Key Value SSD: a Scalable Smart Storage for Objects

Yang Seok Ki
Director/Principal Engineer
Memory Solutions Lab
Samsung Electronics
This presentation and/or accompanying oral statements by Samsung representatives collectively, the “Presentation”) is intended to provide information concerning the SSD and memory industry and Samsung Electronics Co., Ltd. and certain affiliates (collectively, “Samsung”). While Samsung strives to provide information that is accurate and up-to-date, this Presentation may nonetheless contain inaccuracies or omissions. As a consequence, Samsung does not in any way guarantee the accuracy or completeness of the information provided in this Presentation.

This Presentation may include forward-looking statements, including, but not limited to, statements about any matter that is not a historical fact; statements regarding Samsung’s intentions, beliefs or current expectations concerning, among other things, market prospects, technological developments, growth, strategies, and the industry in which Samsung operates; and statements regarding products or features that are still in development. By their nature, forward-looking statements involve risks and uncertainties, because they relate to events and depend on circumstances that may or may not occur in the future. Samsung cautions you that forward looking statements are not guarantees of future performance and that the actual developments of Samsung, the market, or industry in which Samsung operates may differ materially from those made or suggested by the forward-looking statements in this Presentation. In addition, even if such forward-looking statements are shown to be accurate, those developments may not be indicative of developments in future periods.
Agenda

• Background
• Use Cases & Challenges
• Key Value SSD
• Experiments
• Conclusion
BC/AD in IT

Source: Human Computer Interaction & Knowledge Discovery

Structured Data

Unstructured Data

Before Cloud

Anno Datum

IDC and EMC project that data will grow to 40 ZB by 2020.
Everything is object!

OSD Object Storage
- ID
- Attributes
- User Data

Key Value Storage
- Key
- Value
Key Value Stores in Systems at Scale
Case 1: Abstraction of Scalable Store in DC

Applications

Scalable Storage (Block/Object)

KV Store

File System

Applications

Scalable Database (Redis, MongoDB)

KV Store

File System
RocksDB: Key Value Database

- Application database
- Write-optimized storage
**Performance Implications of IO Amplification**

Ceph performance on RocksDB with an SSD in the case of IO saturation

<table>
<thead>
<tr>
<th>Block Size</th>
<th>Data Written</th>
<th>WRITE</th>
<th>READ</th>
<th>WAF</th>
<th>RAF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>IOPS</td>
<td>BW (MBps)</td>
<td>IOPS</td>
<td>BW (MBps)</td>
</tr>
<tr>
<td>4K</td>
<td>47.01 GB</td>
<td>2416</td>
<td>9.43</td>
<td>13733</td>
<td>53.64</td>
</tr>
</tbody>
</table>

- Device Throughput from Ceph: 9.43 x 12.05 = 113.63 MB/s
- Device Throughput with 4KB blocks from FIO: 117.16MB/s
Case 2: Extent Service for Scalable Storage

- Extent store or chunk store uses a key value store to map a logical block address space to a physical block address space.
Extent Store Use Case

- Storage capacity tends to be limited by DRAM capacity of server to manage a map between logical address space and physical address space
System Resource

- Return the DRAM space for mapping in the host to applications
- Return the user storage space for GC to applications (e.g., 5% user OP)
Key Value SW Stack Consolidation

- SSD with native key value interface through hardware software co-design

**Datacenter S/W Infra**
- Storage Plugin Interface
- Key Value Glue Logic
- Key Value API
  - Index
  - S/W Key Value Store
  - Log
  - POSIX API
  - Block Map
  - File System
  - Journal
- Block Interface
- Block Device Driver
- Command Protocol
  - Map
  - Block Device
  - Log

**Datacenter S/W Infra**
- Storage Plugin Interface
- Key Value Glue Logic
- Key Value API
  - Thin KV Library
- KV Device Driver
- Command Protocol
  - Index
  - KV Device
  - Log

**TX/s**
**WAF, RAF, Latency**
KV SSD Design Overview

- **Key/Value Range**
  - Key: 4~255B
  - Value: 64B~2GB (32B granularity)
  - The large value is stored into multiple NAND pages
Samsung KV-PM983 Prototype

NGSFF KV SSD

Form factor: NGSFF/U.2
Capacity: 1-16TB
Interface: NVMe, PCIe Gen.3
NVMe Extension for Key Value SSD

- Defines a new device type for a Key Value device
- A controller performs either KV or traditional block storage commands

### New Key Value Commands
- PUT
- GET
- DELETE
- EXISTS

### Existing Command Extension
- Admin command
- Identify commands for KV
- Other non-block specific commands
KV SSD Ecosystem

- Partners
- Standard
- Product
- SDK
- Applications

Key Value SSD

(nvm EXPRESS™)

(Linux, Windows)

(SNIA)

(RocksDB, MongoDB, Redis, ceph)

(SAMSUNG)

(COLLABORATE. INNOVATE. GROW.)
Scale-Out: RocksDB & KV Stacks Configuration

Client: kvbench

RocksDB
XFS
Page Cache
RAID0
Block Driver

KV Stacks
ADI + KV User Driver

Client: kvbench

RocksDB
XFS
Page Cache
RAID0
Block Driver

KV Stacks
ADI + KV User Driver

Client: kvbench

RocksDB
XFS
Page Cache
RAID0
Block Driver

KV Stacks
ADI + KV User Driver

Client: kvbench

RocksDB
XFS
Page Cache
RAID0
Block Driver

KV Stacks
ADI + KV User Driver

RocksDB vs KV Stacks

NVMeoF over RDMA

Mission Peak
KV-PM983 SSDs

Software CentOS 7.3 w/ KV SW
NIC 2x 100GbE
CPU Xeon E5-2699V4 CPU @2.20GHz
1-node 2-socket
44-core 88-thread
DRAM 256 GB

Client: kvbench

Xeon 8160 CPU @2.10GHz
1-node 2-socket, 48-core 96-thread
36x 1TB SSDs

CPU

Software CentOS 7.3 w/ KV Target
NICs 2x 100GbE + 2x 50GbE
Performance: Random PUT

- 8x more QPS (Query Per Second) with KV Stacks than RocksDB on block SSD
- 90+% less traffic goes from host to device with KV SSD than RocksDB on block device

* Workload: 100% random put, 16 byte keys of random uniform distribution, 4KB-fixed values on single PM983 and KV-PM983 in a clean state
Scale-up Performance: Sequential Key PUT

- **3.4x** IO performance over S/W key value store on block devices

Relative performance to the maximum aggregate RocksDB random Put QPS for 1 SSD with a default configuration for 1 PM983 SSD in a clean state.

System: Ubuntu 16.04.2 LTS, Ext4, RAID0 for block SSDs, Actual CPU utilization could be 90% at CPU saturation point.

Workload: 100% puts, 16 byte keys of random uniform distribution for RocksDB v. 5.0.2, 4KB-fixed values, 36 RocksDB instances with 1 client thread, 34GB/Instance or 1.2TB Data is used.
Local vs NVMeoF PUT Latency

Average Latency

@Qdepth: 1-8
Overhead: 4-7us
CPU Utilization for Clients

Fill Random
Avg 170K QPS@72% CPU

Fill Sequential
Avg 400K QPS@80% CPU

KV Stacks
Avg 2.1M QPS@30% CPU

2.1 M QPS
Key Value SSD is a Scalable Solution with Better TCO

- **Scale-Up**
  - Performance
  - Capability

- **Scale-In**
  - CPU
  - Server

- **Scale-Down**
  - TCO
  - Power

- **Scale-Out**
  - Capacity
  - Performance

**KV SSD**
Questions?

kvssd@ssi.samsung.com