Computational Storage

Jamon Bowen
Product Planning & Storage Segment Director
Bottlenecks Remain for Data Intensive Applications

Processor-centric architecture

- Excessive data transfers
- High latency
- Limited BW

CPU not optimized for these tasks
Emergence of Computational Storage as the Solution

Computational storage architecture

- More available CPU cycles
- Reduces required bandwidth
- Reduces latency
- Compute acceleration close to storage

Controller

PCIe

CPU

DRAM

Flash
How FPGAs Address the Computational Storage Problem
FPGAs in Storage Today

› Flash controllers

› Storage Systems
   » Cache-offload
   » Storage System & Switching connectivity
   » Data Reduction
FPGA Advantages for Computational Storage

› Flexible, fully customizable architecture adapts to specific applications
  » Massive parallelism, I/O and customizable data path

› Performance, power and latency of dedicated HW + reconfigurability of SW

› More economical than ASIC/ASSP for many applications
FPGA Advantages for Changing Standards

Architecture easily adapts to latest compression algorithms

Gzip Accelerator  Brotli Accelerator  Zipline Accelerator
Computational Storage Deployment Options
Integrated Accelerator and Flash

Benefits:

» Easy to implement - plug & play
» Adding capacity adds accelerators + performance
» Ability to optimize BW between accelerator and flash
» Ability to customize FTL for specific workloads

Xilinx Partners:

» Samsung
» Scaleflux
Computational Storage Processor (CSP)

- Accelerator and Storage on same PCIe subsystem

- Benefits:
  - Independent SSD & acceleration scaling
  - Plugs into standard slot
  - PCIe Peer-to-peer transfers for high bandwidth and low latency

- Xilinx Partners:
  - Bittware
  - Eideticom
  - Xilinx
Computational Storage Array (CSA)

- Accelerator in-line with storage

- Benefits:
  - SSD vendor independence
  - Independently scale accelerators and SSDs
  - Ability to optimize BW between accelerator and SSDs

- Xilinx partners:
  - Bittware
Computational Storage Example Application
Example of Analytics Acceleration

Q1: “Which cities originate the most flights with >10min delays?
Q2: “Which airport in the Bay Area has the worst record?

Airline traffic in the USA from 1970 to Present
- Flight Data — 1.2B Entries
- Airport Data — 500M Entries
- Planes Data — 700M Entries

QUERY PERFORMANCE

<table>
<thead>
<tr>
<th># FPGA Accelerators</th>
<th>Relative Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>1x</td>
</tr>
<tr>
<td>1</td>
<td>4x</td>
</tr>
<tr>
<td>2</td>
<td>7x</td>
</tr>
<tr>
<td>4</td>
<td>13x</td>
</tr>
</tbody>
</table>
Storage Developer Flow
Runtime Stack

- **Storage Accessed via NVMe Stack**

- **Computational Storage / Accelerator Discovered, Managed, Orchestrated by OpenCL Stack**

- **Shared Memory Space in the Compute Function Glues the Datapaths together**
Problem: Need to parse through large amount of data to find the records of interest.

Example:
- Analytics – Need the records for a time range for just one of many products included in the database.

Solution: Push down Scan, Filter, Aggregate to storage.

Why?
- Higher Throughput
- Lower Latency
- CPU offload
- Lower TCO
Scan Query on standard NVMe

Query Execution / Storage Engine

tblIterator = new tableIterator(myTableName);
void* blockBuffer = malloc(..);
tblIterator.getNextBlock(blockBuffer);
tuples = scanBlock(blockBuffer,schema,predicates);
Scan Query on computational NVMe

tblIterator = new tableIterator(myTableName);
cl_mem blockBuffer, tuples = clCreateBuffer(...);
tblIterator.getNextBlock(blockBuffer);
runKernel(scanBlock, blockBuffer, schema, predicates, tuples);
clMigrateMemObjects(tuples)
OpenCL CSD example
Why does P2P improve performance?

- Interrupts
- Kernel / User mode transitions
- Copy time
- 1 us = 1000 (1 GHz) – 3000 (3 GHz) cpu clock cycles

I/O Latency Optimization with Polling, Damien Le Moal
Vault – Linux Storage and Filesystems Conference – 2017
Examples: SmartSSD® CSD and Ecosystem

**Storage Services:** Comp/Decomp, Encrypt/Decrypt, Metadata management, Erasure Coding, Real-time Analytics & Biz Intelligence: DB Query (Spark, PostgreSQL, ..), Log analytics, genomics, physics, Rich Media and ML: transcoding, live streaming, object detection

**Storage Acceleration IP**

**IP Dev Toolchain:** Runtime, Libraries, API, Drivers

**Connectors to Application Frameworks**

**Video encoding**
- AI and ML
  - Accelerated application frameworks
- DB acceleration
  - Storage and Virtualization
- Comp/decomp
  - Encrypt/decrypt
  - Erasure coding
- Accelerated storage services

**Acceleration platform**
- Bring your own IP

---

**SmartSSD® CSD and Ecosystem**

**Examples:**
- SmartSSD® CSD and Ecosystem
- Storage Services
- IP Dev Toolchain
- Connectors to Application Frameworks

**Storage Services:**
- Comp/Decomp, Encrypt/Decrypt, Metadata management, Erasure Coding
- Real-time Analytics & Biz Intelligence: DB Query (Spark, PostgreSQL, ..), Log analytics, genomics, physics
- Rich Media and ML: transcoding, live streaming, object detection

**Connectors to Application Frameworks**

**Storage Acceleration IP**

**IP Dev Toolchain:** Runtime, Libraries, API, Drivers

---

**Video encoding**
- AI and ML
  - Accelerated application frameworks
- DB acceleration
  - Storage and Virtualization
- Comp/decomp
  - Encrypt/decrypt
  - Erasure coding
- Accelerated storage services

**Acceleration platform**
- Bring your own IP
Technology Preview SmartSSD® IP Development

- Deploy off-the-shelf IP and solutions from our partners
- Use familiar Xilinx tools to develop new IP or redeploy existing accelerator IP from ASICs or FPGAs
- Use custom IP development services from Samsung and Xilinx partners
- Enterprise Class SSD Controller: NVMe1.3, CMB, AES256
  - 4TB Capacity
  - 330K LUTs total in dynamic region available for acceleration IP
  - 4GB FPGA DDR
  - External interface: PCIe Gen3x4
Future Directions
Current Data Center Architecture:
Fixed Resources, Sub-optimal Utilization
Future Data Center: Disaggregated and Composable

Challenge: Reduced Bandwidth and Increased Latency
Computational Storage & Fabric

- Enables composability without significant performance penalty

**Benefits**

- Performance and latency benefits of computational storage
- Scale compute / storage independently
- Higher density per rack
- Lowest TCO
Future Data Center: Distributed Adaptive Acceleration

- Composable accelerated storage, networking and compute
- Optimized for each workload
- Optimal infrastructure utilization
Computational Storage: Accelerating High-Speed Storage Systems

Computational storage addresses a broad range of application bottlenecks

Offers data center operators >5x performance boost and up to 2x reduction of TCO

Xilinx is leading the way in distributed adaptive acceleration