FlashAlloc on Cosmos+ OpenSSD

Quick Start Guide

Revision	Date	Name
v1.0	2023.09.01	Jonghyeok Park

1. Overview

FlashAlloc is a novel interface for flash storage, which is used to the logical address ranges of objects to the underlying flash device and thus to enlighten the storage device to **stream writes by objects**.
You can learn more details about FlashAlloc in our VLDB 2023 paper [1].

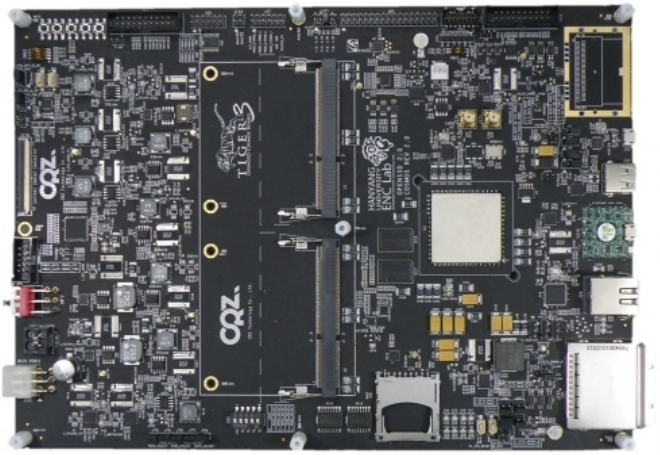
This quick guide document provides the information to build / run FlashAlloc on Cosmos+ OpenSSD platform.

1.1 Cosmos+ OpenSSD

To run the FlashAlloc command, you will need the Cosmos+ OpenSSD with hynix NAND firmware version.
You can get more accurate information at CRZ Technology [2].

- **Features**

FPGA	Xilinx XC7Z045-3FFG900 Zynq-7000
CPU	Dual ARM Cortex-A9 1GHz core Neon DSP co-processor for each core
SRAM	256 KB
DRAM	1GB DDR3
NAND	512 GB Hynix NAND
Ethernet	1 Gigabit Ethernet Interface
PCIe	Dual PCIe Gen2 x8 End-points (Cabled PCIe Interface)



- **Softwares**
  You also need software and tools to set up the Cosmos+ OpenSSD in your server.
  1) Xilinx ISE design suited 19.1 system edition [3]
  2) UART terminal emulator software (Xilinx software Development Kit SDK includes UART terminal) [4]
  3) USB-to-UART driver [5http://www.openssd-project.org/]

1.2 Tested environment

In our paper [1], we conducted a series of experiments using a server with specifications outlined in the table below.

CPU	Dual socket Intel Core i7-6700 at 3.40 GHz (8 core total)
DRAM	50GB
OS	Ubuntu 18.04 LTS
SSD	Cosmos+ OpenSSD 16GB

2. Project Structure

This section explains a brief introduction of the FlashAlloc project structure. You can learn more about it in our repository [6].

- **Flashalloc**: Cosmos+ OpenSSD firmware source code which implements the FlashAlloc interface.
- **Multistream**: Multi-stream SSD prototype for Cosmos+ OpenSSD supporting eight stream-ids like the commercial one. This prototype only provides physical streaming features without a garbage collection scheme.
- **host**: This directory includes implementation of both RocksDB and MySQL, utilized for evaluation in our paper. It also includes a simple host application example to test the FlashAlloc command.

2.Build

To run the FlashAlloc, you need to prepare Cosmos+ OpenSSD board. You can learn more detail instruction in this guide document [7]
In summary, please follow these steps.

1. Prepare Cosmos+ OpenSSD board, Window and Linux PC, respectively.
2. Connect the Cosmos+ OpenSSD to Window PC (JTAG digilent module and USB cable for UART), and install the PCIe to connect between Cosmos+ OpenSSD and Linux PC.
3. Open the FlashAlloc project in Xilinx SDK
4. Right click on the project, and Run As > 1. Launch on Hardware (GDB)
5. Click the firmware to execute, and click OK > wait UART message
6. Press X to make bad block table (We recommend you to this process for the first time)
7. Wait and turn-on (or reboot) the Linux PC
8. You can check the Cosmos+ OpenSSD using nvme-cli or lspcie command (check the nvmeXnXXX in the device list)

3. FlashAlloc command

To test FlashAlloc command, I will recommend playing with a simple host application in our repository.

```
git clone https://github.com/JonghyeokPark/Flashalloc-Cosmos.git
cd host
make -j

sudo ./playground /dev/nvmeXXX 0 32
```

- playground : host application name
- /dev/nvmeXXX : Cosmos+ OpenSSD device path
- 0 : start LBA
- 32 : LBA length

4. Running the Experiments

This document provides step-by-step guide to RocksDB, MySQL, and Multi-tenant experiments conducted in our paper.

1.1 RocksDB

- **Build**

1) Build the RocksDB database engine for oth flashalloc or multi-streame ssd version
2) You can also build the vanilla version of Rocksdb using `bash ./scripts/build.sh` command

```
cd host/RocksDB/rocksdb-{flashalloc | msssd}
bash ./scripts/build.sh cosmos
```

- **Configure File system**

1) You can mount either EXT4 or F2FS filesystem with TRIM support on top of Cosmos+ OpenSSD

```
# EXT4 Filesystem
bash ./scripts/setup-ext4-trim.sh

# F2FS Filesystem
bash ./scripts/setup-f2fs-trim.sh
```

- **Run `db_bench` with 4 tenants**

```
# FlashAlloc version
bash ./scripts/run_flashalloc.sh

# Vanilla version
bash ./scripts/run_vanilla.sh

# Multi-stream SSD version
bash ./scripts/run_msssd.sh
```

- **Run `db_bench` with single tenant**

```
# FlashAlloc version
bash ./scripts/falloc-single.sh

# Multi-stream SSD version
bash ./scripts/msssd-single.sh
```

1.2 MySQL

- **Build**

```
cd hsot/MySQL
bash ./build.sh
```

- **Initialze data directory**

```
bash init.sh
```

- **Run MySQL server**

```
bash ./run.sh
```

- **Run TPC-C Benchmark**

1) You can also modify the configuration for TPC-C Benchmark (e.g., # of clients, wrapup time, duration, and etc)
2) To run the either vanilla or multi-stream ssd version, you can execute `run.sh` script.

```
cd host/tpcc-mysql

bash ./run-flashalloc.sh
```

1.3 Multi-tenant

- In this experiment, you can evaluate with multi-tenant configuration.

```
# configure the filesystem and data directory for TPC-C and
db_bench

bash ./setup-multi.sh# run multi-tenant experiments

bash run-multi.sh
```

5. Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIT) (No.2022R1A2C2008225, No.RS-2023-00251665), Samsung Research Funding & Incubation Center of Samsung Electronics under Project Number SRFC-IT1802-07, Hankuk University of Foreign Studies Research Fund of 2023, and Samsung Electronic

6. References

[1] Jonghyeok Park, Soyee Choi, Gihwan Oh, Soojun Im, Moon-Wook Oh, Sang-Won Lee, “FlashAlloc: Dedicating Flash Blocks By Objects”. VLDB 2023.
[2] “Crz Technology”,https://www.crz-tech.com/crz/article/CosmosPlus/
[3] “Xilinx SDK Downlods” , https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools/archive.html
[4] “CP210x USB to UART Bridge VCP Drivers”, https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers
[5] “The OpenSSD Project”, http://www.openssd-project.org/
[6] “Flashalloc-Cosmos”, https://github.com/JonghyeokPark/Flashalloc-Cosmos
[7] “Cosmos+OpenSSD-2017-Tutorials”, https://github.com/JonghyeokPark/Flashalloc-Cosmos/blob/master/doc/Cosmos+OpenSSD-2017-Tutorial.pdf