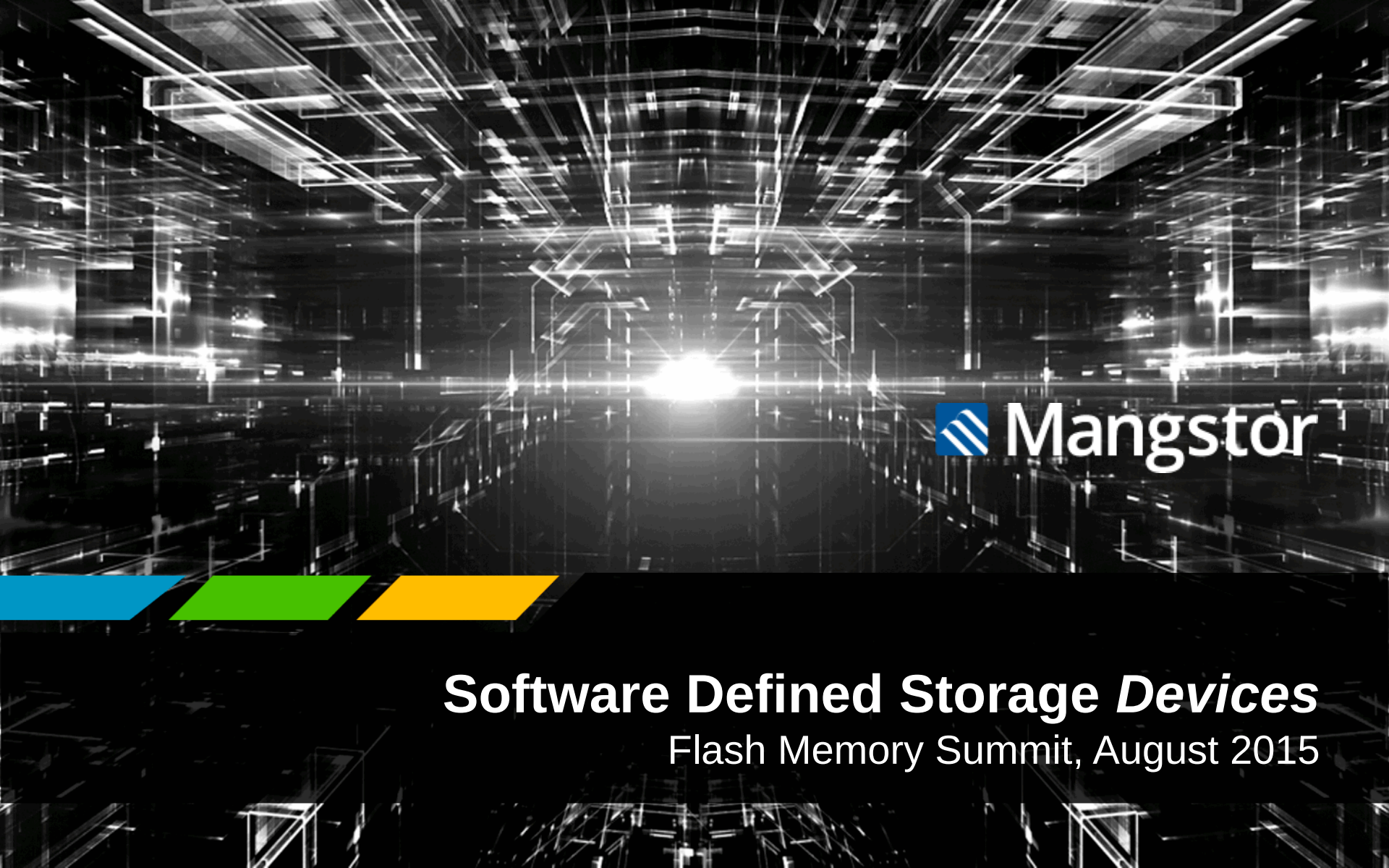
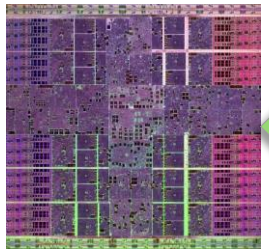


Software Defined Storage *Devices*

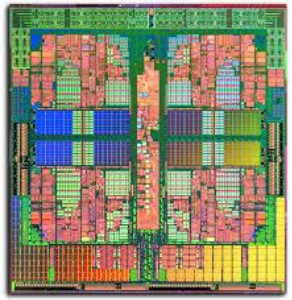
Flash Memory Summit, August 2015

Circa 2003

Application



Offload GFX computation



CPU

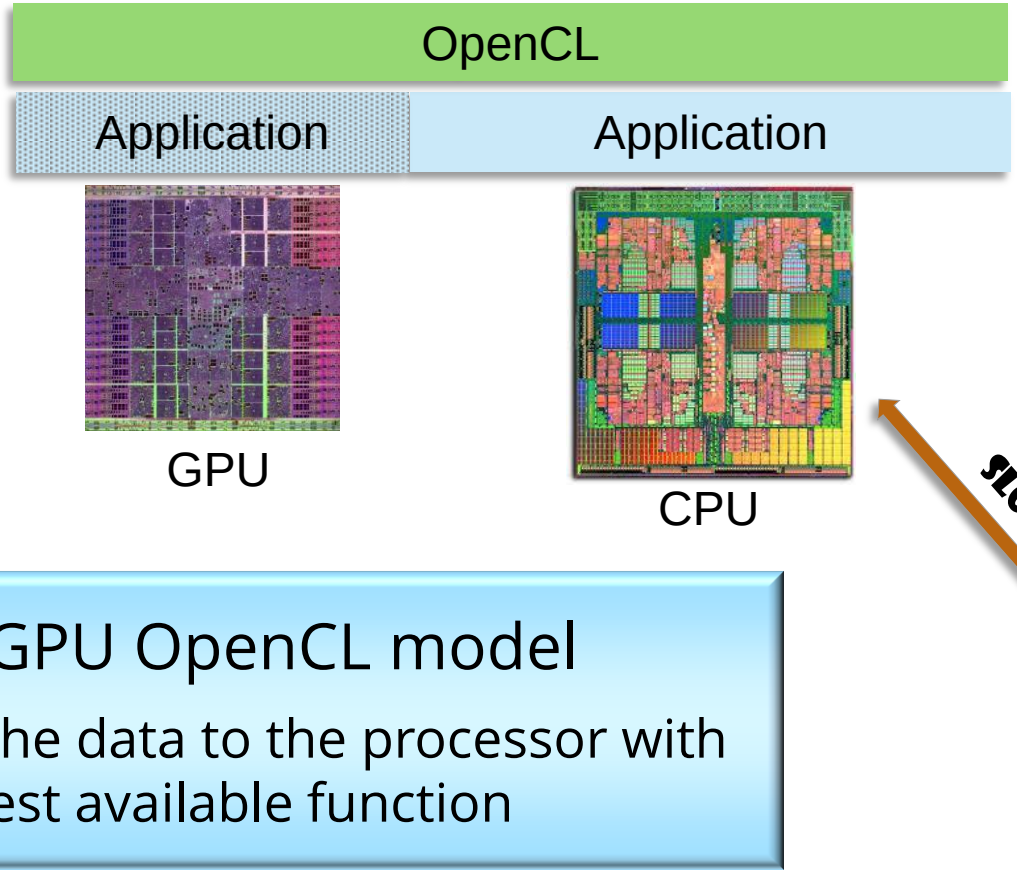
Interface: GPU
OpenGL
Direct-X

SLOOOOOW

HDD

GPUs get *very fast*

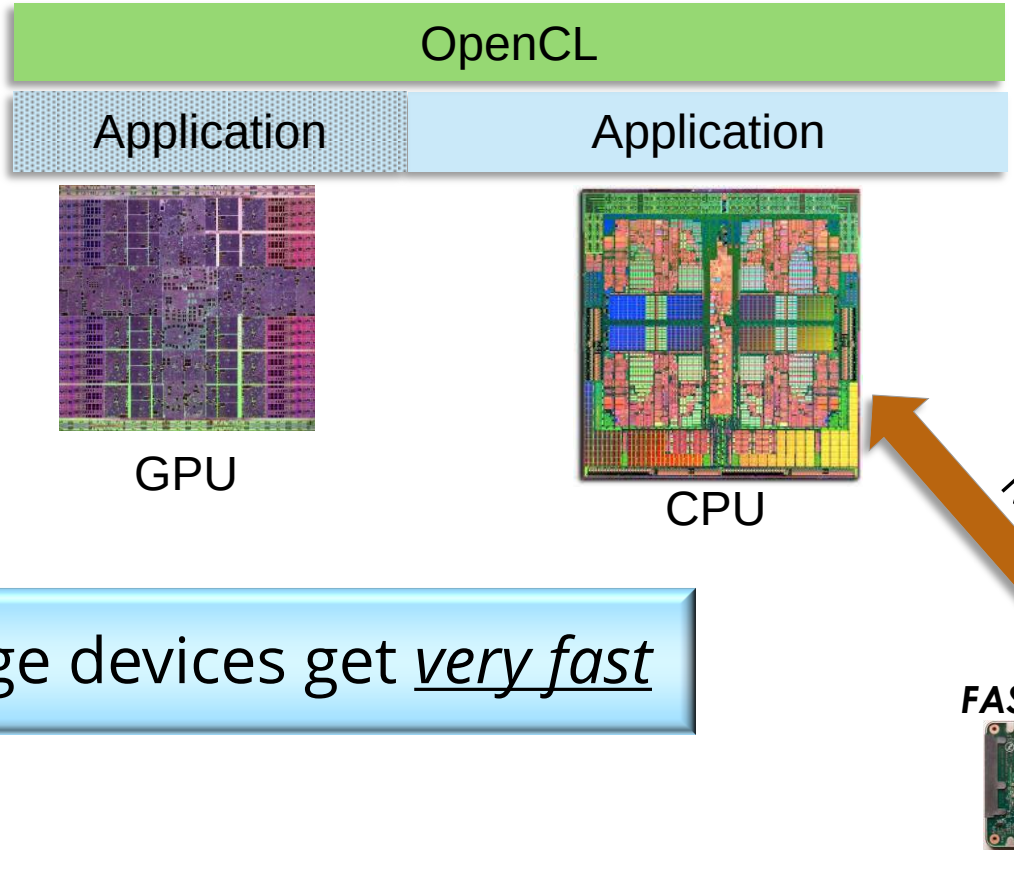
Circa 2008



CPU & GPU OpenCL model

- Ship the data to the processor with the best available function

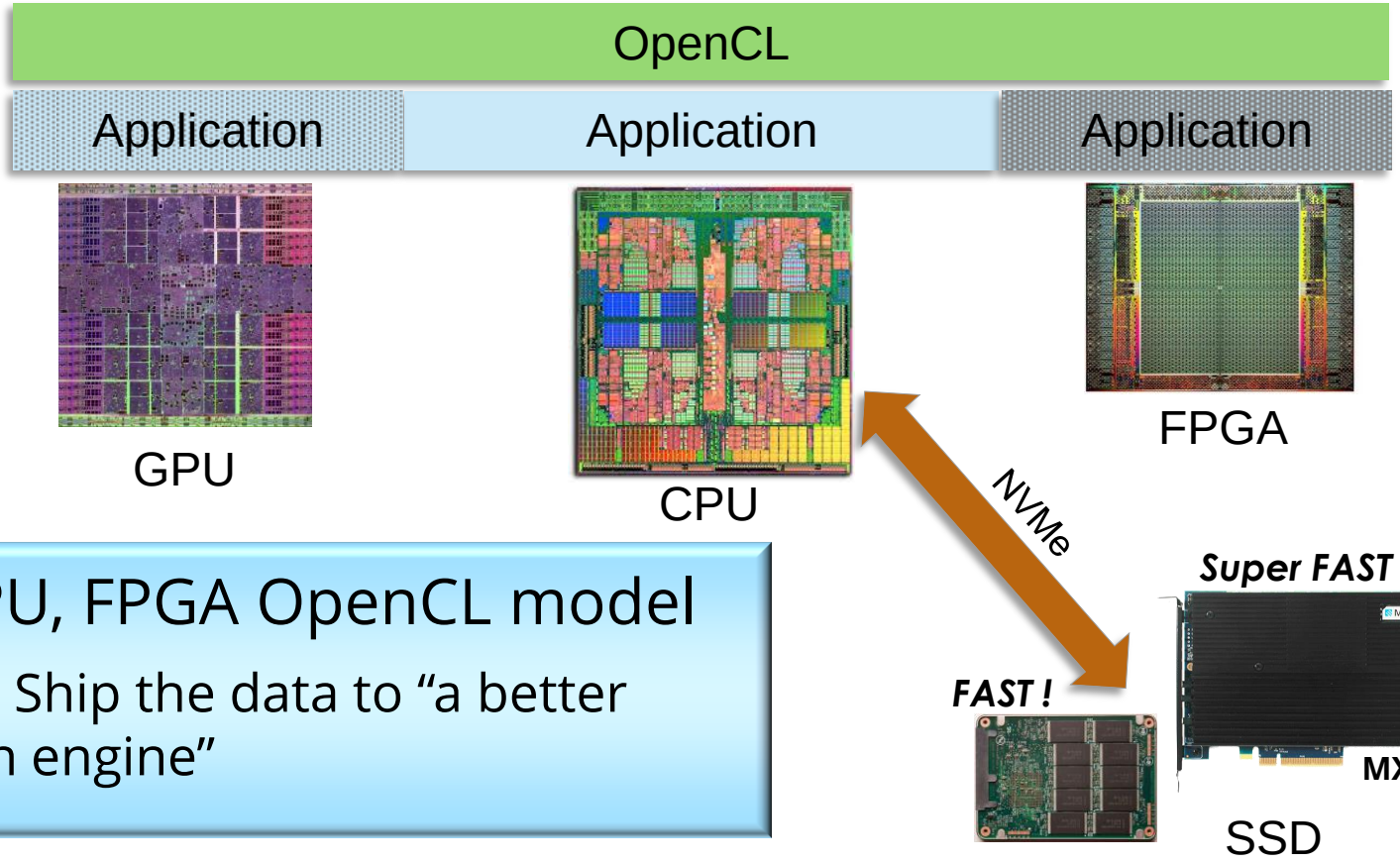
2015



Storage devices get *very fast*

Mangstor

2015

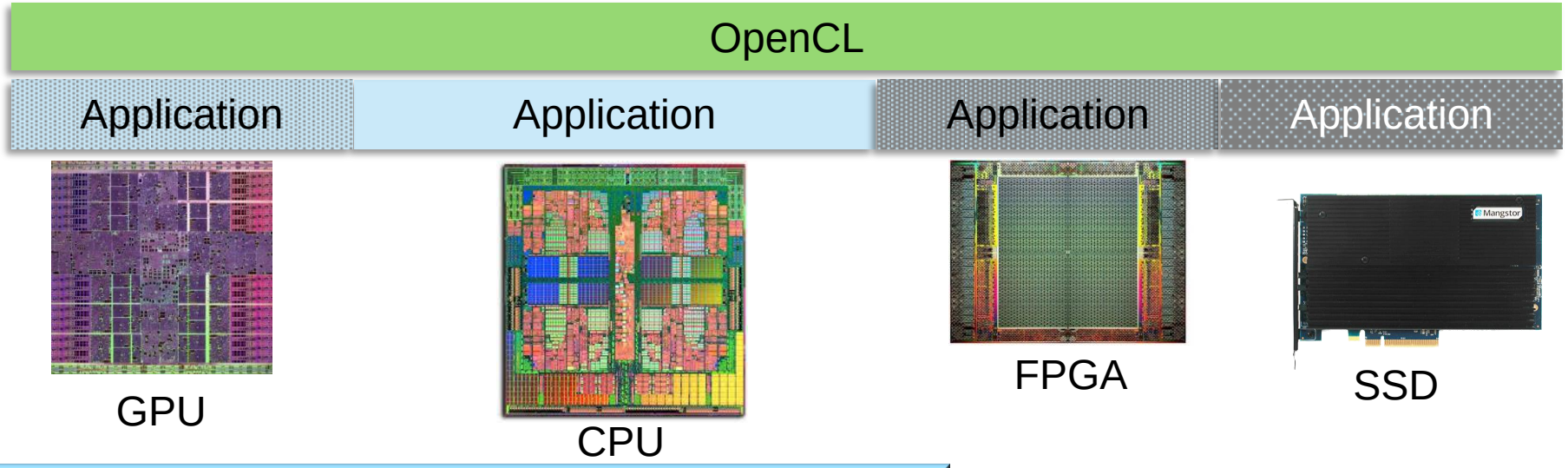


CPU, GPU, FPGA OpenCL model

- (FPGAs) Ship the data to “a better function engine”

Mangstor

We want to do this…



CPU, GPU, FPGA, SSD OpenCL model

- (SSD)
  - Embed the function with the data
  - Ship the function to the data

How do we get there?

- How did GPUs do it?
  1. Do some things far better than the CPU
  2. Standardize access to the functions
     - OpenGL has been tracking device improvements since 1992
  3. Define compelling use cases

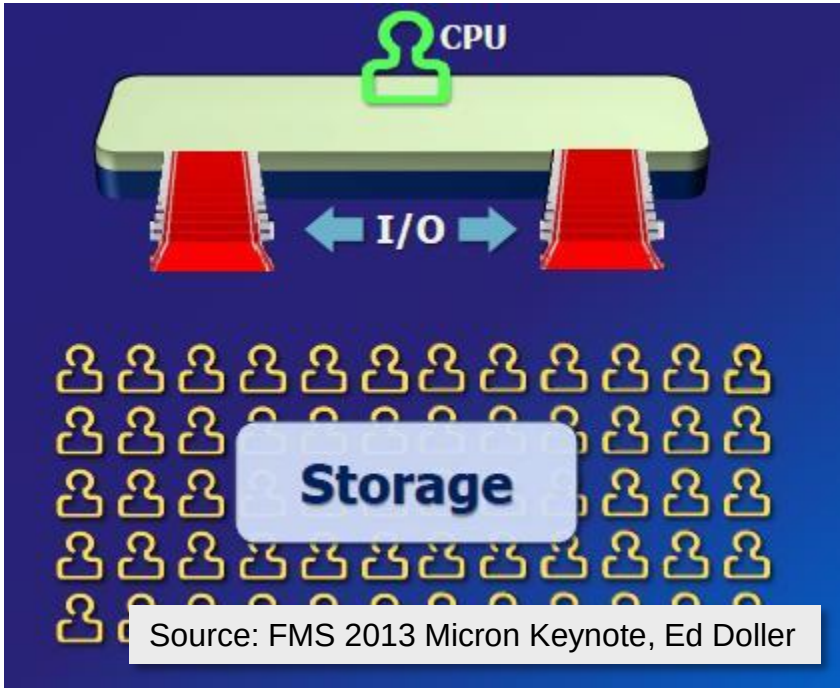
What do we do better than the CPU?

Answer: **Store massive amounts of data**…

So… Anytime it makes sense to

"bring the function to the data"

- Database
  - Search / Compare, Merge, Update, …
- Video
  - Encode, Transcode
- Scientific
  - "Simple" data transformations



Source: FMS 2013 Micron Keynote, Ed Doller

What about Software Defined Storage?

- "SDS" today generally implies an x86 server with software services running on it
  - Great start for offloading fixed functions, e.g.
    - RAID, Encryption, Compression, Deduplication, Tiering, Snapshot, Replication, HA, …
  - Need standard APIs for using an internal, or *user-generated*, function (e.g. an OpenCL "Kernel")
  - Attached via legacy iSCSI or FC… ***SLOOOOW***

NVMe over Fabric
Software Defined Storage

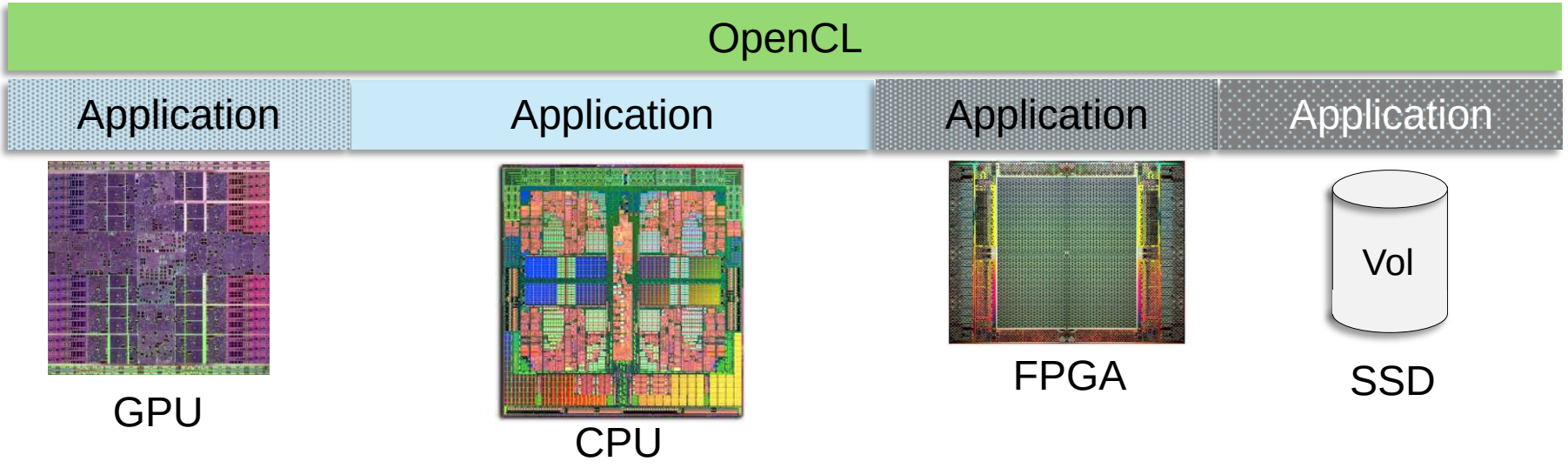
Advantages of DAS:
- Low latency
- High BW

Advantages of SAN:
- Isolate storage from server

… Easier to service
… capacity can be re-allocated

The best of both storage approaches… plus:
- Non-Proprietary architecture
- Standard Ethernet switch fabric
- Open industry management tools
- Completely compatible with legacy applications
  - Volumes appear as a local NVMe device

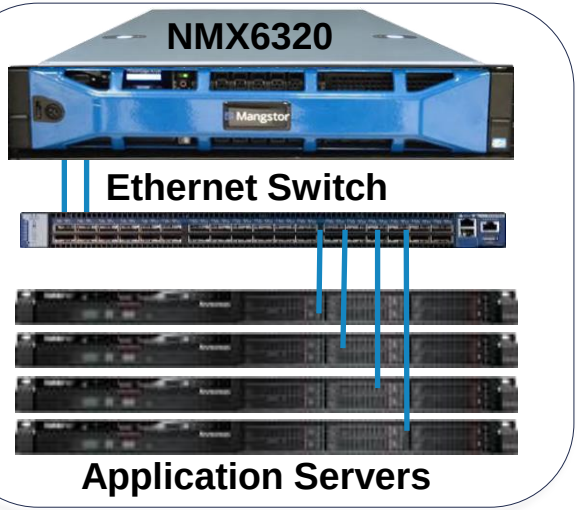
With NVMe over Fabric, we can do this…



OpenCL using CPU, GPU, FPGA, *and* SSDs

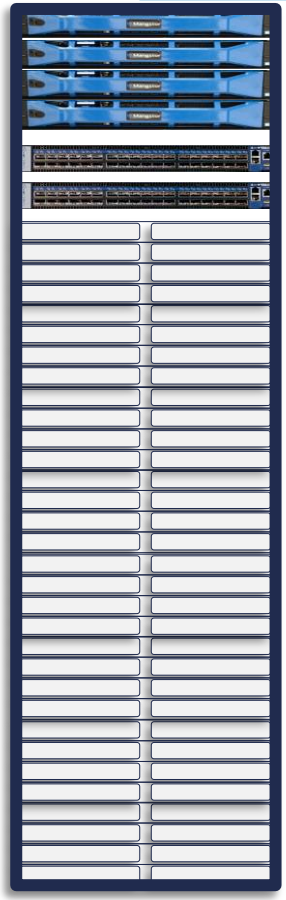
NMX series Scalable Storage Solutions

The Mangstor NMX series of storage arrays is the perfect high performance Server-SAN storage for clustered applications...



Small workgroup clusters

Dense full-rack clusters



Storage arrays

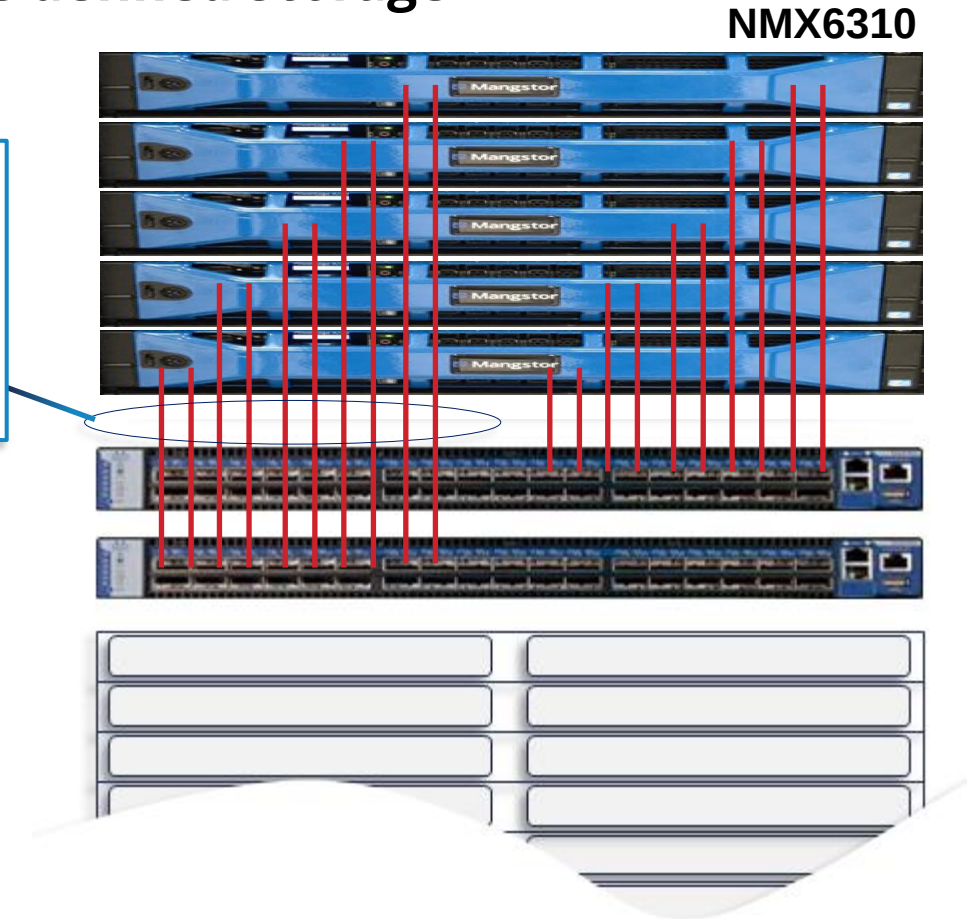
Redundant Switches

Application Servers

Full-Rack Cluster

NVMe over Fabric, Software defined Storage

- Scalable capacity & BW shared by multiple clients
- Over 500 Gbps BW *
  - With Failover / redundancy



* (5 Storage arrays, using 56Gb links)

Why not Software Defined Storage *Devices*?

- With NVMe Storage Devices offload *media control*
  - FTL mapping, Wear leveling, Garbage collection, ECC, …

- Mangstor has developed an SSD with a 100 core processor onboard
  - NVMe host interface and Flash control completely done in Firmware on these cores
  - Other functions to be provided by these cores

- Others are developing SSDs with many more cores than in previous generations

Industry Call to Action

- Continue NVMe standard development to take advantage of compute functions in the SSD.
- Begin deploying OpenCL compatible reference code and whitepapers using in-storage processing
  - Available first in SDS arrays
  - Migrating into SDS devices

Intelligent storage products

Paul Prince, CTO
peprince@Mangstor.com
August 2015