A Comparison of In-Storage Processing Architectures and Technologies

Jérôme Gaysse
Senior Market and Technology Analyst
Funny time for architects

- The evolution of server technologies

- 6 pieces
- 30 pieces
- 200 pieces
Agenda

- Part 1: Concept and market adoption
- Part 2: Future applications
- Part 3: Architecture roadmap
- Part 4: Technology roadmap
Part 1

- Concept and market adoption
The data movement problem

- For data processing
  - From storage to RAM
  - From RAM to CPU
Making data and processing closer

Computational storage       Smart SSD

In-situ processing   In-storage processing

- Reduce « distance » between storage and compute
  - Lower latency => better performance
  - Less energy for data transfer => lower power consumption
Data to CPU…?

- NVDIMM-based
  - Not the focus of this talk

Storage in-memory
Or CPU to data?

- The choice of implementation today
1GB data movement example

CPU 150W / DDR4 6W-25.6GB/s / NVMe 15W 3GB/s

1400pJ/bit 330ms

4x faster

10x power efficient

SSD 15W / 32 ONFI Channels 400GB/s

150pJ/bit 80ms
Theory of operation

- The NVMe interface provides all the requirements
  - Standard interface
  - Configurability: namespaces, vendor specific commands
  - Low latency
Various implementations

SSD Controller
With computing

NVM
NVM
NVM

SSD Controller
With computing

M.2
M.2

SSD Controller

NVM
NVM

SSD Controller

NVM
NVM

Coprocessor

NVM
NVM

NVM

M.2
M.2

Peer2peer

Accelerator
Applications

- Compression
- Encryption
- Search
- Keay-Value store
- ...

...
Market: more and more players

- Products
  - NGD Systems
  - Samsung
  - NVXL
  - StarBlaze
  - ScaleFlux
  - EIDETICOM

- Technology providers
  - Xilinx
  - ARM
  - Marvell
Market adoption, standard needed

- SNIA SDC agenda
  - Birds of a Feather: Computational Storage
  - Key Value Storage Standardization Progress
  - FPGA-Based ZLIB/GZIP Compression Engine as an NVMe Namespace
  - Deployment of In-Storage Compute with NVMe Storage at Scale and Capacity

- SNIA workgroup
  - Define technical interface/standard
  - Need a universal name!
Part 2

Future applications
The real value

- For data analytics and deep learning
  - Need of high performance and low latency
  - Power budget limitation for edge computing
  - Huge improvement to come for hardware accelerator

=> data movement problem will increase on standard architectures
Deep learning

- Inference / training
- Training problem
  - Long time process
  - Expensive resources

} business impact
Deep learning training system

- Based on GPUs

![Diagram of a deep learning training system with GPUs and SSDs connected via CPUs.](image-url)
Performance analysis

- ResNet50
  - 50 layers
  - 25M parameters
  - 3.9GMACs/image through the NN

- Many benchmarks available
  - HPE, DELL, NVIDIA
Performance model

- Basic computing model based on:
  - Hardware parameters: FLOPS, memory bandwidth, compute nodes…
  - Configuration parameters: batch size, NN weights number and resolution…
- Model used for new architecture performance estimation
Performance analysis

- Throughput: 400 images/s / GPU
  - FP32, 100MB model size
- Data set read:
  - 100kB images at 400FPS : 40MB/s

=> No IO storage bottleneck (today…)
Performance at system level

- 3U server, 3000W
- 3200 FPS
- 320 MB/s
Deepl learning processing improvements

- Lower resolution (FP32, FP16, INT8, INT4)
- Pruning (neuron connexion optimization)

Less computing requirement
Less memory bandwidth requirement

=> Training throughput to increase by 25
(10kFPS/GPU)
Performance at system level

- 3U server, 3000W
- 80kFPS, 8GB/s read
The latency problem

- It is not a problem of bandwidth, but a problem of latency!
- Deep learning training is based on:
  - Random access
  - Low QD IO
- Leading to the use of an additional AllFlash Array: 2U, 1000W
Computational storage for DL training

- Hardware options
  - FPGA, ASIC/CPU, Manycore, AI chip

- Criteria
  - Power, Performance, flexibility
Target System Configuration

- 2U server, 24 U.2, 1000W

```
     C     C     C     C
     S     S     S     S
     C     C     C     C
     S     S     S     S
     C     C     C     C
     S     S     S     S

  CPU  --  CPU
```
CS vs GPU

- **GPU**
  - 4000W, 80kFPS, 5U
  - 20 FPS/W
  - 16 kFPS/U

- **Computational storage**
  - 1000W, 24kFPS, 2U
  - 24 FPS/W
  - 12k FPS/U

What about cost and TCO?

What about scalability?
FPGA architecture

- Need NN accelerator
- HBM compatibility
- Power budget ok
SoC architecture

- Software flexibility
- NN IP mandatory
- SoC with HBM?
- Power budget ok
Manycore co-processor architecture

- Software flexibility
- Power budget ok
AI co-processor architecture

- Power budget ok
- Al optimized

Diagram:

- FPGA
- NVMe
- ONFI
- NVM
- AI chip
Part 3

- Architecture roadmap
Room from improvement

- Current computational storage is just the first page of a new chapter of computing architecture
- Computational storage is nice…
- But…
  - Data not shared between devices
  - No cache coherency
New interconnects

- GenZ
- CCIX
- OpenCAPI

- How to apply it to computational storage?
System topology examples

ARM Tech Con 2017
CCIX as a new interface

- With Cache coherency
CCIX as an interconnect

- With Cache coherency
CCIX as an interconnect

- With Cache coherency
GenZ as a new interface

- For disaggregated computational storage
GenZ as an interconnect

- High throughput data sharing
GenZ as an interconnect

- High throughput data sharing
GenZ as an interconnect

- Sharing dataset between systems
Part 4

- Technology roadmap
Processing and storage on the same die

- Processor in memory
- Memory in processor
Processor in memory

- Higher level of integration
- Keeping the same NVMe interface

Diagram:
- CPU
- SSD Controller
- ONFI
- NVM
- CPU
- NVM
Memory in processor

- Less storage capacity
- Better computing efficiency
Conclusion

- Demand for computational storage and existing solutions
- Standard needed for system integration and validate the market adoption
- Standard must be open enough to support the technology roadmap
Thanks!

Any questions?

jerome.gaysse@silinnov-consulting.com