Multi-Namespace Management

& Performance Optimization

Ron Yuan

Vice President of Firmware

Customers Need Simple Adoption of New NVMe Features

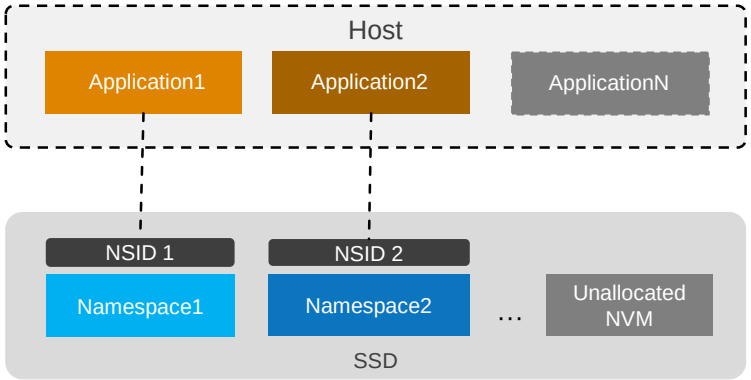
Contents

1. Utilization of Multiple Namespace and Quota by Namespace
2. Solve MySQL Doublewrite Bottleneck and Over-consumption on SSD Endurance with Mixed-media based Multi-Namespace Management

Flexible Utilization with Multiple Namespace

Benefits of Multi-Namespace for SSD with high capacity:

- Lower cost per GB
- Space saving
- Multiple users/applications



For example:

SSD supports only 1 namespace

```
#nvme list
/dev/nvme0n1        -- 8TB
```

PBlaze5 8TB U.2

Create 4 namespaces

```
# nvme list
/dev/nvme0n1        -- 1TB
/dev/nvme0n2        -- 1TB
/dev/nvme0n3        -- 3TB
/dev/nvme0n4        -- 3TB
```

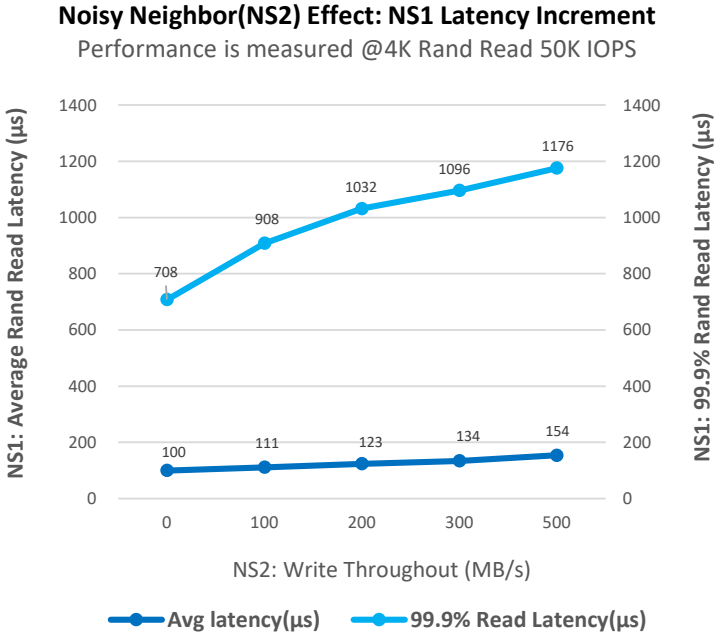
Multi-Namespace on PBlaze5:

- PBlaze5 SSD supports up to 32 namespaces
- Standard management command (nvme create-ns)
- Different AES-256 key
- Different sector size / PI
- Share capacity and performance

Flash Memory Summit

Customer Needs: Performance Control over Namespace

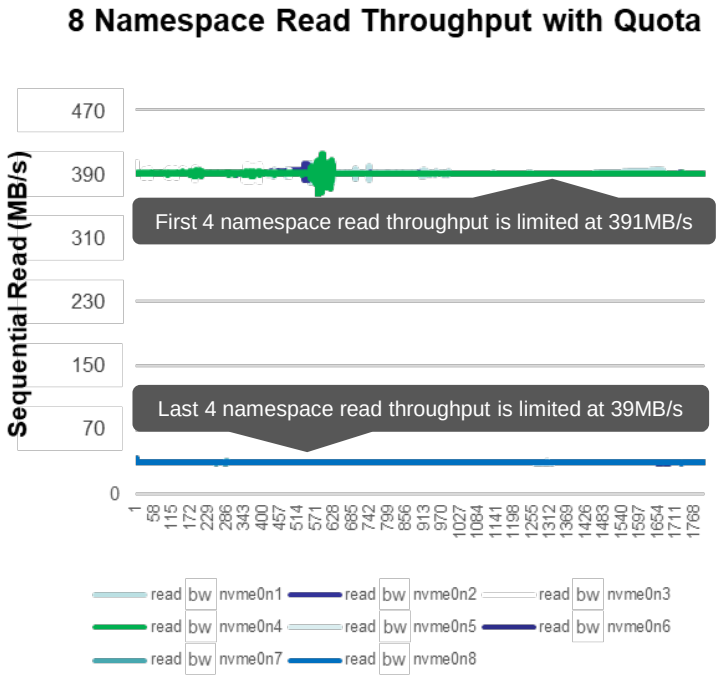
- A big problem for customer:
  when two applications share the same SSD, how can the SSD evenly serve two of them?
- For most of SSDs, greedy application gets more service, slower application needs to suffer long latency
- IOD serve the needs but also brings problems



MEMBLAZE

PBlaze5 QoS Improvement with Quota by Namespace

- Memblaze's solution is to provide customer with VS command to set Bandwidth Quota for each namespace.
- Easy setup, flexible to use.
- Example:
  Create 8 namespaces with the same size
  NS1~4 Seq Read: 391MB/s
  NS5~8 Seq Read: 39MB/s



Customers need simple adoption of new NVMe features

Contents

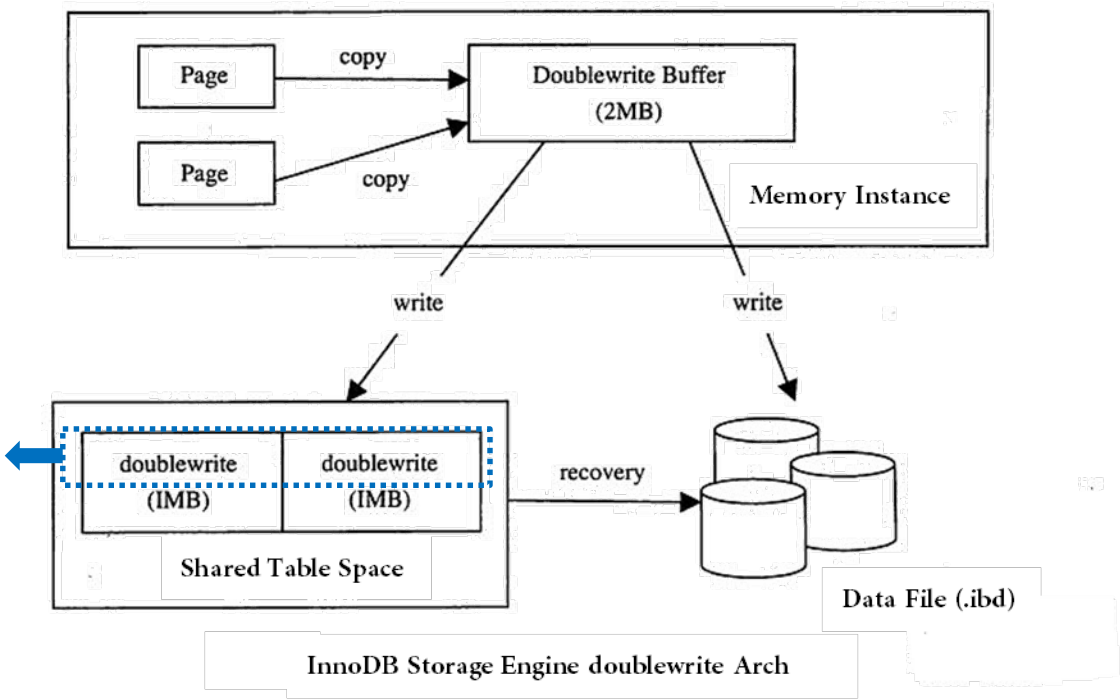
1. Utilization of Multiple Namespace and Quota by Namespace
2. Solve MySQL Doublewrite Bottleneck and Over-consumption on SSD Endurance with Mixed-media based Multi-Namespace Management

MySQL Doublewrite Buffer & Doublewrite Space

Doublewrite is a mechanism to prevent data corruption during accident power loss, partial data is written to the drive.

Doublewrite Space:

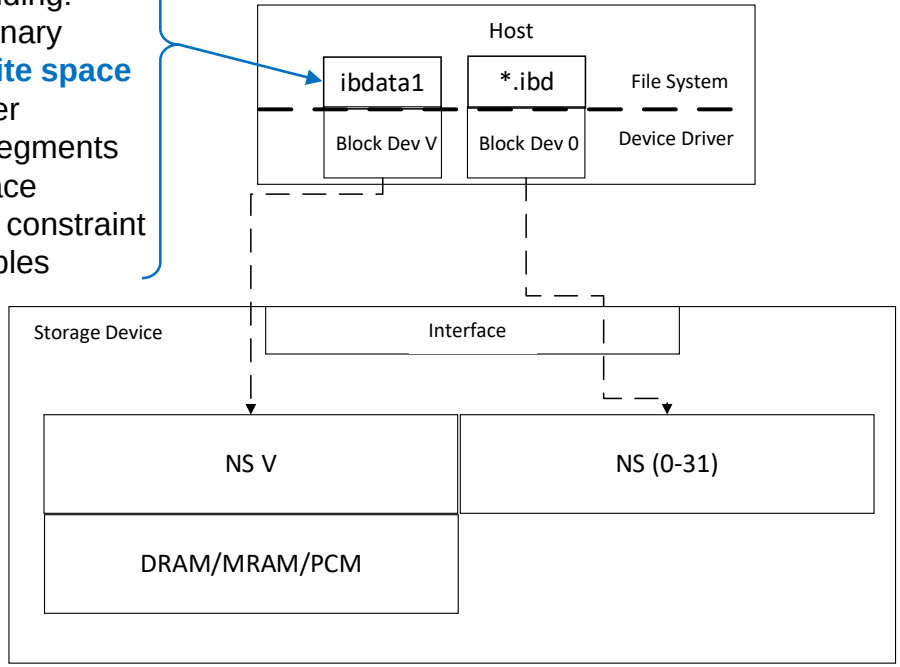
- Data is written twice, in some heavy workloads the doublewrite buffer becomes a performance bottleneck
- Massive writes lead to SSD wears out quickly



InnoDB Storage Engine doublewrite Arch

Mixed-media based Multi-Namespace Management on Memblaze PBlaze5 NVMe SSD

Ibdata1 including:
1.Data dictionary
2.Doublewrite space
3.Insert buffer
4.Rollback segments
5.UNDO space
6.Forign key constraint
7.System tables



Normal Solution:
Put doublewrite buffer on separated drive using high performance media (MRAM/PCM/Xpoint), isolates with NAND based SSD.

Normal solution 2:
Use Atomic write feature to replace double write buffer.

Memblaze Solution:

Put doublewrite buffer on DRAM based namespace,

Doublewrite Buffer Analyze

```
mysql> show variables like "%pool_instance%";
+------------------------------+-------+
| Variable_name                | Value |
+------------------------------+-------+
| innodb_buffer_pool_instances | 4     |
+------------------------------+-------+
1 row in set (0.01 sec)

mysql> show variables like "%double%";
+---------------------------------+----------------------+
| Variable_name                   | Value                |
+---------------------------------+----------------------+
| innodb_doublewrite              | ON                   |
| innodb_parallel_doublewrite_path | /DWB/doublewrite.file |
+---------------------------------+----------------------+
2 rows in set (0.01 sec)

mysql> exit
Bye
[root@localhost data1]# df -h | grep -i dwb
/dev/nvme0n1p2   24M   16M  6.2M  72% /DWB
[root@localhost data1]# ll /DWB/
total 15376
-rw-r----- 1 root root 15728640 Mar 28 22:09 doublewrite.file
drwx------ 2 root root    16384 Mar 28 22:08 lost+found
[root@localhost data1]# du -sh /DWB/doublewrite.file
15M     /DWB/doublewrite.file
```

4 buffer pool instances allocate 8 doublewrite shards, nearly 16MB

1. Double write buffer(DWB) is very small in size
2. only used after sudden power loss event.
3. Enterprise SSD has large DDR to serve as DWB
4. Enterprise SSD has native power loss protection by capacitor.

Here we use Percona MySQL as test case:

1. Percona Parallel Doublewrite Buffer is designed to solve the performance bottleneck which is introduces by traditional Doublewrite buffer.
2. Percona doublewrite space has been separated into a single file (non-tablespace). This file contains shards for all buffer pool instances. Each shard has different offsets.

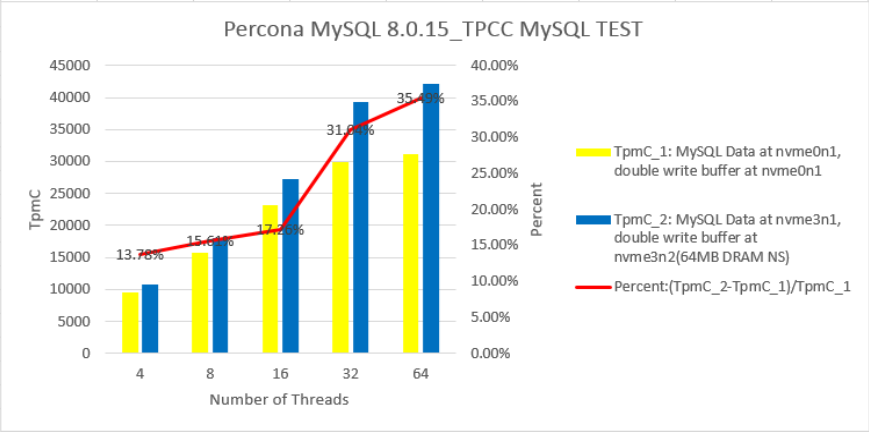
MySQL TPCC Test Environments

Use Memblaze customized firmware, customer can allocate Namespace from DDR space like a RAM disk.

1. PowerEdge R730xd
(1) CPU: Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz 8 Cores * 2
(2) Memory: DDR4 96GB
(3) Memblaze PBlaze5 910 NVMe SSDs:
3.84TB U.2 SSD with 3.84TB namespace * 1 (nvme0n1)
3.84TB U.2 SSD with 3.80TB namespace * 1 (nvme3n1) and 64MB DRAM Namespace *1 (nvme3n2)
2. Centos 7.4 with NVMe driver 1.0, ext4 filesystem
3. Percona MySQL 8.0.15
datadir=/data1, innodb_doublewrite=on, innodb_parallel_doublewrite_path=/DWB/doublewrite.file
VS.
datadir=/data1, innodb_doublewrite=on, innodb_parallel_doublewrite_path=/data1/xb_doublewrite
innodb_buffer_pool_size=16GB, innodb_buffer_pool_instances=8 => need 32MB double write file
innodb_flush_log_at_trx_commit = 1, innodb_flush_method=O_DIRECT
innodb_write_io_threads=16, innodb_read_io_threads=8
innodb_io_capacity=10000, log_bin=/data1/mysql-bin
4. TPCC MySQL
(1) connections=4,8,16,32,64,128
(2) warehouse=28000 => Test data amount is 3TB
(3) warmup_time=600
(4) running_time=10800

```
[root@localhost ~]# du -sh /data1/tpcc/
3.0T    /data1/tpcc/
[root@localhost ~]# du -sh /data1/tpcc/*
569G    /data1/tpcc/customer.ibd
48M     /data1/tpcc/district.ibd
160G    /data1/tpcc/history.ibd
17M     /data1/tpcc/item.ibd
8.7G    /data1/tpcc/new_orders.ibd
1.2T    /data1/tpcc/order_line.ibd
76G     /data1/tpcc/orders.ibd
1011G   /data1/tpcc/stock.ibd
14M     /data1/tpcc/warehouse.ibd
```

MySQL TPCC Test Results



Put Innodb parallel double write file on DRAM Namespace, performance **improves 35.49%** under 64 thread concurrency.

THREAD	TpmC_1: MySQL Data at nvme0n1, double write buffer at nvme0n1	TpmC_2: MySQL Data at nvme3n1, double write buffer at nvme3n2(64MB DRAM NS)	TpmC_2-TpmC_1	(TpmC_2-TpmC_1)/TpmC_1
4	9505.272	10814.7	1309.428	13.78%
8	15678.145	18124.768	2446.623	15.61%
16	23205.877	27210.906	4005.029	17.26%
32	29988.883	39297.746	9308.863	31.04%
64	31138.277	42189.027	11050.75	35.49%

MySQL TPCC Test Results -- IO Press

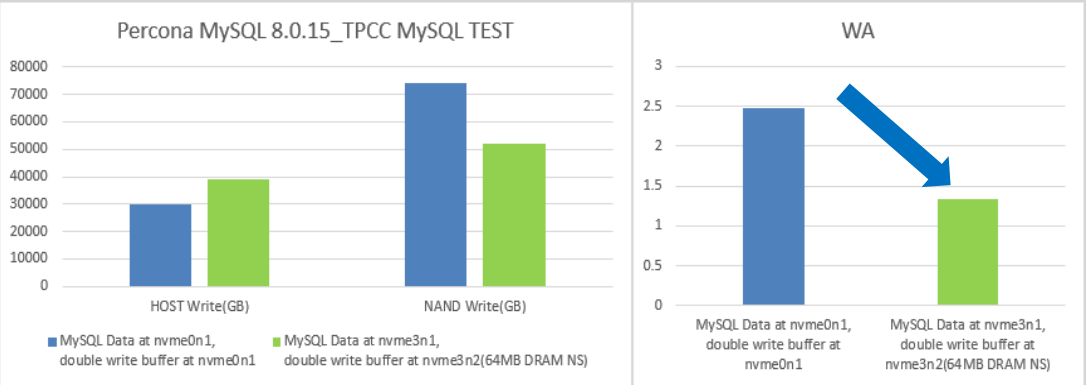
Put Innodb parallel double write file on DRAM Nanmespace:

1. SSD Avg Random Read IOPS and **Random Read throughput improves 38.3%;**
2. SSD Avg Random **Write throughput improves 37.1%;**
3. SSD Avg Random **Read latency reduces 7.4%;**
4. SSD Avg Random **Write latency reduces 52.6%;**

Percona_8.0.15 TPCC TEST Parameters: Warehouse=28000 Warmup_time=600 Running_time=10800
MySql Parameters: innodb_buffer_pool_size=16GB innodb_buffer_pool_instances=8 innodb_flush_log_at_trx_commit=1

TPCC MySQL Test THREAD=64		Avg. Read IOPS	Avg. Read MBPS	Avg. Read Block Size	Avg. Write IOPS	Avg. Write MBPS	Avg. Write Block Size	Avg. Read Latency	Avg. Write Latency
TpmC_1	nvme0n1(MySQL Data & double write buffer)	46888.194	732.626	16	20410.696	540.13	27.1	0.405	2.758
TpmC_2	nvme3n1(MySQL Data)	64845.669	1013.213	16	28242.749	396.073	14.36	0.375	1.306
	nvme3n2(double write buffer in 64MB DRAM NS)				2837.63	344.312	124.25	N/A	1.775
	nvme3n1 + nvme3n2		1013.213			740.385			
Percent		Improves 38.3%	Improves 38.3%			Improves 37.1%		Reduces 7.4%	Reduces 52.6%(MySQL Data)

MySQL TPCC Host & NAND Write



Put Innodb parallel double write file on DRAM Namespace, **WA reduces 46%**.

Test Case	Host Write(GB)	NAND Write(GB)	WA
MySQL Data at nvme0n1, double write buffer at nvme0n1	29,953	74,229	2.478
MySQL Data at nvme3n1, double write buffer at nvme3n2(64MB DRAM NS)	39,114	52,220	1.335
Percent	Improves 31%	Reduces 30%	Reduces 46%

Summary

- NVMe namespace feature provides a great possibility to manage different media, different performance, different security and more.
- Customer is always asking for
  - Lower cost
  - Robust product
  - Simplified adoption
- We have seen slow adoption of NVMe new features such as streams or IOD.
- Focus on what customer really needs