White Paper

Intel Helps iQIYI Optimize the Performance of MySQL Database Engines to Improve Customer Experience

iQIYI has always been pioneering innovations in both content and technology. New technologies play a significant role in everything from identifying the applications and challenges to introducing new hardware and implementing it in actual use cases. iQIYI has recently introduced Intel® Optane™ persistent memory, which is an important innovation in the hardware architecture for databases. It can accommodate more service requests with improved performance, delivering the ultimate technical experience to users.

— Li Sun, iQIYI Scientist

Contents

Introduction

According to the "China Internet Development Report (2021)", in 2020, China's online video market size hit 241.2 billion CNY, an increase of 44% from the previous year, and the number of active online video users in the country reached 1,001 million, an increase of 2.14%[1] from previous year. The huge user base creates great business opportunities in China's online video market.

As an innovative company with both technology and entertainment genes, iQIYI leverages innovative technologies to empower entertainment and reduce entertainment costs. The company strives to provide a content experience that features vitality and positivity and can fuel the dreams of its users.

iQIYI is a leading online video portal in China, and its huge number of users poses challenges to its database system. In high-QPS and low-latency application services, the servers often incur I/O bottlenecks, resulting in a significant drop in MySQL performance and compromising the customer experience. iQIYI partnered with Intel and adopted their large-capacity, low-latency Intel® Optane™ persistent memory and Storage Performance Development Kit (SPDK), which addressed the increased latency and reduced performance of MySQL for high read I/O scenarios and improved the user experience.

Challenge

MySQL has long been a core component of various software systems, and is used for the running of a large number of applications, middleware, and management tools. Originally designed as a database based on disk storage, MySQL has always been plagued by high latency in some high-QPS use cases. This represents a performance bottleneck, which is hard to eliminate, even by using SSDs. For example, let's look at iQIYI's MySQL database, which adopts InnoDB as the storage engine.

[1] Cited from "China Internet Development Report (2021)"

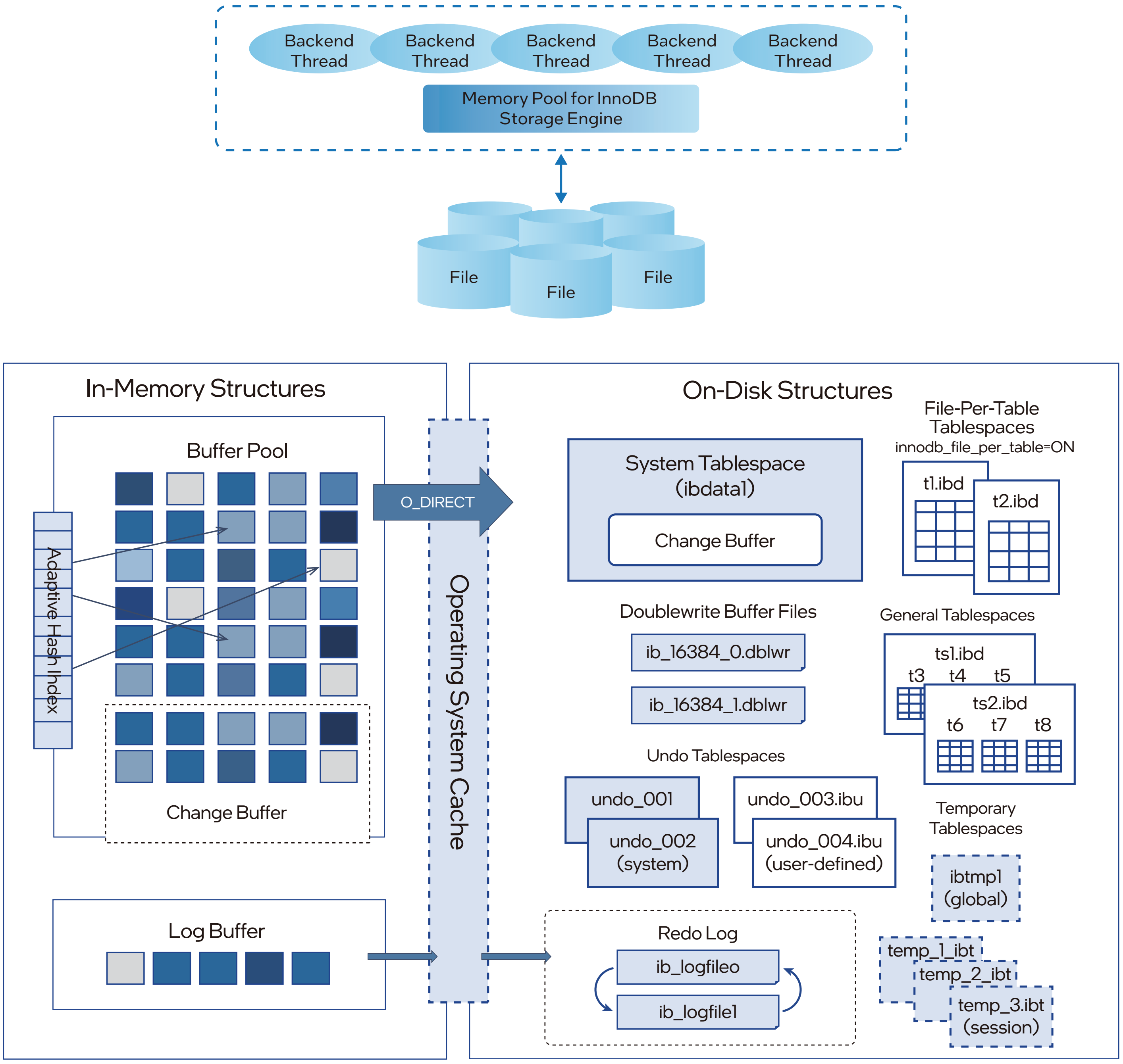


Figure 1. MySQL InnoDB engine currently in use

InnoDB implements two file read/write methods: synchronous I/O and asynchronous I/O. For read operations, data requests triggered by user threads are usually synchronous. When a user thread executes an SQL statement, if the requested data page is not in the Buffer pool, it is required to load the data page in the file into the Buffer pool. At this time, if there is a bottleneck in I/O and a delayed response, the thread will be blocked. The above problem can result in slow SQL execution. If the problem worsens and a large number of slow queries appear, the running thread will be in a wait state, occupying the InnoDB thread. For a system with high concurrency, this problem will cause a large number of other threads to be waiting in the queue. The high number of concurrent threads will cause frequent context switching, which increases CPU usage and greatly reduces the performance.

There are two methods to solve the problem. First, you can divide the database table into multiple partitions, put each partition in a MySQL instance, and allocate to it with CPU, memory, and storage resources to reduce read pressures. Second, you may try to utilize the memory resources of the server as much as possible, and to expand the Innodb_buffer_pool_size buffer to deliver the required performance. However, this buffer cannot be expanded indefinitely as it is limited by the memory space.

Each method requires more memory resources; therefore, a new solution is needed to solve the latency problem and improve the performance of MySQL.

Solution

iQIYI's database team used to adopt SSDs as the storage device to solve the troubles caused to MySQL by I/O bottlenecks. This, however, did not offer an effective solution to the latency problem. Because the database server implements both sequential and random IO concurrent, and most of the hotspot data is scattered randomly, which must be buffered in memory to eliminate I/O bottlenecks.

With this in mind, iQIYI worked with Intel to adopt Intel® Optane™ persistent memory as the storage medium for the Buffer pool. As shown in Figure 2, the optimized database engine solution has two parts. In the one part, the memory carries the index data. In the other part, the persistent memory stores massive data as a cache and provides a place for the read data to interact with the memory. This helps to save I/O overhead, thereby greatly improving the performance of MySQL.

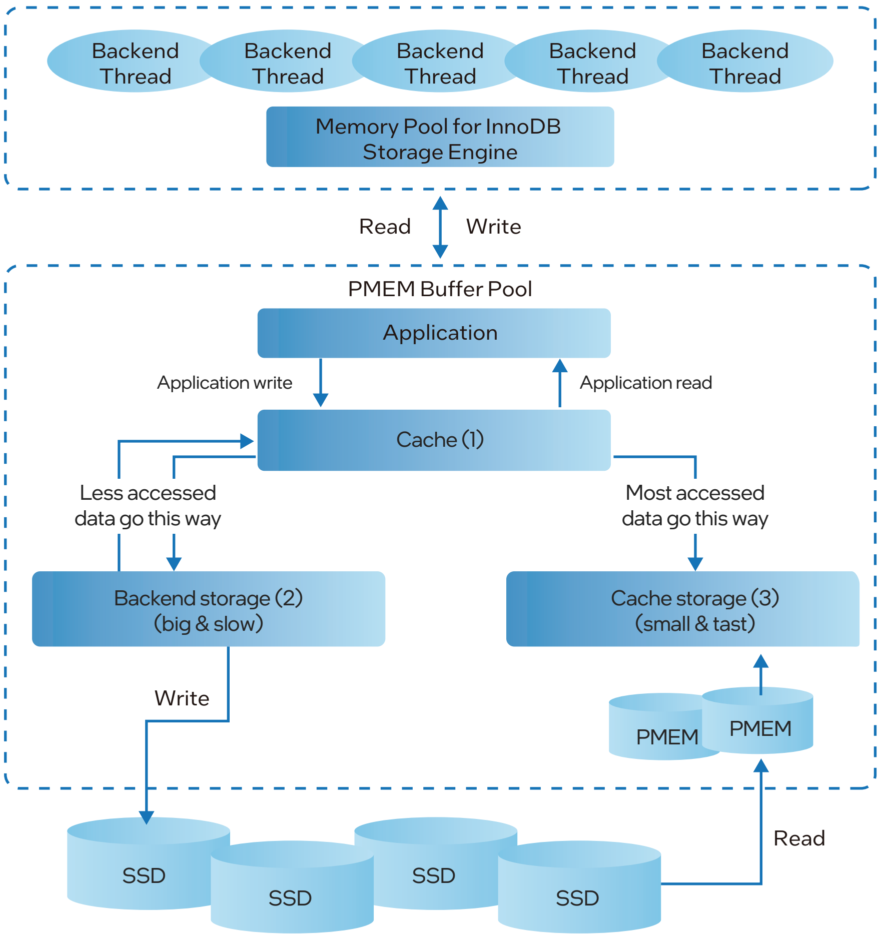


Figure 2. Optimized solution

Intel® Optane™ Persistent Memory

Intel® Optane™ is a revolutionary memory product based on 3D XPoint media. It offers several advantages including high speed, low latency, large capacity, high cost efficiency, persistent data retention, and advanced encryption etc.

Persistent memory changes the original storage hierarchy (see Figure 3) to provide similar performance and the same access method compare with DDR RAM ("DRAM"), but as well as non-volatile persistent storage data like SSDs. In addition, persistent memory offers a larger capacity and is cheaper than DRAM. Figure 4 shows a comparison of capacity, price, and performance in the new storage hierarchy. Its unique combination of high performance, large capacity, and cost efficiency helps users reduce the total cost of ownership with little impact on the system performance.

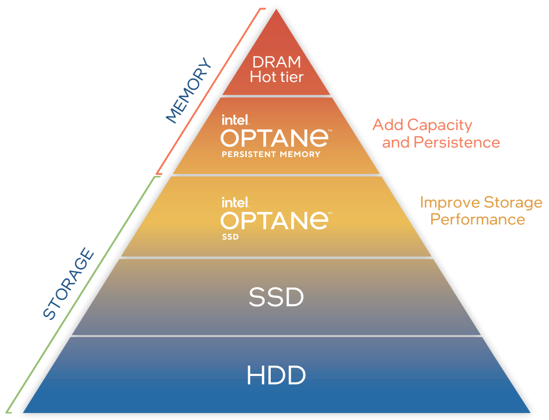

MEMORY
STORAGE
DRAM Hot tier
intel OPTANE PERSISTENT MEMORY
intel OPTANE SSD
SSD
HDD
Add Capacity and Persistence
Improve Storage Performance

Figure 3. In the storage hierarchy, Intel® Optane™ persistent memory is below DRAM

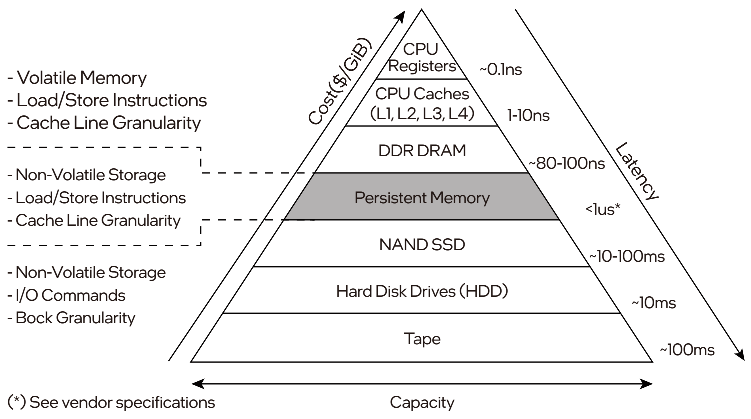


Figure 4. Capacity, price, and performance of the storage hierarchy

Intel® Optane™ persistent memory is byte-addressable. Traditional databases store data primarily on disks, which, however, are not addressable by bytes, so they usually read data by blocks. This is how the read data process going: First, the entire block of data is read to the memory, and then the specific data is read based on the offset in a byte-addressing manner. Persistent memory is byte-addressable, which dramatically simplifies lower-level I/O for quick queries.

Intel® Optane™ supports two modes: Memory Mode and App Direct Mode ("AD Mode").

In Memory Mode, the CPU memory controller sees persistent memory as volatile system memory and DRAM as a high-speed cache for persistent memory. This mode offers a larger memory capacity. However, for any data request submitted in this mode, the CPU memory controller will first attempt to retrieve data from the DRAM cache. When the data is present, it will return the request from the DRAM cache. When the data is not present, the request will be sent to persistent memory.

In AD Mode, software and applications that support the industry standard NVM persistent memory programming model can communicate directly with Intel® Optane™ persistent memory and take full advantage of its byte addressability to access smaller block files than the file system. such as 64-byte, 128-byte, and 256-byte files. Therefore, this mode can significantly reduce latency and meet the requirements of high-speed use cases. Take full advantage of persistent memory large capacity compare with DRAM, then store most of data in the large-capacity persistent memory and indexes are stored in the DRAM cache. This ensures that most keyword queries are performed in the faster DRAM cache, allowing the DRAM cache and persistent memory to complement each other with their advantages.

Unlocking the Potential of MySQL with SPDK

To further tap the potential of MySQL, iQIYI used Intel® SPDK to optimize its MySQL. The SPDK provides a set of tools and libraries for writing high performance, scalable, user-mode storage applications. It achieves high performance through the use of a number of key techniques:

- Moving necessary drivers into the user space, which avoids system calls and enables zero-copy access from the application
- For high-performance use cases, polling hardware for completions instead of relying on interrupts, which lowers both total latency and latency variance.
- Avoiding all locks in the I/O path, instead relying on message passing.
- Linearly scaling with end-to-end storage capability as the number of CPU cores increases.
- Efficiently supporting the latest platforms and hardware such as storages and network offload engines to provide holistic storage optimization.

The bedrock of SPDK is the user space, polled-mode, asynchronous, lockless application acceleration framework, block layer, and NVMe driver. This provides zero-copy, highly parallel access directly to a local NVMe SSD from a user space application, while allowing for efficient access to back-end storage from NVMe-oF.

SPDK further provides a full block stack as a user space library that performs many of the same operations as a block stack in an operating system. This includes unifying the interface between disparate storage devices, queueing to handle conditions such as out of memory or I/O hangs as well as rich storage services, such as logical volume management, compression encryption, OCF cache, and so on.

Finally, SPDK provides storage-related protocols and APIs such as NVMe-oF, iSCSI, vhost-user, BlobFS, and Blobstore to move closer to different storage services for applications. Servers built on top of these components are capable of providing high-performance, scalable storage services over the network or to other application processes. These optimized servers can be up to an order of magnitude more CPU efficient and deliver higher performance than other kernel-based implementations.

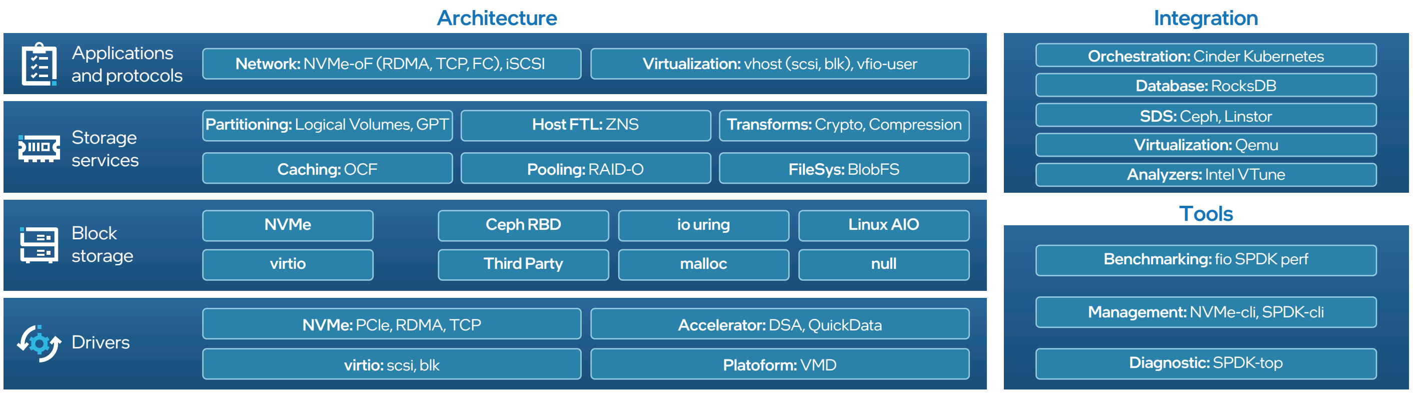


Figure 5. SPDK architecture

OCF

As users are increasingly demanding on storage speed, many high-speed and ultra-high-speed storage devices have been brought to the market, such as the persistent memory and Optane NVMe SSD powered by Intel® Optane™ technology. Today, there are still large numbers of SATA SSDs and even HDDs in data centers. The SPDK 19.01 release introduces a new user-mode bdev module called Open CAS Framework (OCF). It takes advantage of high-speed media (Persistent Memory, Optane SSD, NVMe SSD, etc.) to provide cache services for other existing bdev block devices to increase throughput and reduce latency.

OCF is the open source version of Intel® Cache Acceleration Software (Intel® CAS). Before we dive into OCF, let's talk about CAS first. CAS is an enterprise-class software solution designed to accelerate storage performance by I/O classification and caching frequently used data on high-speed media. It "fuses" a high-speed device and a low-speed device into one device, and mounts the device to the upper-layer service just as it does for other devices. CAS interoperates with memory to create a multilevel cache that optimizes the use of system memory and automatically determines the best cache level depending on the specifics of the data itself.

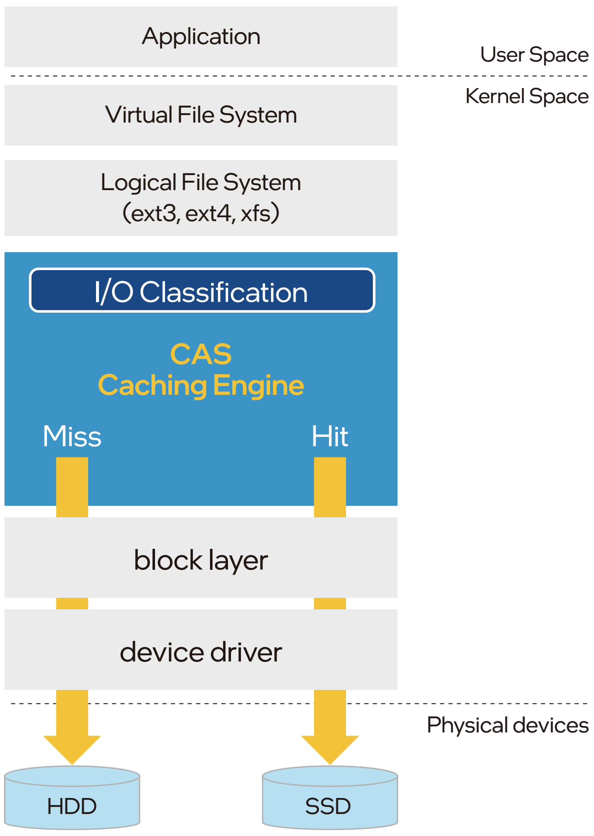


Figure 6

Beginning with the SPDK 19.01 release, OCF has been supported as a virtual bdev in the SPDK general bdev layer to provide block-level cache acceleration. OCF can easily interoperate with upper-layer storage services and applications, while providing several cache modes, including Write Back, Write Through, Write-Around, and more.

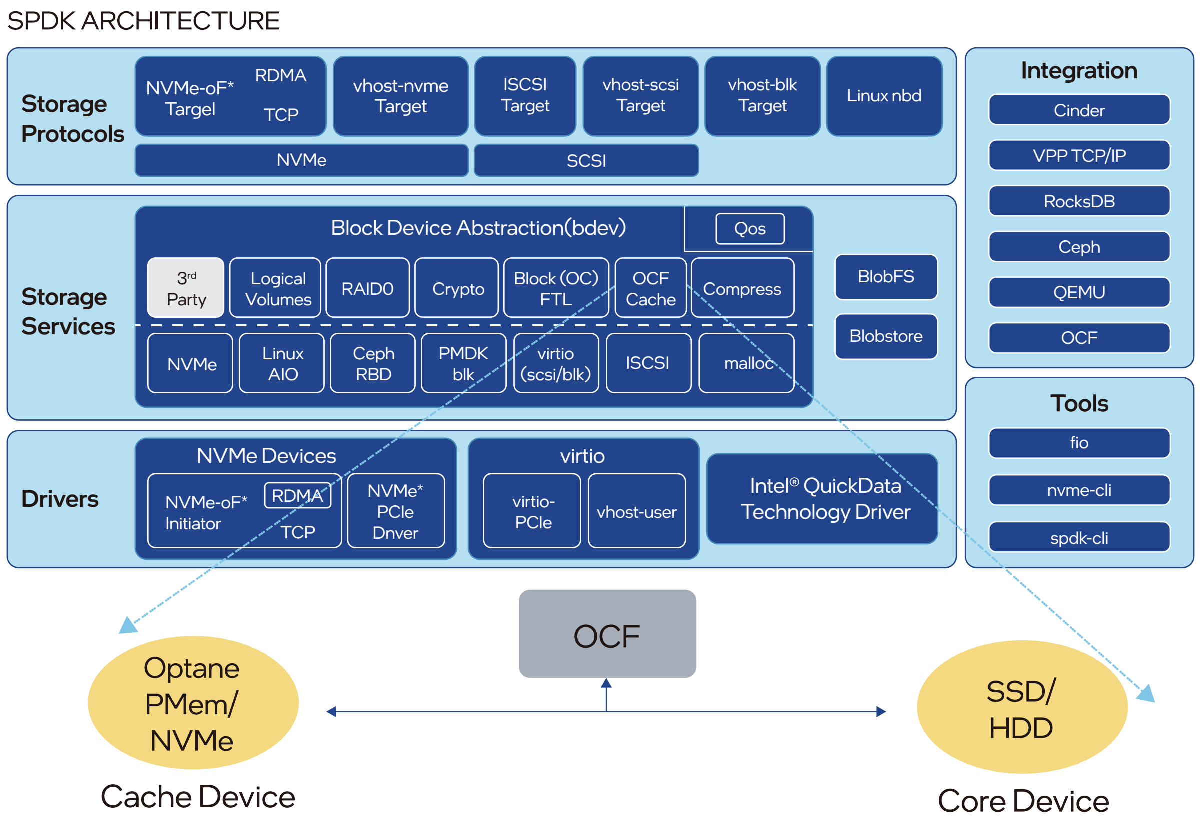


Figure 7. In SPDK, OCF serves as a new virtual bdev.
Other bdevs can be used as the core device and cache device.

iQIYI's database team used the OCF module to build a user-mode-based cache engine for underlying devices, providing cache services for existing bdev devices. SPDK moves necessary drivers into the user space, which avoids system calls and enables zero-copy access from the application. For highly-concurrent access, it polls hardware for completions instead of relying on interrupts, which lowers both total latency and latency variance. Furthermore, SPDK performs much better than kernel drivers in terms of IOPS for each CPU core. Additionally, SPDK has a lockless high-performance I/O path mode that avoids locks in key I/O paths, and instead relies on message passing to share resources among multiple threads to achieve higher concurrency performance.

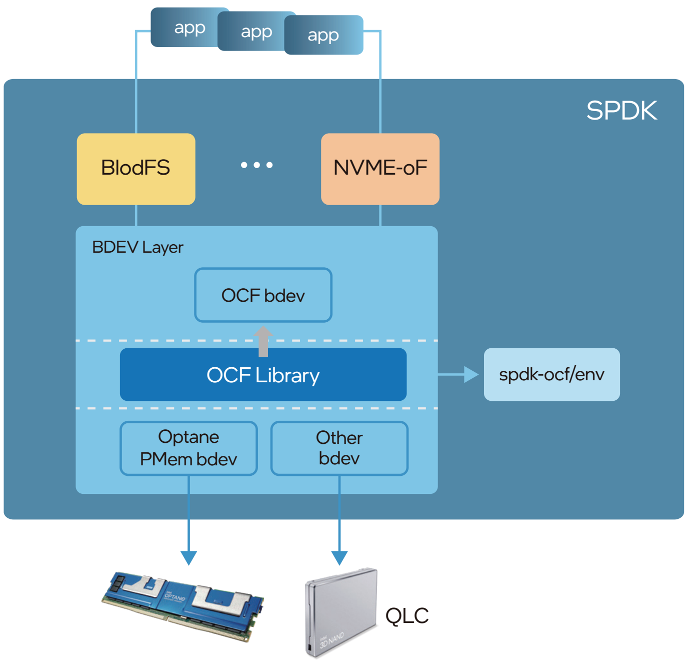


Figure 8. SPDK and OCF architectures

Performance Verification

To verify the read and write performance of the new MySQL engine that adopts Intel® Optane™ persistent memory as the Buffer pool under the same service architecture, iQIYI's database team worked with Intel to test the average response time of I/O requests, throughput, and IOPS.

Hardware Requirements:

CPU that supports Intel® Optane™ persistent memory

Cache: Intel® Optane™ persistent memory (PMEM 100 series 128 GB used in this test)

Specifications:

CPU:	INTEL 5218 and above
Memory:	192GB and above
Persistent Memory:	128GB and above
Hard Disk:	SATA SSD or NVMe SSD
NIC:	25GB minimum

Memory Requirements:

Symmetric Population within the Socket													
	iMC1						iMC0						
	Channel 2		Channel 1		Channel 0		Channel 2		Channel 1		Channel 0		
Modes	**Slot 1**	**Slot 0**	**Slot 1**	**Slot 0**	**Slot 1**	**Slot 0**	**Slot 1**	**Slot 0**	**Slot 1**	**Slot 0**	**Slot 1**	**Slot 0**	
AD	DCPMM	DRAM1	DCPMM	DRAM1	DCPMM	DRAM1	DCPMM	DRAM1	DCPMM	DRAM1	DCPMM	DRAM1	2-2-2
MM	DCPMM	DRAM1	DCPMM	DRAM1	DCPMM	DRAM1	DCPMM	DRAM1	DCPMM	DRAM1	DCPMM	DRAM1	2-2-2
AD + MM	DCPMM	DRAM3	DCPMM	DRAM3	DCPMM	DRAM3	DCPMM	DRAM3	DCPMM	DRAM3	DCPMM	DRAM3	2-2-2
AD	—	DRAM1	—	DRAM1	DCPMM	DRAM1	—	DRAM1	—	DRAM1	DCPMM	DRAM1	2-1-1
MM	—	DRAM2	—	DRAM2	DCPMM	DRAM2	—	DRAM2	—	DRAM2	DCPMM	DRAM2	2-1-1
AD + MM	—	DRAM3	—	DRAM3	DCPMM	DRAM3	—	DRAM3	—	DRAM3	DCPMM	DRAM3	2-1-1
AD	—	DRAM1	DCPMM	DRAM1	DCPMM	DRAM1	—	DRAM1	DCPMM	DRAM1	DCPMM	DRAM1	2-2-1
MM	—	DRAM1	DCPMM	DRAM1	DCPMM	DRAM1	—	DRAM1	DCPMM	DRAM1	DCPMM	DRAM1	2-2-1
AD + MM	—	DRAM3	DCPMM	DRAM3	DCPMM	DRAM3	—	DRAM3	DCPMM	DRAM3	DCPMM	DRAM3	2-2-1
AD	—	DCPMM	—	DRAM1	—	DRAM1	—	DCPMM	—	DRAM1	—	DRAM1	1-1-1
MM	—	DCPMM	—	DRAM1	—	DRAM1	—	DCPMM	—	DRAM1	—	DRAM1	1-1-1
AD + MM	—	DCPMM	—	DRAM3	—	DRAM3	—	DCPMM	—	DRAM3	—	DRAM3	1-1-1
AD	—	DCPMM	DRAM1	DRAM1	DRAM1	DRAM1	—	DCPMM	DRAM1	DRAM1	DRAM1	DRAM1	2-2-1

Figure 9

1. Insert one PMem stick into each Channel. 2. Go with IMC0 channels first and then IMC1 ones. 3. Always insert PMem into Slot 1. Insert it into Slot 0 only when there is only one PMem stick on the channel. 4. PMems installed on the same Socket must be of the same capacity and firmware version.	System: CentOS 7.6 X64 and above Default cache read-write mode: Write-through Mode

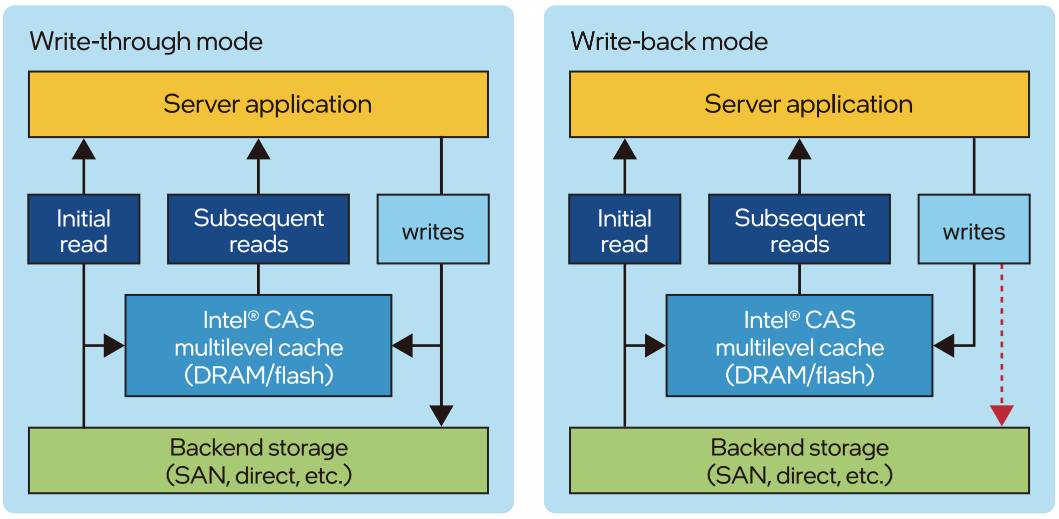

Write-through mode
Server application
Initial read
Subsequent reads
writes
Intel® CAS multilevel cache (DRAM/flash)
Backend storage (SAN, direct, etc.)
Write-back mode
Server application
Initial read
Subsequent reads
writes
Intel® CAS multilevel cache (DRAM/flash)
Backend storage (SAN, direct, etc.)

Figure 10. How CAS Works

Test results:

avg Latency (ms)		4K	8K	16K	32K	64K	128K	256K	512K	1M
	read	0. 0447	0. 0549	0. 0755	0. 1244	0. 2056	0. 3642	0. 6802	1. 4333	2. 8008
	read	0. 0088	0. 0088	0. 0143	0. 0297	0. 0533	0. 1036	0. 2012	0. 4238	0. 8093

BW (MiB/s)		4K	8K	16K	32K	64K	128K	256K	512K	1M
	read	86	140	204	249	302	342	367	348	356
	read	431	866	1076	1044	1168	1204	1241	1178	1235

IOPS		4K	8K	16K	32K	64K	128K	256K	512K	1M
	read	21900	17900	13100	7972	4837	2737	1467	695	356
	read	110000	111000	68900	33400	18700	9631	4963	2356	1234

Figure 11. 480G SATA SSD + 128G PMem

The database team conducted read and write tests on data of sizes 4 KB to 1 MB. From the results shown in Figures 11, we can see that in the test environment, the data read I/O, latency, and bandwidth performance of the SATA SSD with persistent memory is at least three times higher than those of the server with SATA SSD.

According to the test on SQL query statements (see Figure 12), the execution time is reduced by more than 30% for all of the slow queries, and even by more than 50% for a few of them.

SQL Statements for Execution	Execution Time of 480G SATA (sec)	Execution Time of 128G PMEM + 480G SATA SSD (sec)	Decrease in Execution Time (%)
SELECT t1.id, t1.hostname, t1.uuid, t2.grp_en_name FROM t1 LEFT JOIN t2 ON t1.id = t2.host_id LEFT JOIN t3 ON t2.grp_id = grp.id WHERE t1.type = 0 AND t1.host_status = 1;	10.46	4.74	54.68%
select t1.grp_id, t1.host_id, COALESCE(t1.rootgrp, ") from t1, t2, t3 where t1.host_id = t2.id and t1.grp_id = t3.id	6.46	4.1	36.53%
select t1.host_id, t2.plugin_id from t1 inner join t3 on t1.host_id = t3.id inner JOIN t2 on t1.grp_id=t2.grp_id where t3.agent_version != “and t3.uuid != ” and (UNIX_TIMESTAMP(NOW()) - UNIX_TIMESTAMP(t3.update_at)) < 604800	2.25	1.36	39.56%
select uuid, hostname from t1 where t1.host_status != 0 and uuid <> “and hostname <>”	1.78	1.12	37.08%
select t1.host_id, t1.plugin_id, t2.uuid, plugin.dir, t1.target_version from t1 left join t3 on t1.plugin_id=plugin.id left JOIN t2 on t1.host_id=host.id where t2.uuid <>”	5.1	3.08	39.61%
select a.tpl_id, b.host_id from t1 as a inner join t2 as b on a.grp_id=b.grp_id	9.51	5.46	42.59%
select uuid, host_status from t1 where host_status in (1,2)	1.37	0.87	36.50%

Figure 12

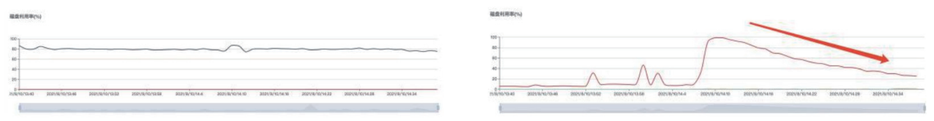
The test server was switched online to test MySQL traffic and observe disk I/O.

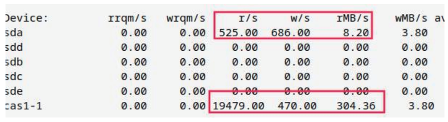
Disk I/O status of online server: 960G*8 (RAID10)

Disk I/O status when online traffic is directed to test server:
480G SATA SSD + 128G PMEM

Disk utilization (%)

Disk utilization (%)





```
Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s av
sda               0.00     0.00  525.00  686.00     8.20     3.80
sdd               0.00     0.00    0.00    0.00     0.00     0.00
sdb               0.00     0.00    0.00    0.00     0.00     0.00
sdc               0.00     0.00    0.00    0.00     0.00     0.00
sde               0.00     0.00    0.00    0.00     0.00     0.00
cas1-1            0.00     0.00 19479.00  470.00   304.36     3.80
```

Figure 13

Figure 13 shows the SSD disk I/O status of two servers in the same service environment. When service traffic is directed to the server that adopts persistent memory, the disk I/O of SSDs shows a gradual decline, but the service traffic does not decline. Here is the explanation. As most of the hot data is cached by persistent memory, more than 90% of the read I/O is undertaken by persistent memory. This dramatically reduces the pressure of reads on SSDs, providing stronger read service capabilities and improving the service efficiency.

The large capacity of persistent memory allows more data to be cached in persistent memory. As the data is cached in persistent memory, it can be flushed more rationally, which enhances the I/O efficiency of back-end devices. Compared to the Page Caching model in which data is flushed to the disk as soon as it is written, the new I/O model is more rational. This is because the latter allows accumulating small pieces of written data into a large data block and then flushing that data block to the disk, which helps to avoid blockage.

Future Prospects

The adoption of Intel® Optane™ persistent memory and SPDK provides a new solution to decreased MySQL performance due to I/O bottlenecks with servers. The solution has been proven to be excellent in terms of IOPS, bandwidth, latency, etc. It is expected that the new software and hardware technologies will allow iQIYI's membership system to serve members much better.

iQIYI will continue to work with Intel to further unlock the potential of Intel® Optane™ persistent memory and SPDK, and collaborate more closely on relevant technologies to make persistent memory benefit more use cases and services and provide a better experience for the large base of iQIYI users.