SoftFlash: Programmable Storage in Future Data Centers

Jae Do
Researcher, Microsoft Research
The world’s most valuable resource

Data is everywhere!

Values from Data!

Need infrastructures for Volume and Velocity of data!
Case Study: OCP storage server (1/2)

- High cost of moving data
- Needs to scale up due to data growth
Case Study: OCP storage server (2/2)

Throughput gap of 66x

16 lanes of PCIe = ~ 16 GB/s

32 channels X ~500 MB/s = ~16 GB/s

64 Flash SSDs X ~16 GB/s/SSD = ~1TB/s
Programmable components in data center

- Software-defined control and management is an inevitable trend that is already touching other parts of the data center infrastructure
  - Software-Defined Networking (SDN) making network switches and server NIC cards increasingly programmable for enhanced network-wide functionality
  - Programmable GPGPUs and FGPAs leveraged by new generations of applications like deep learning
- Rapidly-changing requirements can be supported on-the-fly once DC infrastructures become dynamically programmable

Don’t Leave Storage Behind!
Next up: Storage (1/2)

- Unfortunately, the lack of such programmable capabilities in storage results in a major disconnect in terms of **the speed of innovation** between application/OS and storage infrastructures!

- While application/OS is patched with new/improved functionality **every few weeks** at cloud speed while storage devices are off limits for such sustained innovation during their hardware life cycle of **3-5 years** in data centers.
A fully programmable storage gives opportunities to better bridge the gap between application/OS needs and storage capabilities/limitations, while allowing us to innovate in-house at cloud speed.

Flash SSDs can be programmable?
Today’s NAND Flash SSD

Host Server

(PCIe) Flash SSD

PCIe gen3 (4 lanes)
Hardware b/w = 4 GB/s
Achievable b/w = 2-2.5 GB/s

Throughput gap of 2x - 8x

16-32 channels @ 500MB/s
Total = 8-16 GB/sec
Past efforts – Smart SSD (1/2)

- Goal: Exploring the opportunities and challenges associated with running selected database operations inside an SSD
  - Used Samsung SAS SSD
  - Modified Microsoft SQL Server 2012 to offload database operations onto a Samsung Smart SSD
  - Simple selection and aggregation operators were hard-coded and compiled into the firmware of the SSD

Query Processing on Smart SSDs: Opportunities and Challenges, SIGMOD 2013
Past efforts – Smart SSD (2/2)

TPC-H Q6:

```
SELECT SUM (EXTENDEDPRICE*DISCOUNT) 
FROM LINEITEM 
WHERE SHIPDATE >= 1994-01-01 AND 
    SHIPDATE < 1995-01-01 AND 
    DISCOUNT > 0.05 AND 
    DISCOUNT < 0.07 AND 
    QUANTITY < 24
```

Result: Compared to the base case,
- 70% better query response time
- 2X energy efficiency

Device became a performance bottleneck!
The dev. environment was not ready!
Past efforts – YourSQL (1/2)

Goal: Accelerating data-intensive queries with the help of hardware pattern matcher
- Used Samsung PCIe SSD
- Modified a variation of MySQL to realize early filtering of data by offloading data scanning of a query to programmable SSDs
- Developed a framework (called Biscuit) that follows a data-flow model
Past efforts – YourSQL (2/2)

Result: Compared to the base case,
- 11X Speed up

Computing resource is not powerful enough!

Existing applications need to be redesigned!

Filtering Query
SELECT l_orderkey, l_shipdate, l_linenumber
FROM lineitem
WHERE l_shipdate = '1995-1-17'
Challenges of past efforts

- Not enough “spare” processing power
- H/W architecture limitations
- Programming tools are not application-developer friendly
- Prototype devices are not accessible to application developers
Disruptive trend that enable SoftFlash

- Abundant resources inside SSD
- Frugal resources inside SSD
- Embedded CPU, proprietary firmware “OS”
- General purpose CPU, server-like OS (Linux)

(Ease of programmability inside SSD)
The SoftFlash project

- **Goal:** Embrace flash SSDs as a first-class programmable platform in the cloud data center
  - Add custom capabilities to storage over time
  - Better bridge the gap between application needs and flash media capabilities/limitations
  - Innovate in-house at cloud speed

<table>
<thead>
<tr>
<th>Hardware Prototype</th>
<th>Powerful and flexible prototype board with enterprise-grade capabilities and resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Framework</td>
<td>Linux, SDK, user/kernel libraries for the on-chip H/W accelerators, built-in FTL</td>
</tr>
<tr>
<td>Application</td>
<td>Moving compute closer to storage, flexible storage interface, secure computation</td>
</tr>
</tbody>
</table>
Hardware prototype (1/3) – Dragon fire card

- DragonFire Card (DFC): A programmable SSD device designed by DellEMC and NXP (acquired by Qualcomm)
  - Developed for research purposes
  - Can be equipped with various forms of NVM (RAM, Flash, etc)
  - Composed of two types of boards: main and storage boards
- An open-source community ([http://github.com/DFC-OpenSource](http://github.com/DFC-OpenSource)) is organized around this hardware platform with teams working on a wide range of issues
Hardware prototype (2/3) – Main board

- 4 X 10Gbps SFP+ - RoCE protocol
- 16GB DRAM
- 1 X 4-Lane PCIe 3.0
- 2 X 4-Lane PCIe 3.0
- 8-Core ARMv8 (1.8GHz)

Hardware Accelerators
- Compression (10Gbps)
- Encryption (10Gbps)
- PatternMatching (20Gbps)
Hardware Prototype (3/3) – Storage Board

- 4 X DIMM Slots
- FPGA (storage controller)
- 2 X M.2 SSDs
Revisit: OCP storage server

16 lanes of PCIe = ~ 16 GB/s

2.3GHz/core X 20 cores = 46GHz

Throughput gap of 66x

64 Flash SSDs X ~16 GB/s/SSD = ~ 1TB/s

1.8GHz/core X 8 cores X 64 SSDs = 921.6GHz

Compute capability gap of 20x

+ DRAM, H/W accelerators!
Application 1: Move compute closer to data

- Reduced data movement across storage/network/memory/CPU for compute

Example:
- Stream analytics over data logs
- Select/Project/Aggregation for relational database/data warehouse services
Application 2: Agile, flexible storage

- Custom SSD capabilities to better meet application needs

  - Example:
    - Multi-streamed writes for cloud storage platform
    - Read I/O priority for latency-sensitive services
    - Atomic writes for transactional services

Agile, flexible storage interface leveraging programmability within SSD
Example) Stream-Aware Flash

- **Data Stream #1**
  - Incoming Writes
  - Flash pages
  - Erase Block A
  - Regular (stream-unaware) SSD

- **Data Stream #2**
  - Tag writes with stream ID
  - Erase Block B
  - Stream-aware Programmable SSD (2x device lifetime, 1.5x read throughput)
Application 3: Secure computation in cloud

- SSD provides a trusted domain for secure computation on encrypted data, without cleartext leaving the device

Example:
- Existing and new scenarios in a “trusted cloud” setting -- user stores encrypted data in the cloud and needs to do compute over it
Value propositions

Moving compute close data

Agile, flexible Storage

Secure Computation

Programmable SSD
Big Data Analysis (1/6)

- Today, big data analytics fetches huge volumes of data from storage and processes it in host server

- Programmable SSDs enable data analytics “inside” storage
  - Exploit higher bandwidth inside SSD (vs. SSD external interface)
  - Leverage ARM cores + hardware offload engines inside SSD
Big Data Analysis (2/6)

- Efficient use of heterogeneous hardware in the data center for higher performance @ lower power
  - Free up expensive host server CPU + memory resources, opportunities to increase service density
  - Reduced energy footprint due to significantly less data movement + low power compute inside SSD
Big Data Analysis (3/6) – Traditional Arch.

Compute Cluster

1. Compute Node
2. Compute Node

Storage Cluster

1. Storage Node
2. Storage Node
3. Storage Node

A. Fetch compressed data from storage cluster
B. Decompress data
C. Decode data
D. Do required computations (filtering, aggregation, etc.)
Big Data Analysis (4/6) - programmable SSDs

Compute Cluster
- Compute Node
- Compute Node

Storage Cluster
- Storage Node
- Storage Node
- Storage Node

A. Return only results to Storage Cluster
B. Decompress data
C. Decode data
D. Do required computations (filtering, aggregation, etc)
Big Data Analysis (5/6) – Apache Hive

- Hive is a data warehouse infrastructure built on top of Hadoop
  - Designed to enable easy data summarization, ad-hoc querying and analysis of large volumes of data

- Encoding based on data type
  - Run-length encoding for integer
  - Dictionary encoding for string

- Compression using a codec
  - Zlib or Snappy
Big Data Analysis (6/6) – Preliminary Result

- Scanning a ZLIB-compressed, integer dataset (1 Billion rows, ~10GB) on a X86 server or inside the programmable SSD
- Note that only a single core was used!

What if we used +60 programmable SSDs?

![Graph showing throughput improvement with multiple Programmable SSDs compared to X86 server]

- X86 (Intel Xeon @2.3GHz)
- Programmable SSD (ARM @1.8GHz)
- Programmable SSD (ARM @1.8GHz) + Decompression Engine

Throughput (M row)

- 0
- 50
- 100
- 150
- 200
- 250

- 55.4
- 40.6
- 272.47

4.9x improvement
Conclusion

• The SoftFlash project proposes to create a software-defined storage substrate of flash SSDs in the data center that is as programmable, agile, and flexible as the applications and operating systems accessing it from servers.

• This is made possible by recent disruptive trends in the flash storage industry towards increased easy of programmability and abundance of resources in side the SSD

• We are still in an early stage! STAY TUNED!
Thank you!

jaedo@microsoft.com