Introducing **DPU:**
Data-storage Processing Unit

*Placing Intelligence in Storage*

**Qing Yang** 杨庆
Founder & CSO, Shenzhen Dapu Microelectronics Ltd Co.
Distinguished Engineering Professor, IEEE Fellow
University of Rhode Island
Data Explosion!

Δ_{18m} = \sum_{-\infty}^{\infty}
Ali-Cloud last Year: 5 Times!
Data Explosion!

Enterprise SSD Market

Worldwide Enterprise PCIe SSD Shipments (M000)

Rapid Advances of Storage Technologies
Data Growth + Media Tech

✧ **Big Data, Cloud**: Data Explosion
  - Applications Demand Fast Data---
    High Performance, Secure, Reliable, Recoverable

✧ **Emerging Device Tech, more Cost-Effective**:  
  - Flash, PCM, MRAM, RRAM

These placed great challenges to the storage control and management

Existing storage controllers are far behind!
History of Computing

Decades ago, Displays were controlled by CPU/MCU
➢ Resolution, color, pixels increased greatly
➢ CPU/MCU could no longer control modern displays
➢ As a result: GPU was born and developed

Today
GPU Plays a revolutionary role in computing!

We Believe
Storage control of big data has come to a historical point!
CPU/MCU can no longer manage exponential growth of data and a variety of storage media technologies:
Therefore, We introduce the first ever:

Data-storage Processing Unit: DPU
Machine learning optimizing I/O & Prolonging storage life

Data analysis and encryption with hardware inside storage

Greatly reduce total amount data over I/O bus.

Improving data throughput rate
Improving the performance of the entire system
Major Functions in DPU

- **Media Managements**
  - Flash Control
  - Machine Learning of I/O Patterns
  - Minimizing Erasures & Adaptive RAID

- **Advanced Data Analytics**
  - Processing in Storage: PIS
  - Placing data intensive computation closest to where data is stored

- **Storage Architecture**
  - Hierarchy and Tiering:
  - Dedupe, Snapshot, Replication, and Failure Tolerance
  - Distributed SAN, E-W connectivity, NVMe over the fabric
Storage Media Management

- **Physical Properties of Flash Memory**
  - Reads are faster than writes
  - Limited erase cycles
  - No in place write → GC, WL

- **Write Amplification Problem**
  - Slow down I/Os, Increase wearing, and hogging resources

<table>
<thead>
<tr>
<th>Valid page</th>
<th>To be erased block</th>
<th>A new block</th>
</tr>
</thead>
<tbody>
<tr>
<td>V I I V I I I V</td>
<td>V V V V V V V V V</td>
<td>V V V V E E E E</td>
</tr>
<tr>
<td>I I I I I I V I I</td>
<td>E E E E E E E E E</td>
<td>E E E E E E E E E</td>
</tr>
<tr>
<td>I I I I I I V V I</td>
<td>E E E E E E E E E</td>
<td>E E E E E E E E E</td>
</tr>
</tbody>
</table>
Reinforcement Learning

❖ Classify I/Os into groups of similar or same rewrite intervals
  ➢ Features and attributes
    ✓ {R/W LBA, Timestamp, Re-reference interval, Recency, feedback, GC information}
  ➢ Pages of the same class will be written in one block
    ✓ High performance, minimal WA

❖ Recognize I/O Patterns at Production Site
  ➢ Train and learn I/O behaviors after deployment
  ➢ Optimization kicks in after a week or so
  ➢ Adapt to any environment and applications
Measured Erase Count Results

Normalized Erase Counts

Factor

HMO  MDSO  PROJ0  PROJ1  RSRCH0  SRC1-0  STG0  TSO  USR2  WDEVO  WEB2

Random  ML  Normal
Major Functions in DPU

- Media Managements
  - Flash Control
  - Machine Learning of I/O Patterns
  - Minimizing Erasures & Adaptive RAID

- Advanced Data Analytics
  - Processing in Storage: PIS
  - Placing data intensive computation closest to where data is stored

- Storage Architecture
  - Hierarchy and Tiering
  - Dedupe, Snapshot, Replication, and Failure Tolerance
  - Distributed SAN, E-W connectivity, NVMe over the fabric
ADA: HW Search & Sort in DPU

❖ Over 80% of Data are Unstructured
  ➢ Process of text data is critical
  ➢ Software scanning is slow

❖ Research on Accelerators for Text Search
  ➢ Maximizing DRAM bandwidth
  ➢ I/O is still a burden

❖ Sorting & KV Store
  ➢ HW Sorting
  ➢ Graph Processing
ADA1: REGISTER in DPU

• Regular Expression Grabbing Inside STORage
  ➢ HW search in SSD where data is stored
  ➢ Only results or related files are sent to the host
ADA2 In-storage sort module

Divide and Conquer HW Sort Module

Unsorted data input:

- Linear-time Sorter 0
- Linear-time Sorter 1
- FIFO merger 1
- FIFO merger 2
- FIFO merger 3
- Chunk sorted output

A0 A1 A2
B0 B1 B2
ADA2: Sort performance

Single core speedup: 4.6~6x.
Multi core speedup: 2~2.8x
ADA3: Graph Preprocessing in DPU

Minimum Spanning Tree (MST):

MST calculation:
1. Sort the entire edges
2. Edge connection
Post processing on single core

10, 7, 6, 5
21, 12, 11, 3
13, 10, 8, 7
22, 4, 4, 1
39, 23, 32, 5
12, 8, 7, 6
40, 33, 21, 8
24, 23, 20, 1
Post processing on single core

10, 7, 6, 5
21, 12, 11, 3
13, 10, 8, 7
22, 4, 4, 1
39, 23, 32, 5
12, 8, 7, 6
40, 33, 21, 8
24, 23, 20, 1
Post processing on single core

10, 7, 6, 5
21, 12, 11, 3
13, 10, 8, 7
22, 4, 4, 1
39, 23, 32, 5
12, 8, 7, 6
40, 33, 21, 8
24, 23, 20, 1
Post processing on single core

10, 7, 6, 5
21, 12, 11, 3
13, 10, 8, 7
22, 4, 4, 1
39, 23, 32, 5
12, 8, 7, 6
40, 33, 21, 8
24, 23, 20, 1

Finished build B-tree of chunk node

Trim direction
Post processing on single core

Trim form B-tree by in-order traversal:
Remove 1 and connect MST
Post processing on single core

10, 7, 6, 5
21, 12, 11, 3
13, 10, 8, 7
22, 4, 4
39, 23, 32, 5
12, 8, 7, 6
40, 33, 21, 8
24, 23, 20, 1

Trim direction

Insert next chunk head
Post processing on single core

10, 7, 6, 5
21, 12, 11, 3
13, 10, 8, 7
22, 4, 4
39, 23, 32, 5
12, 8, 7, 6
40, 33, 21, 8
24, 23, 20, 1

Trim form B-tree by in-order traversal:
Remove 1 and connect MST
Post processing on single core

10, 7, 6, 5
21, 12, 11, 3
13, 10, 8, 7
22, 4, 4
39, 23, 32, 5
12, 8, 7, 6
40, 33, 21, 8
24, 23, 20

Trim direction

Insert next chunk head
Post processing on single core

Trim form B-tree by in-order traversal:
Remove 3 and connect MST
Post processing on single core

B-tree stops when traverse all the nodes
MST performance

96-cores CISC vs single-core CPU baseline: 11.47~17.2x
ADA4: HW Deserialization

❖ Future world will be sensor driven world
   ➢ Huge amount of sensing data files
   ➢ Numbers are stored in readable and exchangeable formats: ASCII, Unicode etc.

❖ To Process Data Using Computers
   ➢ Readable data need to be converted to binary
   ➢ Host CPU is very inefficient
   ➢ Time Consuming, up to 60% of Total Processing Time
Performance of DPU vs. Server CPU

Throughput of Server

Throughput of DPU

JASPA  B-tree  Breadth first Search0  Breadth first Search1  LU Decomposition1  LU Decomposition2  LU Decomposition3  LU Decomposition4  hybridsort
Major Functions in DPU

- **Media Managements**
  - Flash Control
  - Machine Learning of I/O Patterns
  - Minimizing Erasures & Adaptive RAID

- **Advanced Data Analytics**
  - Processing in Storage: PIS
  - Placing data intensive computation closest to where data is stored

- **Storage Architecture**
  - Hierarchy and Tiering
  - Dedupe, Snapshot, Replication, and Failure Tolerance
  - Distributed SAN, E-W connectivity, NVMe over the fabric
IST: Intelligent Storage Tiering

- **Media:** Flash, PCM, MRAM, Memristor etc:
  - Different Speed
  - Different Cost

- **What Do Users want:**
  - $\downarrow$ & Speed $\uparrow$ & Power $\downarrow$ & Ease of use & Reliability $\uparrow$

---

IST in DPU
Distributed SAN Functions in DPU

❖ **East-West Connectivity**
  ➢ Support Distributed SAN with HW
  ➢ NVMe over the fabric

❖ **DPU-Link**
  ➢ Allow customized HW/Chip to be connected
  ➢ AI training and inference made fast

❖ **Support Multiple Protocols**
  ➢ iSCSI, FC, NVMe over the fabric
  ➢ NAS card
  ➢ Snapshot, Replication, Recovery, and more
Summary and Conclusions

A New Concept for Next Generation Storage
Proven Advantages on Current SSDs

- Optimize Media
- Better I/O
- Analytics PIS
- QoS Apps

DPU For Storage
Thank You!

Q & A