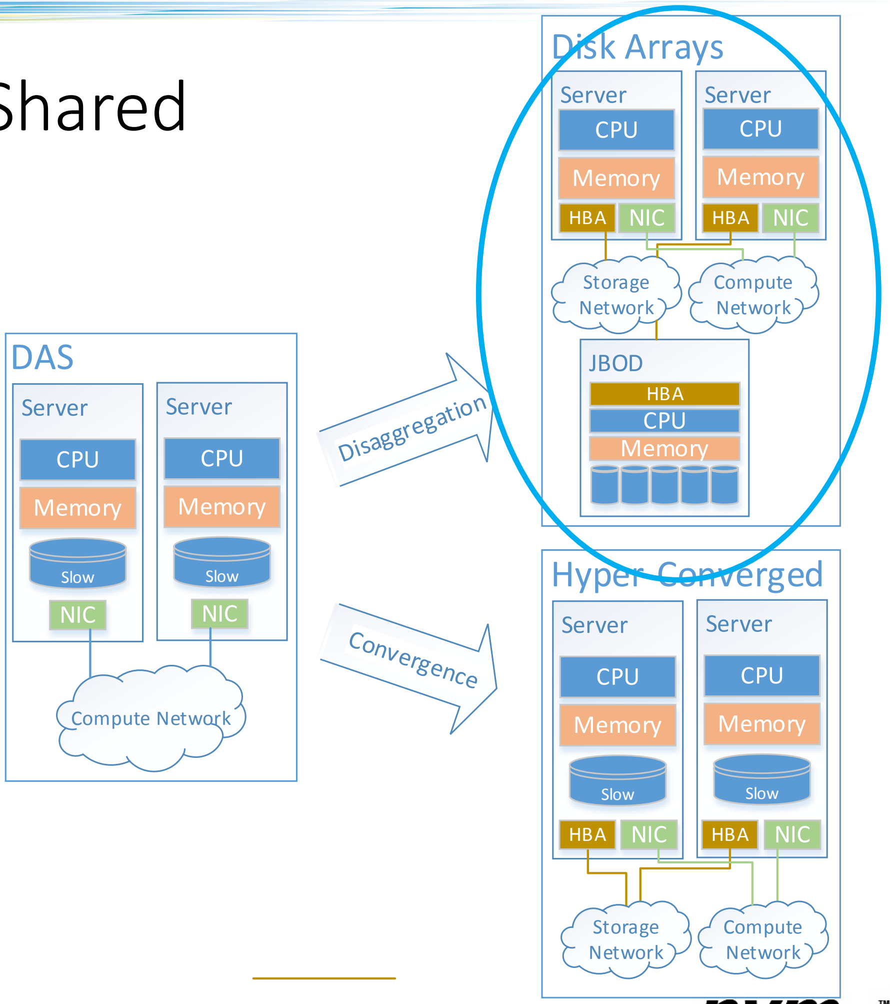
August 11, 2015

Idan Burstein

Storage Architect, Mellanox Technologies

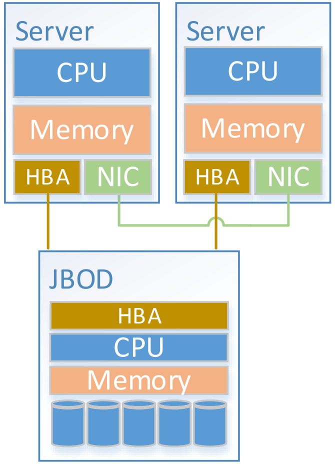
History – Directly Attached to Shared

- Major advantages for sharing
  - High availability
  - Utilization and provisioning
    - Deduplication, compression
    - Thin provisioning
  - Cost
- Historically disks were slow
  - Storage software stack was built for hard disks, very slow relatively to memory
  - Storage network was fast relative to disks, very slow relatively to memory



Evolution of Disk Arrays

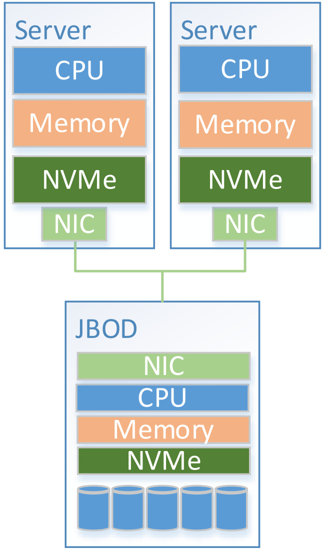
Disk SAN



- Memory was used for caching
- Slow disks

- Storage network has become too slow
- Flash prices dropped
- NVMe
- Demand for data intensive latency sensitive tasks

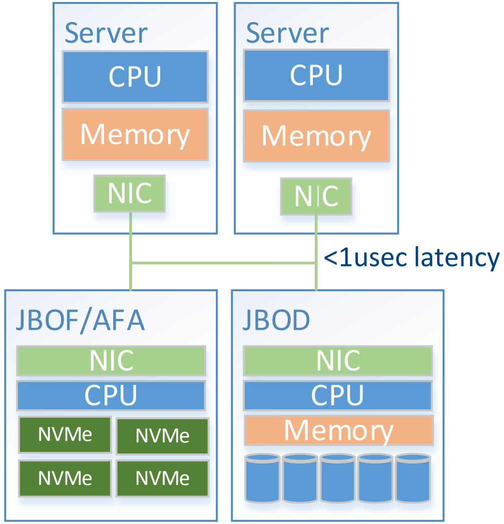
Disk SAN with Local NVMe



- NVMe devices used for caching
- Convergence to fast RDMA fabrics

- Demand for consistent performance from array
- HDD-like Flash disaggregation

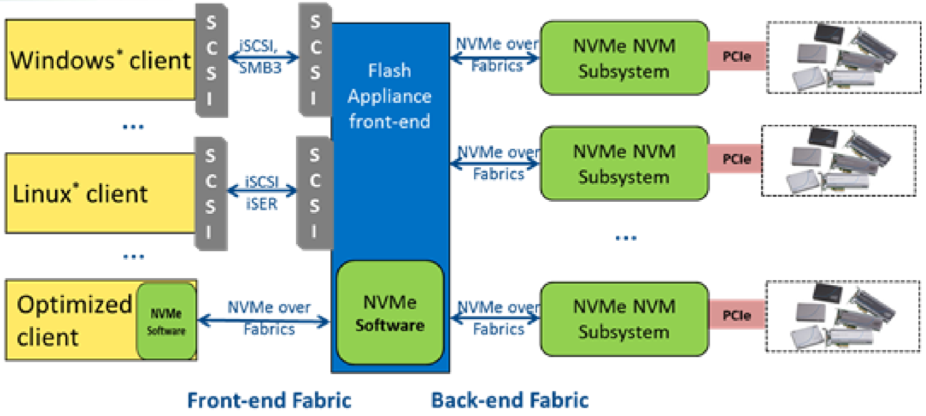
Disk and Flash SAN Local Memory-Like NVM



- All flash arrays used for fast storage (caching)
- JBOD are used for cold storage

Flash Array Use Case

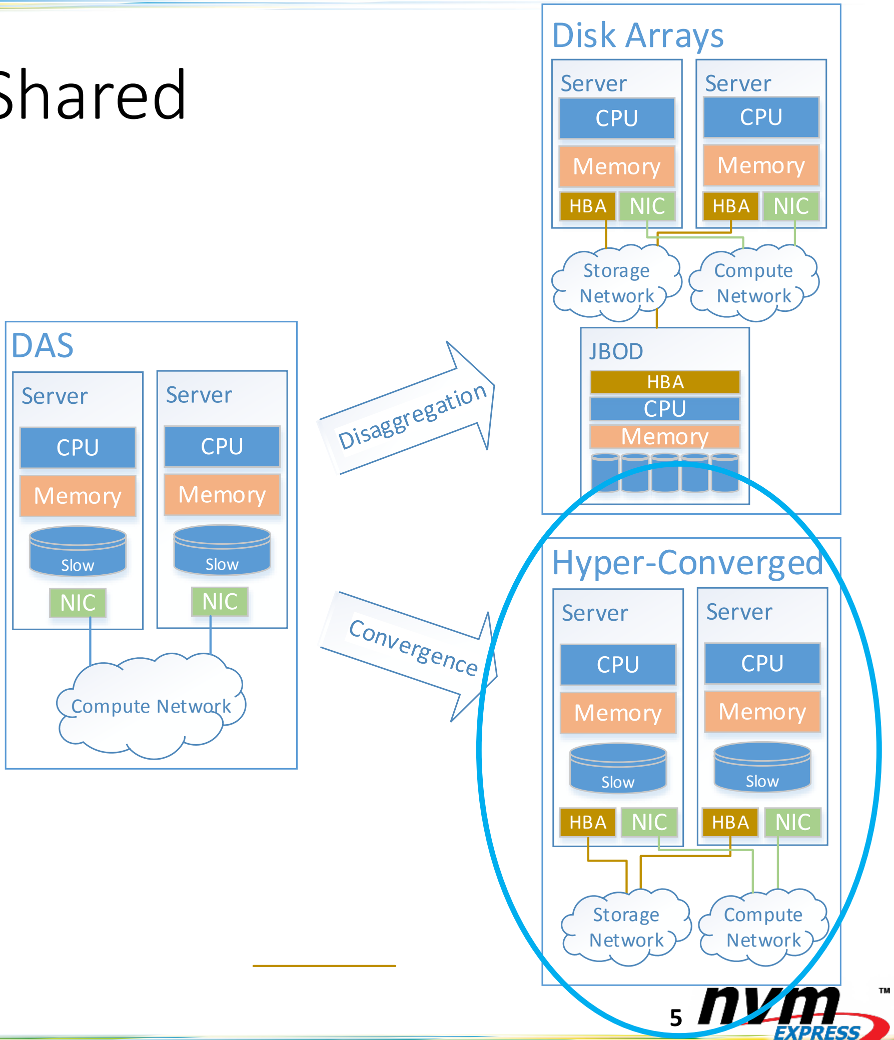
- **Benefits of NVMe over Fabrics for disaggregation**
  - **Scale of RDMA**
    - Scaling out with RDMA networks, beyond PCIe scaling limitations
  - **Performance of RDMA**
    - Low latency, high bandwidth, parallel interface, locally attached like performance for accessing the devices
  - **Minimal CPU utilization at the subsystem and the host**
    - Lockless parallel design from client to disk
    - Reduction of protocol translation
    - Reduction of the CPU overhead of large data transfers through RDMA
  - **Convergence**
    - Compute and storage in the same network



- **Why is it good for backend?**
  - Scaling number of disks independent of the compute
  - Low latency, high bandwidth shared access
    - For example to enable HA and deduplication algorithms
  - Lower CPU%
    - Frontend servers - more CPU% for smart storage algorithms
    - Subsystem servers - enable low cost solutions
- **Why is it good for frontend?**
  - Lower CPU%
    - Frontend servers - More CPU% for smart storage algorithms
    - Client servers – Data is moved without CPU → more compute resources → $
  - Locally attached like performance
  - Disaggregation doesn't require software changes
  - Media is easily managed and shared

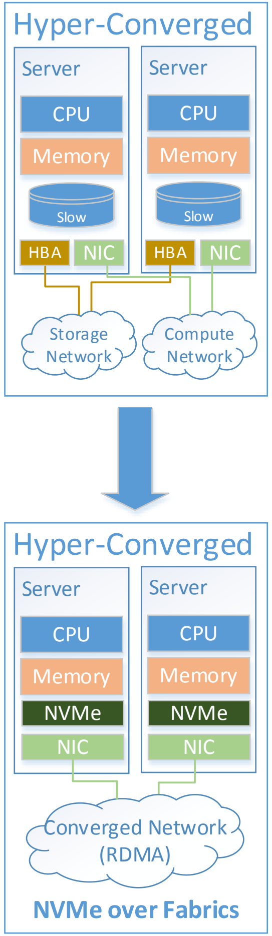
History – Directly Attached to Shared

- Advantages for sharing
  - Management and failover
    - Thin provisioning
    - High availability
    - Utilization
  - Deduplication, compression
- Storage network was fast relative to disks, very slow relatively to memory
- Storage software stack was built for hard disks



Hyper-Converged Use Case

- Storage is distributed across the compute nodes and shared among the nodes
- Storage management and provisioning is software defined and distributed
- Benefits of NVMe over Fabrics
  - The most important: major reduction in CPU utilization while sharing devices, the compute nodes are not disrupted by storage → more compute resources for applications
  - Locally attached like performance
  - Scaling of RDMA network
  - Converged network
    - No protocol translation and no additional dedicated hardware

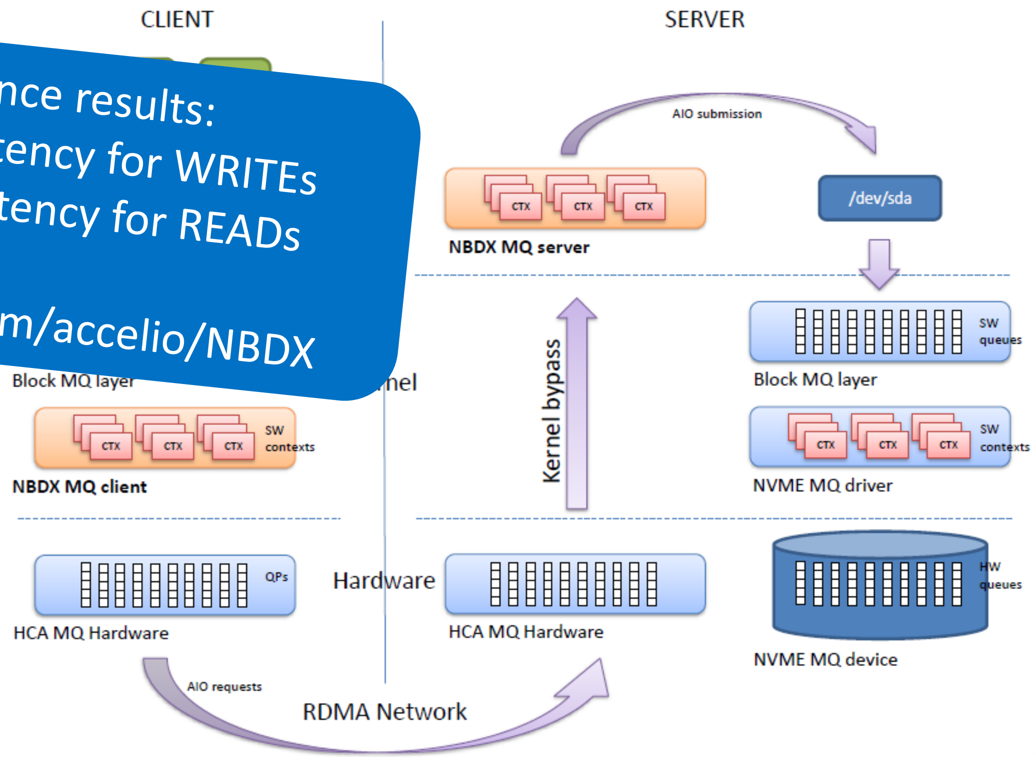


NBDx – NVMe over Fabrics POC

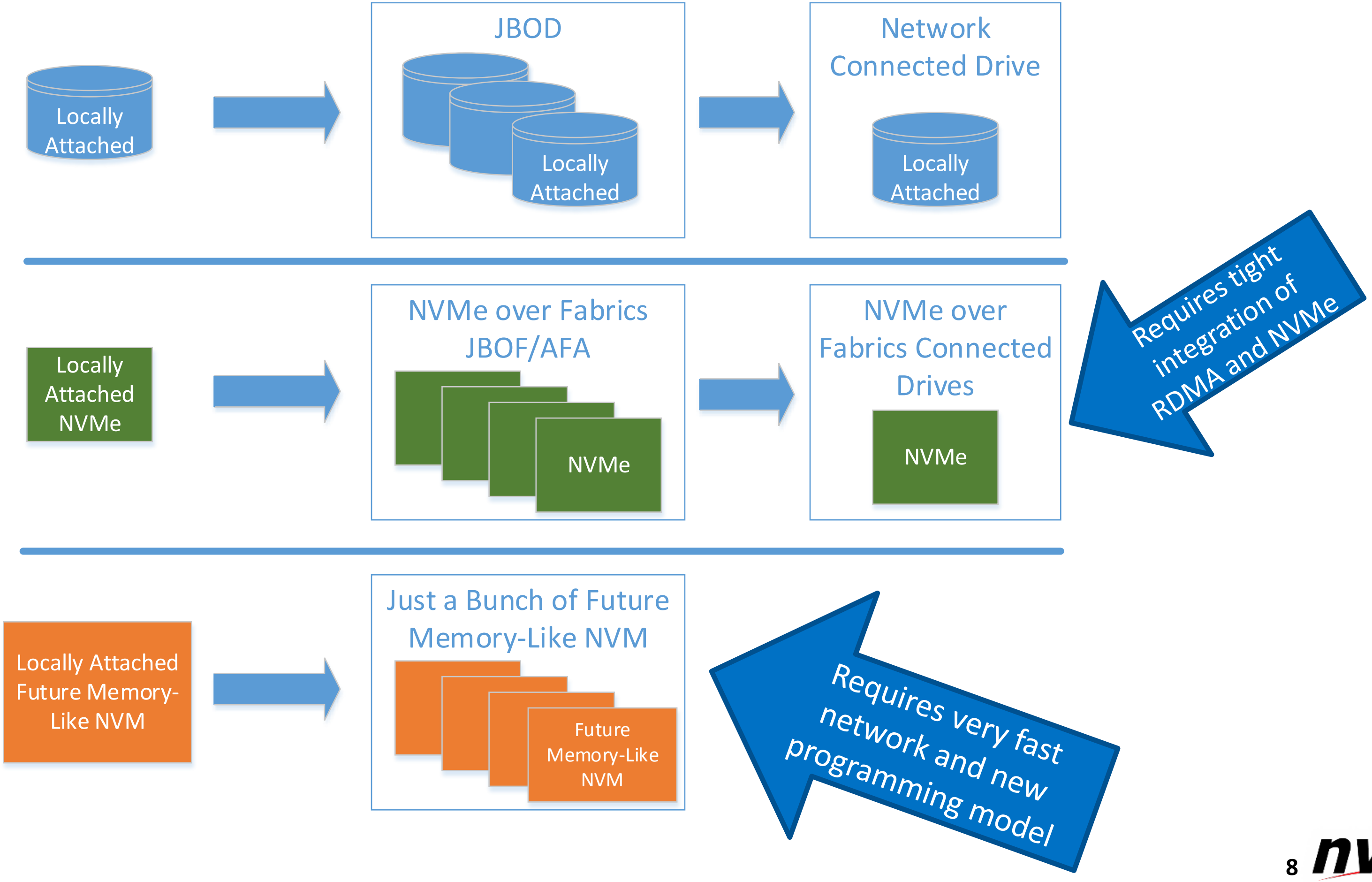
- Open source
- RDMA enabled
- Multi-Queued
  - From submission t
    all on same core, i
    target
- End-to-end lock free
- No protocol translations
- Userspace only demo – FIO
  - Engine that opens QPs, CQs and speaks NBDX

Performance results:
2usec added latency for WRITEs
5usec added latency for READs

https://github.com/accelio/NBDX



Future



Locally Attached
JBOD
Locally Attached
Network Connected Drive
Locally Attached
Requires tight integration of RDMA and NVMe
Locally Attached NVMe
NVMe over Fabrics JBOF/AFA
NVMe
NVMe over Fabrics Connected Drives
NVMe
Locally Attached Future Memory-Like NVM
Just a Bunch of Future Memory-Like NVM
Future Memory-Like NVM
Requires very fast network and new programming model

Architected for Performance