OX-App: A Framework for Application-specific FTLs on Tiered Storage

Ivan Luiz Picoli
Ph.D. Fellow
Near-data Processing

Put Everything in Future (Disk) Controllers (it’s not “if”, it’s “when?”)

Jim Gray
http://www.research.Microsoft.com/~Gray

Acknowledgements:
Dave Patterson explained this to me a year ago
Kim Keeton
Erik Riedel
Catharine Van Ingen

Basic Argument for x-Disks

- Future disk controller is a super-computer.
  - 1 bips processor
  - 128 MB dram
  - 100 GB disk plus one arm

- Connects to SAN via high-level protocols
  - RPC, HTTP, DCOM, Kerberos, Directory Services,
  - Commands are RPCs
  - Management, security,
  - Services file/web/db/… requests
  - Managed by general-purpose OS with good dev environment

- Move apps to disk to save data movement
  - Need programming environment in controller

Jim Gray, NASD Talk, 6/8/98
http://jimgray.azurewebsites.net/jimgraytalks.htm
The time is now!

- **SoC**: 8x ARM v8 64-bit HW accel.
- **PCIe x16**
- **4x 10GB Links**
- **Non-volatile Memory**: M.2 Open-channel SSDs

**Dragon Fire Card**

**IT UNIVERSITY OF COPENHAGEN**
SSD Controller - Flash Management is essential

Flash Translation Layer (FTL)
Implemented on SSD controller or SSD driver
OX Controller

Host Linux Kernel
- NVMe Device Driver + LightNVM

Physical Interconnection (PCIe, Network)

Interconnection Layer
- PCIe Handler
- Network Fabrics Handler

NVMe queue support + command parser Layer
- NVMe Over PCIe
- NVMe Over Fabrics

Storage Translation Layer
- LightNVM Support
- Full-fledged FTL
- Other FTLs

Storage Media Manager Layer
- Specific Storage Media Managers (SMMs)

Storage Media
- NAND SLC
- NAND MLC
- DDR
- Other NVM

IT UNIVERSITY OF COPENHAGEN
Architecture Overview

- Network Fabrics
- Host Applications
- Open Channel
- SSD
- 16 GB DRAM DDR4
- OX Controller - ARM v8
OX-App: A Framework for Application-specific FTLs

- Built on top of OX Controller, in the FTL layer;
- **FTL support:** It provides an interface to develop FTLs;
  - 11 modules with predefined interfaces;
- **Near-data processing:** It allows custom NVMe commands to be processed into the SSD controller;
In progress: Log Management and Recovery
OX-Block

- Maintain the state and metadata of each block during the device lifetime;
- Manage a 4KB-granularity mapping table and map logical-physical addresses;
- Guarantee integrity and recovery of block metadata and mapping table after power-off;
- GC is performed per channel, where several channels might run the GC in parallel to limit the write speed;
- User writes are not allowed in channels running GC;
- Handle write and erase errors;
Near-Data Processing

Back-end NVM Mgmt:
- ECC;
- Data retention;
- RAID;
- Block metadata;

Front-end NVM Mgmt:
- Wear-leveling;
- L2P translation;
- Garbage collection;
- Write-caching;

Application functions:
- Checkpointing;
- Access Methods;
- Log Management;
- Filtering;
- Other I/O rules;
Repository and contact

OX Controller:
https://github.com/DFC-OpenSource/ox-ctrl/

OX, OX-App and OX-Block Documentation:
https://github.com/DFC-OpenSource/ox-ctrl/wiki

Ivan Luiz Picoli
ivpi@itu.dk