Cosmos OpenSSD:
A PCIe-based Open Source SSD Platform

Yong Ho Song[1], Sanghyuk Jung[1],
Sang-Won Lee[2], Jin-Soo Kim[2]

Hanyang University[1], Sungkyunkwan University[2], Korea

OpenSSD Introduction

What's the OpenSSD Project

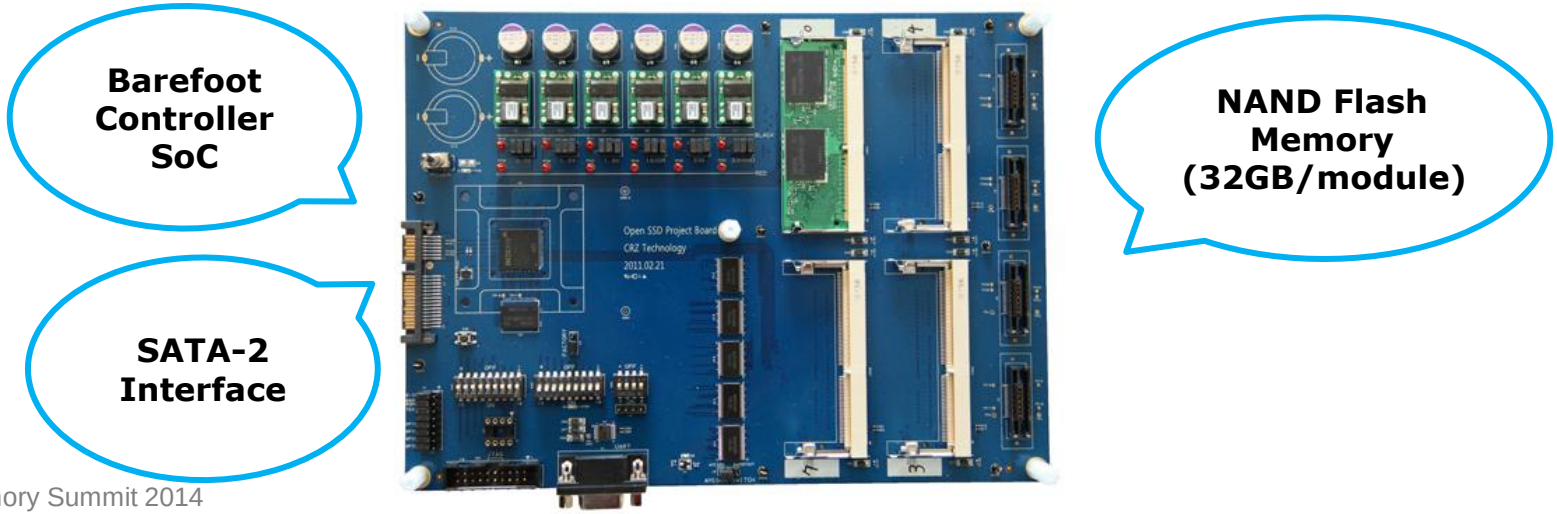
- Open-source SSD platform for research and education on the SSD technology since 2011
- New "OpenSSD platform" for developing SSD firmware, controller hardware, and host software
- Contribution
  - Indilinx (Merged to OCZ in 2012)
  - HYU (Hanyang University), Korea
  - SKKU (Sungkyunkwan University), Korea

Why OpenSSD?

- Solve your problem in a real system
- Share your solution with people in society
- Design your own SSD controller, if possible
- Contribute to “open” community
- Use it as a PC disk
- Play for fun

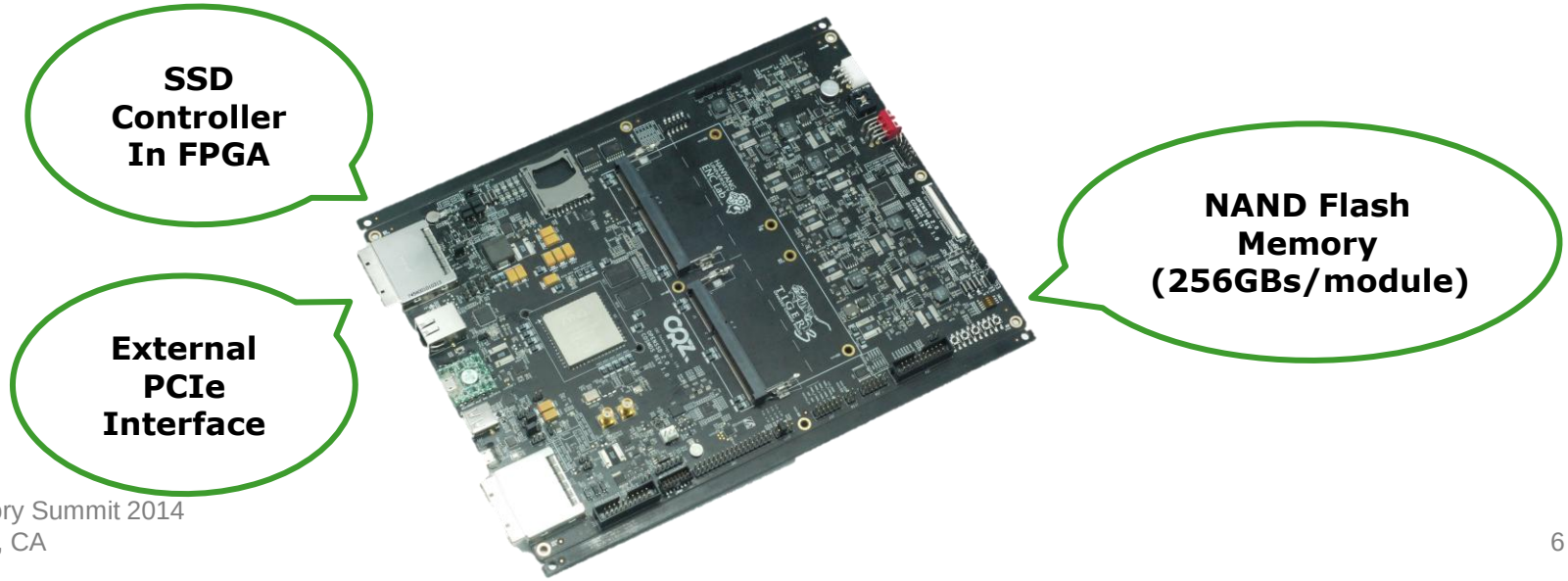
OpenSSD Project History

- Jasmine OpenSSD (2011)
  - SSD controller: Indilinx Barefoot (SoC w/SATA-2)
  - Firmware: SKKU VLDB Lab
  - Users from 10+ countries
  - 10+ papers published



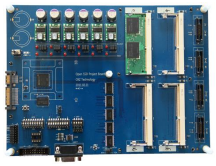
OpenSSD Project History

- Cosmos OpenSSD (2014)
  - SSD controller: HYU Tiger3 (FPGA w/PCIe Gen2)
  - Firmware: HYU ENC Lab, SKKU VLDB Lab
  - Users from ?? countries (at least one)
  - ?? papers to be published (at least three)



Comparison among the Platforms

	Jasmine OpenSSD	Cosmos Prototype (Tiger2)	Cosmos OpenSSD (Tiger3)
SSD Controller	Indilinx Barefoot (SoC)	HYU Tiger2 (FPGA)	HYU Tiger3 (FPGA)
Year	2011	2012	2014
Host Interface	SATA-2	PCIe Gen1.1 (AHCI Subset)	PCIe Gen2 (AHCI Subset)
Storage Capacity	128 GB	512 GB	512 GB
NAND Data Interface	Asynchronous	Asynchronous	Synchronous
DRAM Capacity	64 MB	512 MB	1 GB



OpenSSD Project Homepage

http://www.openssd-project.org

Cosmos OpenSSD Platform Features

Design Objective

- A simple NAND storage platform to be used for
  - Firmware development and evaluation
  - Controller architecture exploration
  - New controller design (e.g. Intelligent SSD)

➔ An open-source SSD platform with AP-class CPUs and other hardware resources

Cosmos SSD Hardware Features

FPGA		Xilinx Zynq-7000 series
MCU	**Type**	Cortex™- A9
	Clock Frequency	667 MHz
Storage	**Total Capacity**	512 GB
	NAND Organization	4-Channel / 8-Way
DRAM	**Device Interface**	DDR3 (533 MHz)
	Total Capacity	1 GB
Bus	**System**	AXI-Lite (Bus width: 32 bits)
	Storage Data	AXI (Bus width: 64 bits, Burst Length: 256)
SRAM		256 KB (FPGA Internal)
Error Correction Code		BCH 32 bits/2 KB
Host Interface		PCI-Express Gen2 4-Lane (2 GB/s)
Power Measurement		NAND Flash and Board Power Measurement (External ADC Module or NI DAC)

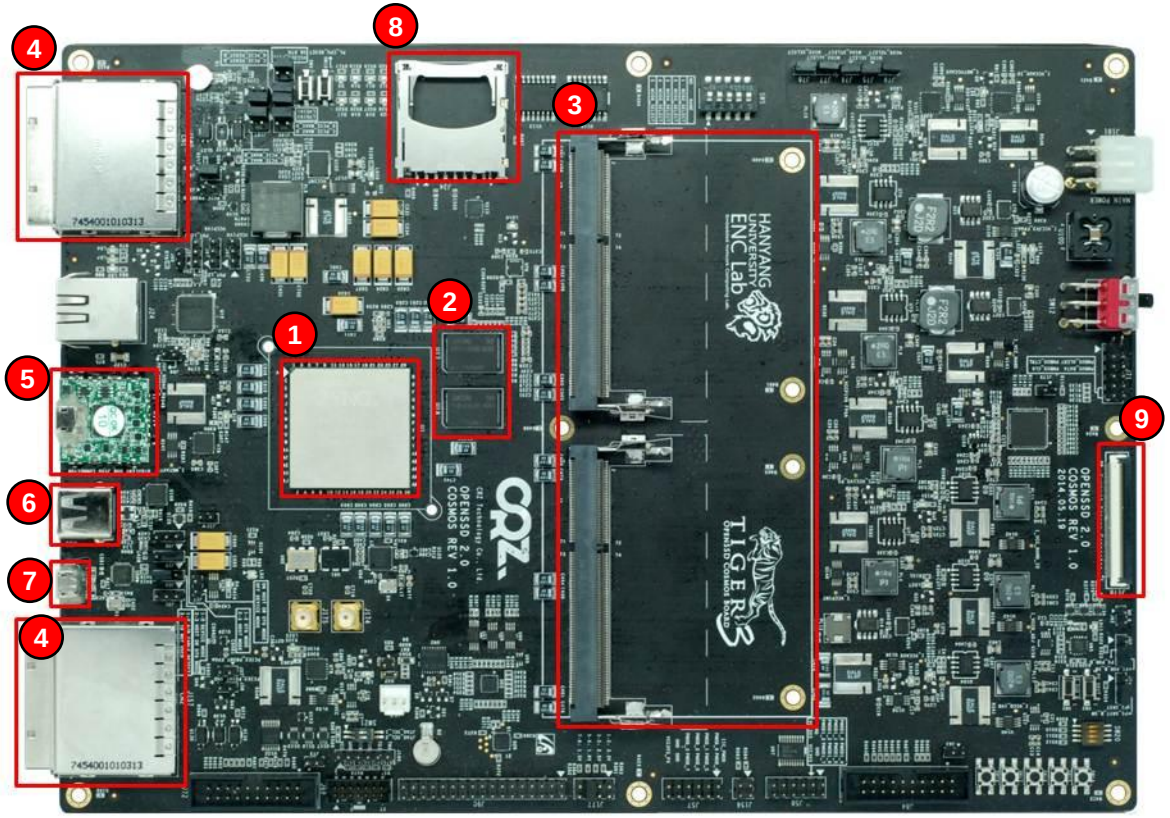
NAND Flash

NAND Flash Memory Specification		
Vendor / Model		SK Hynix / H27QDG8VEBIR-BCB
Device Interface		Synchronous (ONFI 2.2)
Cell Technology		MLC (2 Bits/cell)
Capacity		16 GBs/package
Page Size	Data Area	8192 Bytes/page
	Spare Area	640 Bytes/page
Dies per Package		4 dies (QDP)
Data Cycle		20 ns (DDR: 10 ns)
Page Read		70 us (typical)
Page Program		1400 us (typical)
Block Erase		3.5 ms (typical)
Speed		NV-DDR100

DRAM

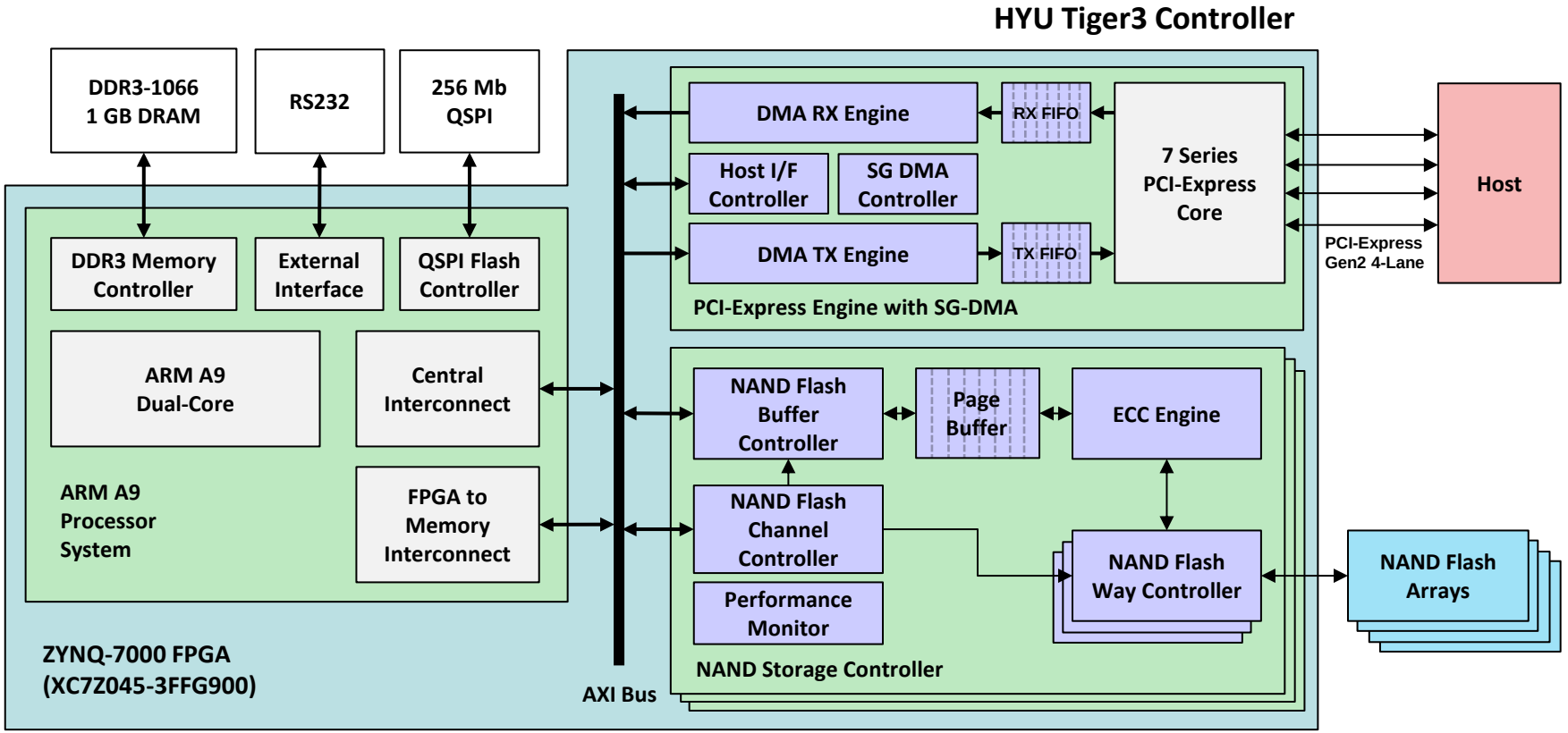
DRAM Memory Specification	
Vendor	Samsung (K4B4G1646B-HCK0)
Device Interface	DDR3-1066F
DRAM Device Bus Width	16 Bits
Capacity	1 GB (512 MBs x2)
CAS Latency	7 Cycles
CAS Write Latency	6 Cycles
RAS to CAS Delay	7 Cycles
Precharge Time	7 Cycles
Throughput	8.5 GB/s

Cosmos SSD Platform Board



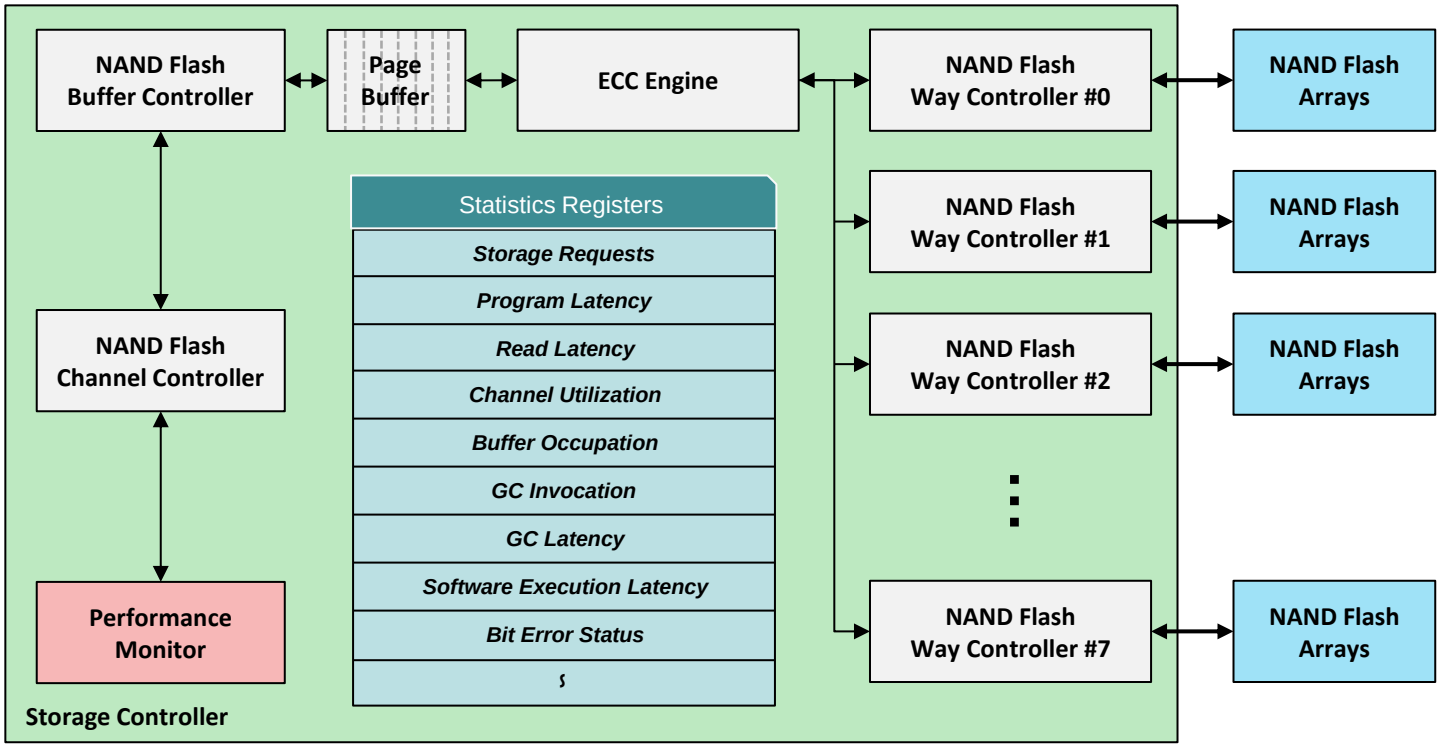
No.	Information
1	Xilinx Zynq-7000 (HYU Tiger3 Controller)
2	DDR3 (512 MB x2)
3	NAND SODIMM x2
4	External PCIe x2
5	USB JTAG
6	UART
7	USB 2.0 OTG
8	SD Card
9	Power Measurement Connector

HYU Tiger3 Controller

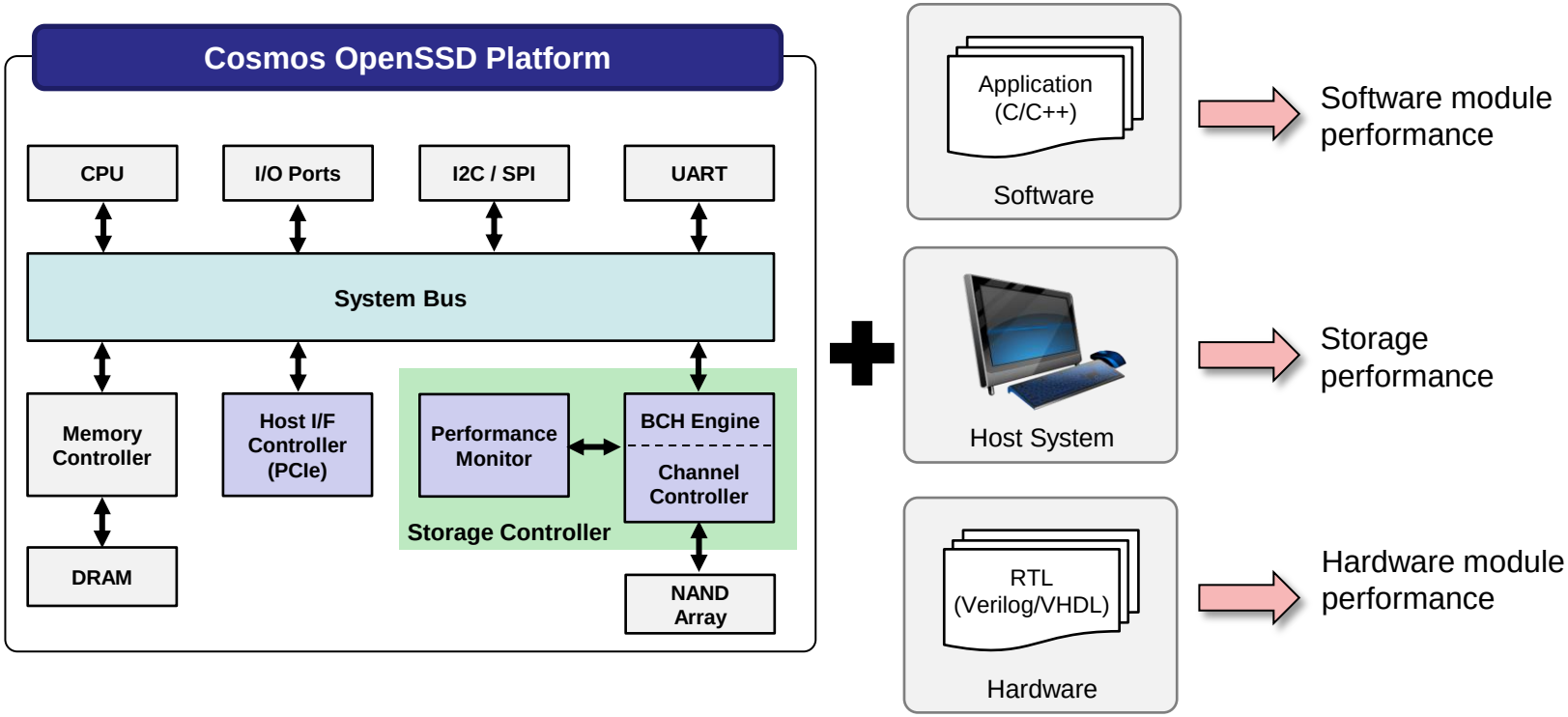


Storage Performance Monitor

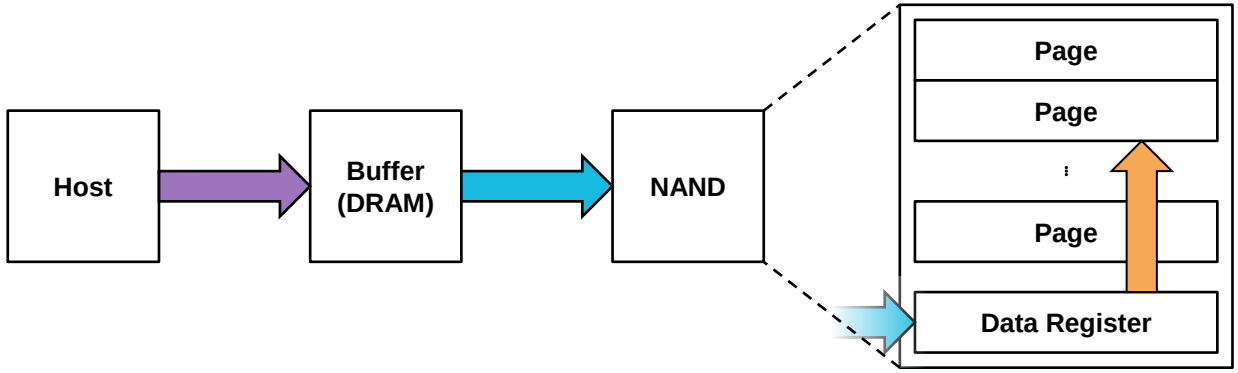
Statistics on the performance and utilization of hardware resources

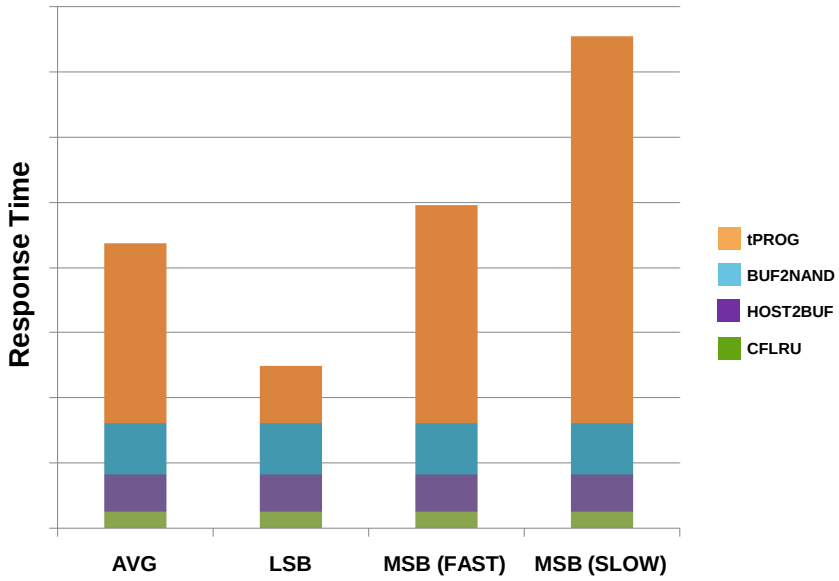


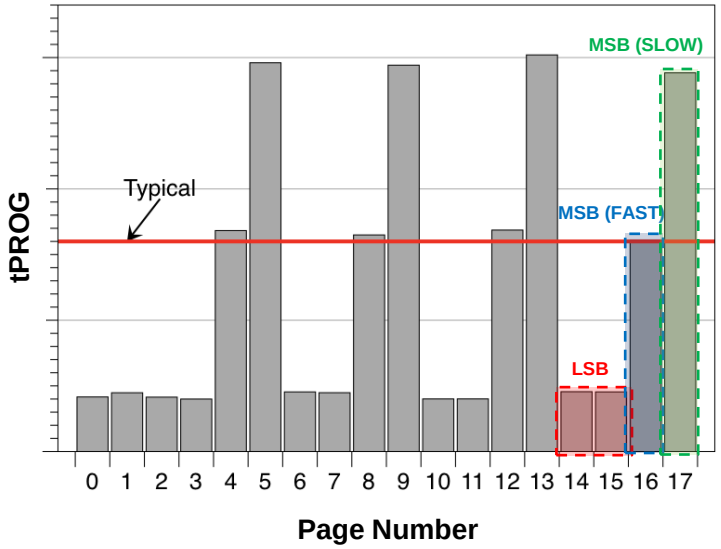
Use of Performance Monitor



Example: Operation Time Breakdown







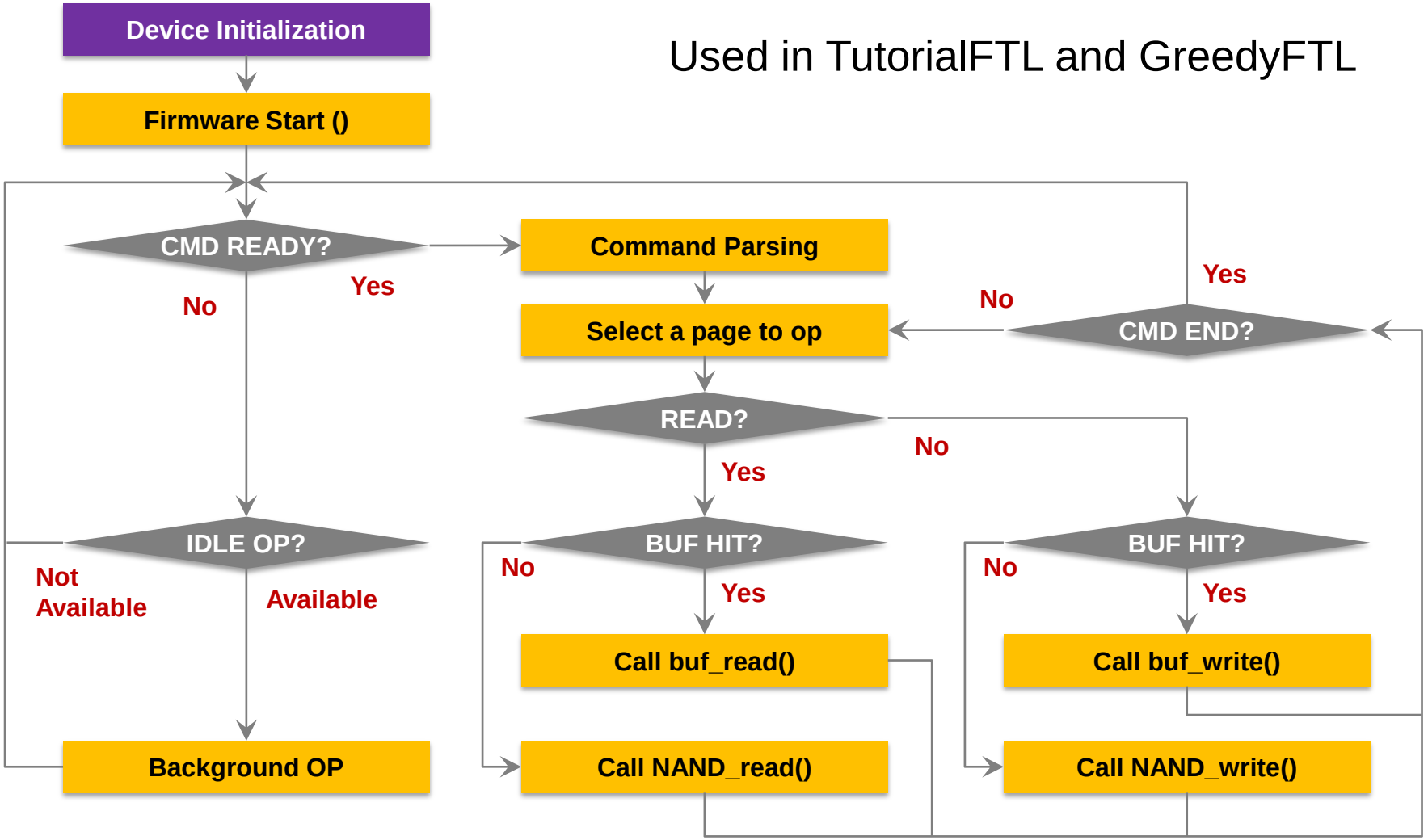
Available FTLs (under GPL)

- TutorialFTL (by HYU ENC Lab)
  - Page-mapping FTL w/o GC
  - Device initialization and interface definition
- GreedyFTL (by HYU ENC Lab)
  - Page-mapping FTL w/ Greedy GC

- DramDisk (by HYU ENC Lab)
  - DRAM Disk for measuring maximum read/write performance (PCIe + DRAM)

Common Operation Flow

Used in TutorialFTL and GreedyFTL



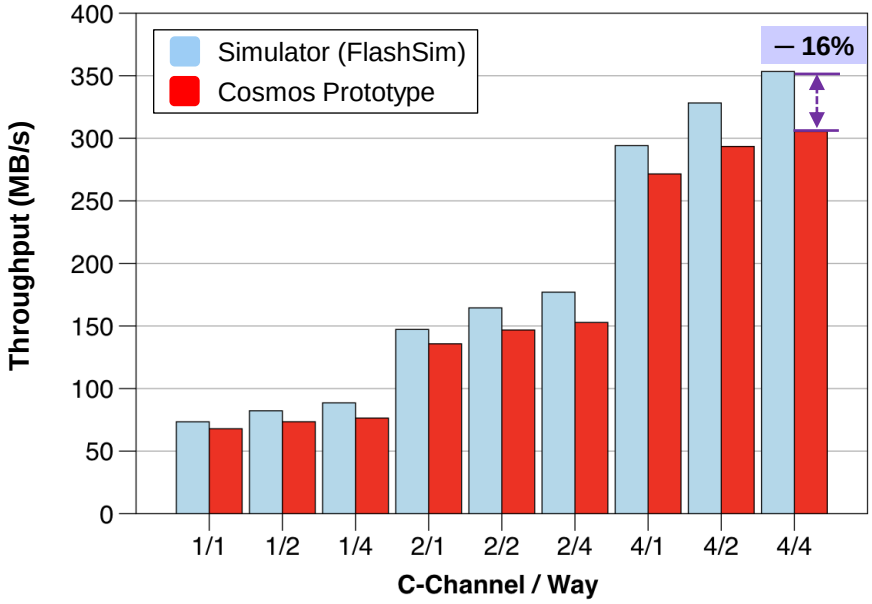
Host Software

- AS-IS
  - Makes the storage work as a drive
  - Handles read/write requests from kernel
  - Needs to install storage-specific device driver
    - Windows7 64-bit device driver (by HYU ENC Lab)
    - Linux 3.2.X kernel device driver (by HYU ENC Lab)
- TO-BE
  - Compatible with PCIe system drivers
    - AHCI support
    - NVMe support

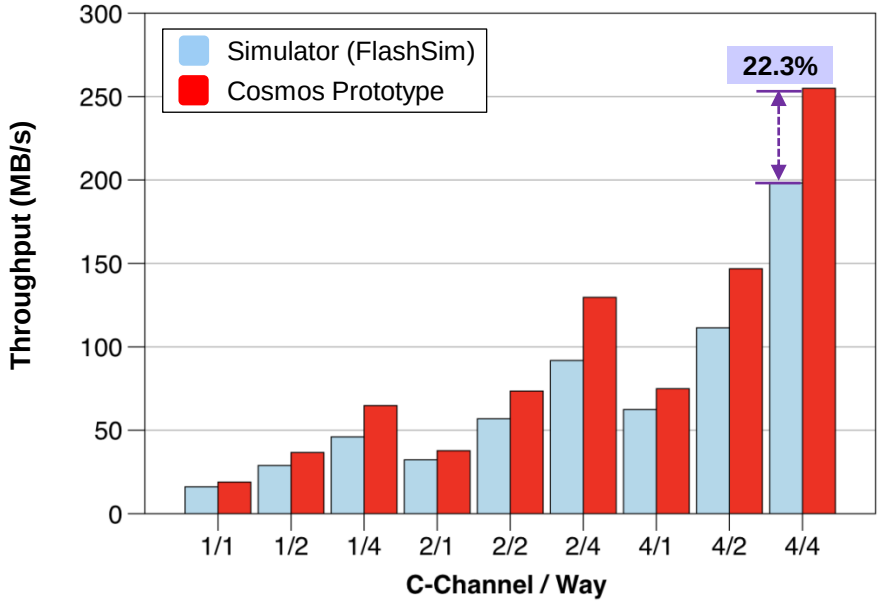
Some Use Cases

Cosmos Prototype (Tiger2) Architecture Exploration

Sequential Read



Sequential Program



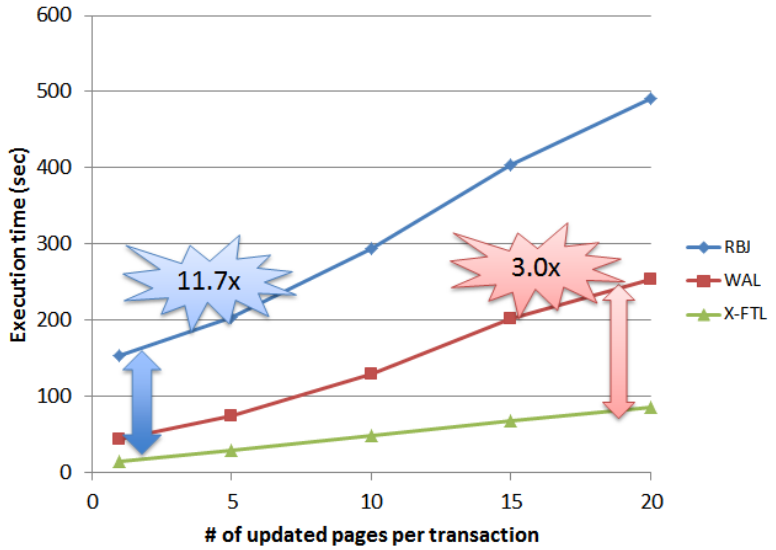
- When compared to the simulator, the performance of cosmos prototype is 16% lower and 22.3% higher in read and program operations, respectively

Note: Cosmos prototype is used in this experiment

C-Channel: Clustering-Channel

X-FTL

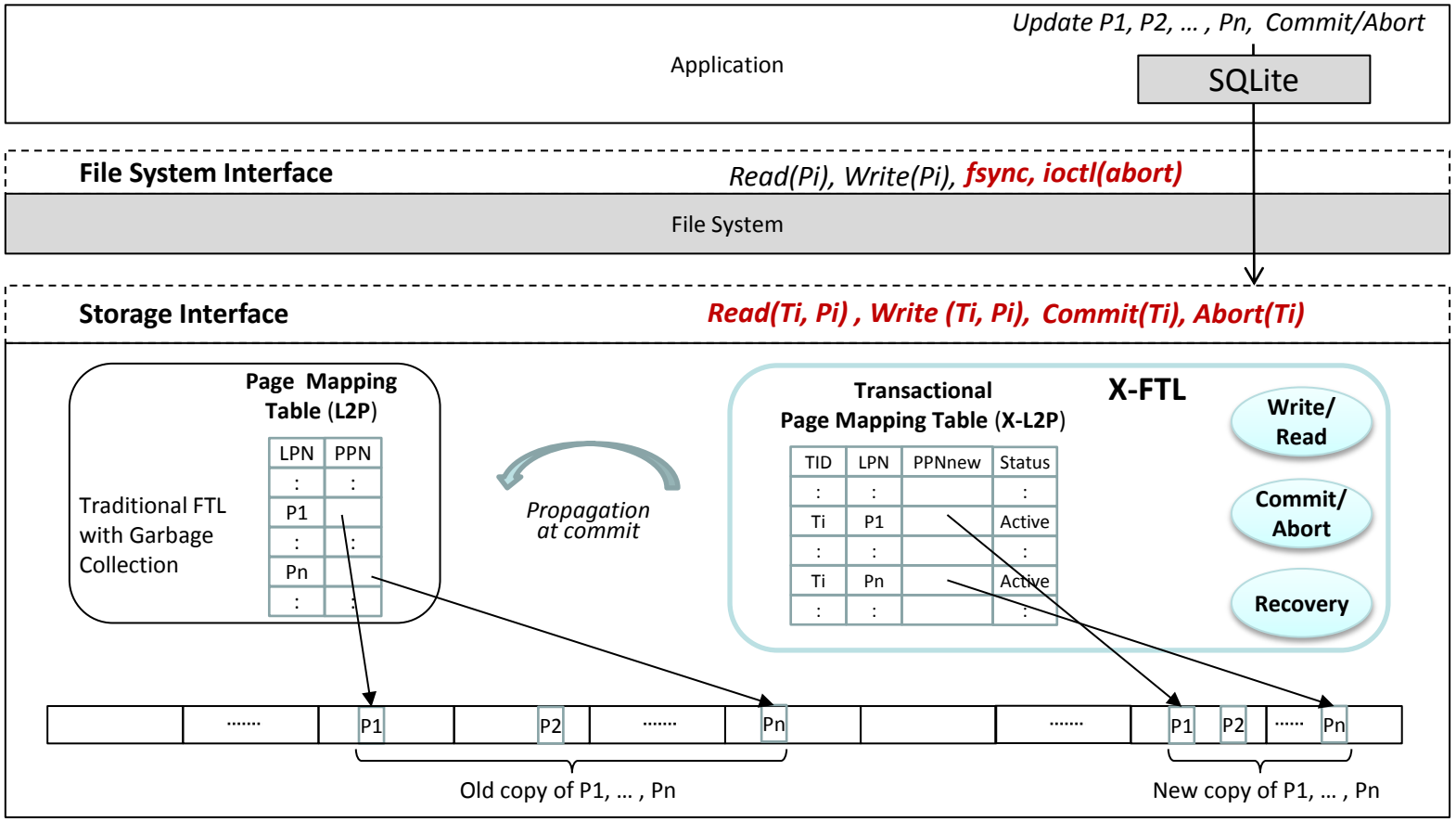
- Transactional FTL for SQLite (SIGMOD 2013)
  - Atomic write of data pages
    - Offload the semantic of all-or-nothing propagation of data pages carried out in host system down to FTL layer
    - Avoid “redundant writes”
  - Originally on top of Jasmine OpenSSD; now in porting to Cosmos



*Ref.) X-FTL: Transactional FTL for SQLite Databases

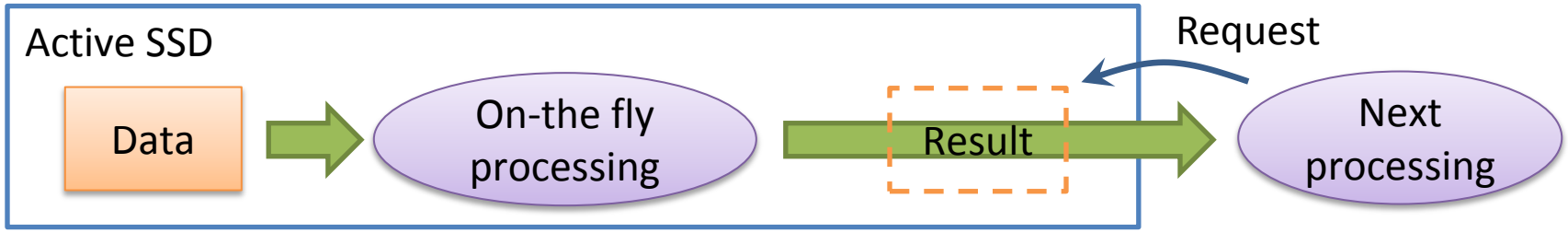
X-FTL (Continued)

- Architecture



ActiveSSD

- ActiveSort (HotStorage 2014)
  - Accelerates external sorting
  - Performs on-the-fly merge inside the SSD when the results are requested
  - Eliminates extra data transfer
  - Increases the lifetime of SSDs



*Ref.) Accelerating External Sorting via On-the-fly Data Merge in Active SSDs

Stay Tuned

- Sources will be available soon at the OpenSSD webpage
- Sources could be updated by other users except us (welcome all the time!)
- And more activities will be posted to the webpage as well

Call For Participation

- Welcome any contributions from
  - SSD manufacturers
  - NAND flash vendors
  - Research groups
  - Individual developers
  - …

Contributors

Prof. Yong Ho Song Ph.D.
Hanyang University

Prof. Sang-Won Lee Ph.D.
Sungkyunkwan Univ

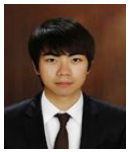
Prof. Jin-Soo Kim Ph.D.
Sungkyunkwan Univ

Jaehyeong Jeong Ph.D.
System Architecture
& SSD HW Controller

Sanghyuk Jung Ph.D.
System Architecture
& SSD SW Firmware

Youngin Jo
SSD Architecture
& PCIe Interface

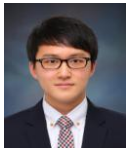
Youngnam Kim
SSD HW Architecture



Taeyeong Huh
SSD HW Controller

Sangjin Lee
Controller Architecture
& PCIe Interface

Ilyong Jung
Controller Architecture
& ECC Engine



Gyeongyong Lee
SSD SW Firmware

Luis Cavazos
Active SSD

Gi-Hwan Oh
xFTL Design

Young-Sik Lee
Active SSD

Sung-Rae Kim Ph.D.
ECC Algorithm

Thank you

For further information,
visit http://www.openssd-project.org