A New Key-value Data Store For Heterogeneous Storage Architecture

Intel APAC R&D Ltd.
Agenda

- Introduction
- Background and Motivation
- Hybrid Key-Value Data Store Architecture
  - Overview
  - Design details
  - Performance overview
- Summary
Introduction

- Intel® Cloud and Bigdata Engineering Team
- Deliver optimized open source cloud and Bigdata solutions on Intel® platforms
  - Open source leadership @Spark*, Hadoop*, OpenStack*, Ceph* etc.
- Working closely with community and end customers
- Bridging advanced research and real-world applications
Non-volatile memory (NVM) provides almost comparable low latency to DRAM and byte-addressability
Characteristic comparison chart

- **Performance**: ~1000x speed-up over NAND, closer to DRAM
- **Cost**: New programming model
- **Capacity**: Large capacity (TBs) compared with DRAM
- **Maturity**: Persistency and uncertain endurance
- **Durability**: 2LM mode, Block mode vs. PM mode
- **Compatibility**
Typical NVDIMM Software Architecture and NVM Library

libpmemobj
libpmemblk
libpmemlog
libpmem
...
Motivation

- With the growing maturity of NVM technology which both hardware and software library support.
- NVM must became a new main addition to storage hierarchy in the future.
- Need re-think storage engine design on how to utilize NVM’s benefits in cloud storage scenarios
Ceph*: OpenStack* de fecto storage backend

Ceph* is an open-source, massively scalable, software-defined storage system that provides object, block and file system storage in a single platform. It runs on commodity hardware—saving you costs and giving you flexibility—and because it’s in the Linux* kernel, it’s easy to consume.

- **Object store (RADOSGW)**
  - A bucket-based REST gateway
  - Compatible with S3 and swift
- **File system (CEPHFS)**
  - A POSIX-compliant distributed file system
  - Kernel client and FUSE
- **Block device service (RBD)**
  - OpenStack* native support
  - Kernel client and QEMU*/KVM driver
BlueStore: a new, faster storage backend for Ceph

- Ceph Cluster at least consists of two types of daemons
  - Monitor and OSD
- OSD support several types store backend
  - FileStore, MemStore, BlueStore
- BlueStore is the newest
  - consume raw block device(s)
  - key/value database (RocksDB) for metadata
  - data written directly to block device
Hyper Converged Cache Architecture

Building a hyper-converged cache solutions for the cloud

- Providing cache for different APIs
  - Block, Object, File cache
- Supporting various cloud storage solutions
  - Ceph, GlusterFS, Swift
- Advanced service
  - de-duplication, compression, QoS, snapshot
- Connecting with different backend:
  - Private cloud as a warm tier
  - Public cloud as cold tier
Hyper Converged Cache: Design Details

Block cache detail

- **Hyper-converged deployment**
- Also, support deduped read cache and persistent write cache for VM scenario.
I/O Characteristic for Read Cache

Two types of KV pairs:
- LBA -> Footprint
- Footprint -> 4K aligned Data
Design Goal

- Simple & General
- Large number of fixed length random write/read
- Full play to NVM performance advantages

![Diagram showing storage service and data flow between SSD/HDD, File System, Traditional Key-Value DB, and Hybrid-DS with NVM and SSD.](image-url)
Hybrid-DS Architecture Overview

- Build a prototype for a typical hybrid hardware environment
  - DRAM, NVM, SSD
- Persistent storage devices work together as a storage tier
- Semantic abstraction of the main data area,
  - Space provisioning & recycling
  - Data placement
Hybrid-DS Architecture

Write

- Aggregate KVSlices to write to a segment
- Write segment to device with Log appending
- Delete operation is equivalent to inserting a KVSlice with a value of null
Accelerated read with key-value LRU cache

The value is obtained by a hash calculation and a read from the disk.
Hybrid-DS Architecture
Space recycling

- Semantic abstraction of space recycling.
  - Migration for NVM
  - Garbage Collection for SSD
Hybrid-DS Design Details
In-Memory Indexing

- **RMD160 (Key) -> Digest**
  Notes: use RIPEMD-160 to generate key digest. The actual probability of collision is infinitesimally small or non-existent in practice for billions of keys.

- **Hash(Digest)**
  - Separate-chaining with linked lists

- **Write amplification**
  - trade-off between performance and space
Hybrid-DS Design Details
Space Provisioning

- Introduce Segment Stat Table (TBD)
Hybrid-DS Design Details

Data Placement

- Semantic abstraction of data placement
  - Normal data layout for NVM (TBD)
  - Optimized data layout adapt to SSD friendly IO pattern
    - Store fixed-length value with page align
Hybrid-DS Design Details

Garbage Collection

- Semantic abstraction of space recycling
  - Pick strategy base on timestamp for NVM (TBD)
  - Pick strategy base on valid space capacity for SSD
    - Always pick segments which the most worthy of recycling until aggregate to a full segment
Hybrid-DS Design Details
Fast Recovery

☐ TBD
Independent cache gateway

- **Key-Value Store**
  - Work as a library

- **Key-Value Cache Gateway**
  - Work as a service
  - Distribute Module provide general functions like network communication, replication logic etc. Make Hybrid-DS become a gateway.
  - Cache Policy Plugin provide general cache logic like flush, evict. Make Hybrid-DS become a cache. Due to is a plugin, Hybrid-DS degenerate a distribute key-value store when disable it.
  - Hybrid-DS can access variety of existing storage systems as a cloud data store backend.
Performance Overview

- TBD
Summary

- TBD
Backup
3D XPoint™ Technology

3D XPoint™
Latency: ~100X
Size of Data: ~1,000X

NAND
Latency: ~100,000X
Size of Data: ~1,000X

HDD
Latency: ~10 MillionX
Size of Data: ~10,000 X

SRAM
Latency: 1X
Size of Data: 1X

DRAM
Latency: ~10X
Size of Data: ~100X

MEMORY

Performance numbers are Intel Internal estimates
For more complete information about performance and benchmark results, visit www.intel.com/benchmarks
Intel and Intel logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries
Technology claims are based on comparisons of latency, density and write cycling metrics amongst memory technologies recorded on published specifications of in-market memory products against internal Intel specifications.

Performance numbers are Intel Internal estimates

For more complete information about performance and benchmark results, visit www.intel.com/benchmarks

Intel and Intel logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries


Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance, Server Configuration: 2x Intel® Xeon® E5 2690 v3 NVM Express* (NVMe) NAND based SSD: Intel P3700 800 GB, 3D Xpoint based SSD: Optane NVMe OS: Red Hat® 7.1
Intel® Optane™ shows significant performance improvement over PCIe SSD for RocksDB* Key/Value cloud benchmark*

Performance numbers are Intel internal estimates.
For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.
Intel and Intel logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

*Benchmarked on early prototype samples, 2S Haswell/Broadwell Xeon platform single server. Data produced without any tuning. We expect performance to improve with tuning.
*Other names and brands may be claimed as the property of others.