Disaggregated Storage Architecture – Challenges and Quality of Service for NVMeoF based storage systems

Intel Corporation

- Vaidy Krishnamoorthy

- Mrittika Ganguli

Notices and Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. **No product can be absolutely secure.**

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. For more complete information about performance and benchmark results, visit http://www.intel.com/benchmarks .

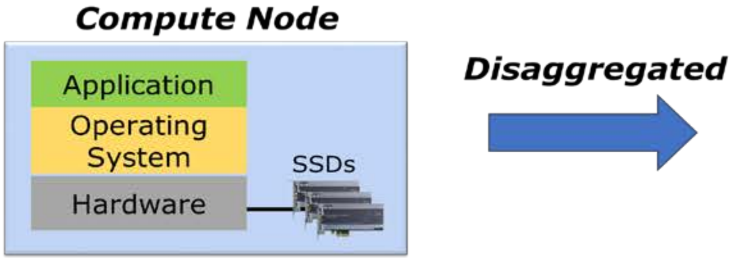
Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit http://www.intel.com/benchmarks .

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

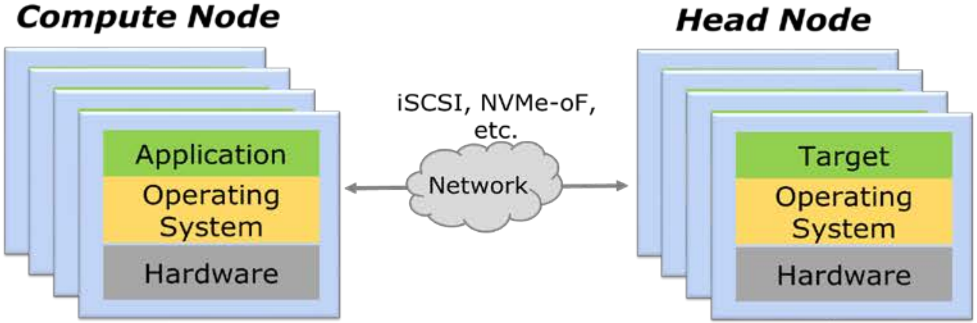
The benchmark results may need to be revised as additional testing is conducted. The results depend on the specific platform configurations and workloads utilized in the testing, and may not be applicable to any particular user's components, computer system or workloads. The results are not necessarily representative of other benchmarks and other benchmark results may show greater or lesser impact from mitigations.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

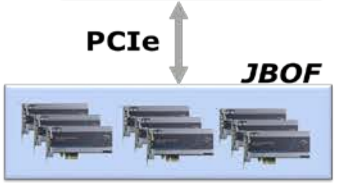
Disaggregated Storage Architecture



- **Local attached storage**
- **Static binding**
- **Stranded capacity**, IOPS
- Inefficient, increased TCO

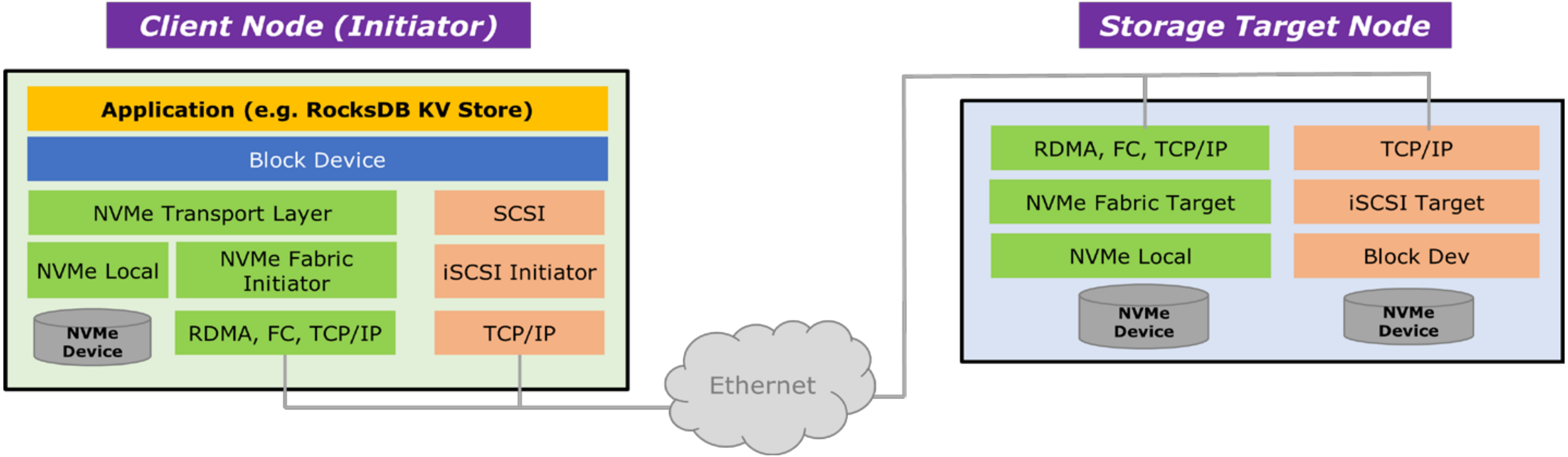


- **Logical disaggregation**
- Consumes physical or logical block devices
- **Dynamic binding** based on workload requirements
- Efficient, improved TCO



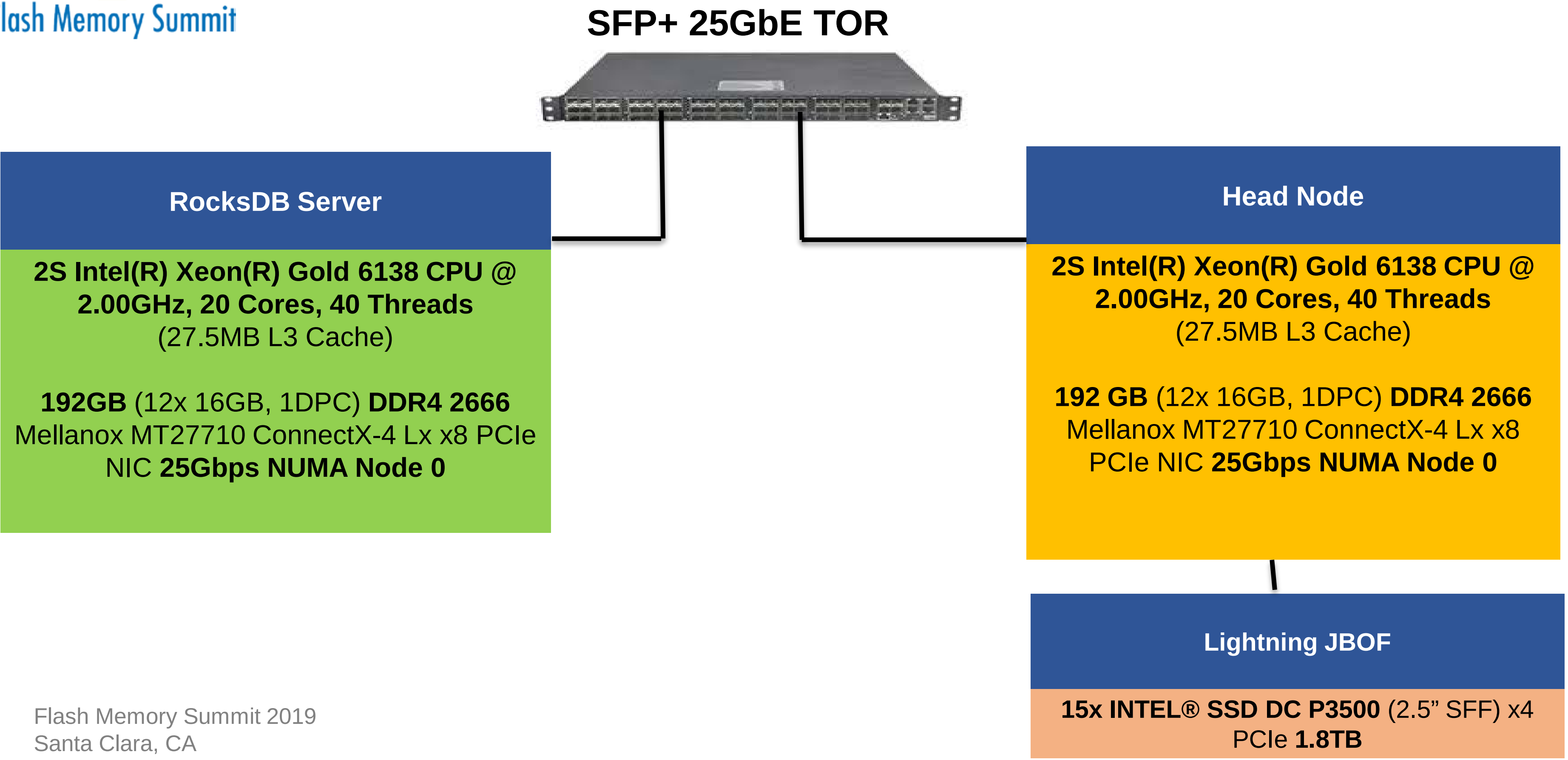
- **Physical disaggregation**
- **Static binding**
- **Shared resources**
- Target can expose physical or logical devices

Remote Block Storage – Network Protocols



- Enables sharing of NVMe flash storage over network
- Can use traditional block protocols (e.g. iSCSI) or NVMe optimized protocols (e.g., NVMe/TCP)
- NVMe over Fabrics – supports multiple transports, extends NVMe efficiency over network
  - Poll and interrupt mode architecture
  - Kernel and user mode implementations

Test Configuration - Hardware



SFP+ 25GbE TOR

RocksDB Server

2S Intel(R) Xeon(R) Gold 6138 CPU @ 2.00GHz, 20 Cores, 40 Threads (27.5MB L3 Cache)

192GB (12x 16GB, 1DPC) **DDR4 2666** Mellanox MT27710 ConnectX-4 Lx x8 PCIe NIC **25Gbps NUMA Node 0**

Head Node

2S Intel(R) Xeon(R) Gold 6138 CPU @ 2.00GHz, 20 Cores, 40 Threads (27.5MB L3 Cache)

192 GB (12x 16GB, 1DPC) **DDR4 2666** Mellanox MT27710 ConnectX-4 Lx x8 PCIe NIC **25Gbps NUMA Node 0**

Lightning JBOF

15x INTEL® SSD DC P3500 (2.5” SFF) x4 PCIe **1.8TB**

Test Configuration - Software

Operating System

Distro: Fedora 27

Kernel: 5.0.0-rc4 **Arch**: x86_64

Tuning:

- XFS filesystem, agcount=32, mount with discard
- CPU Profile: Performance
- NIC MTU: 9000
- Huge Pages: Turned off

***NOTE**: see back up for detailed config*

RocksDB

Version: Master with commit 301da345aed32577da649ffdcea0f3b5e2fe979f

Record Size: Key - 16B, Value – 100B

Database Size: 456 GB, 4 Billion keys

RocksDB Instances: Upto 9 (1 SSD per 3 instances)

Read/Write Dataset: 5 million records

- Dataset size higher (> 3:1 DRAM size)

Testing Tool: db_bench

Block Size: 8KB**, Block Cache**: 16GB

Threads: 32 (for fill), 16 (for randrw & randr), 1 (randw)

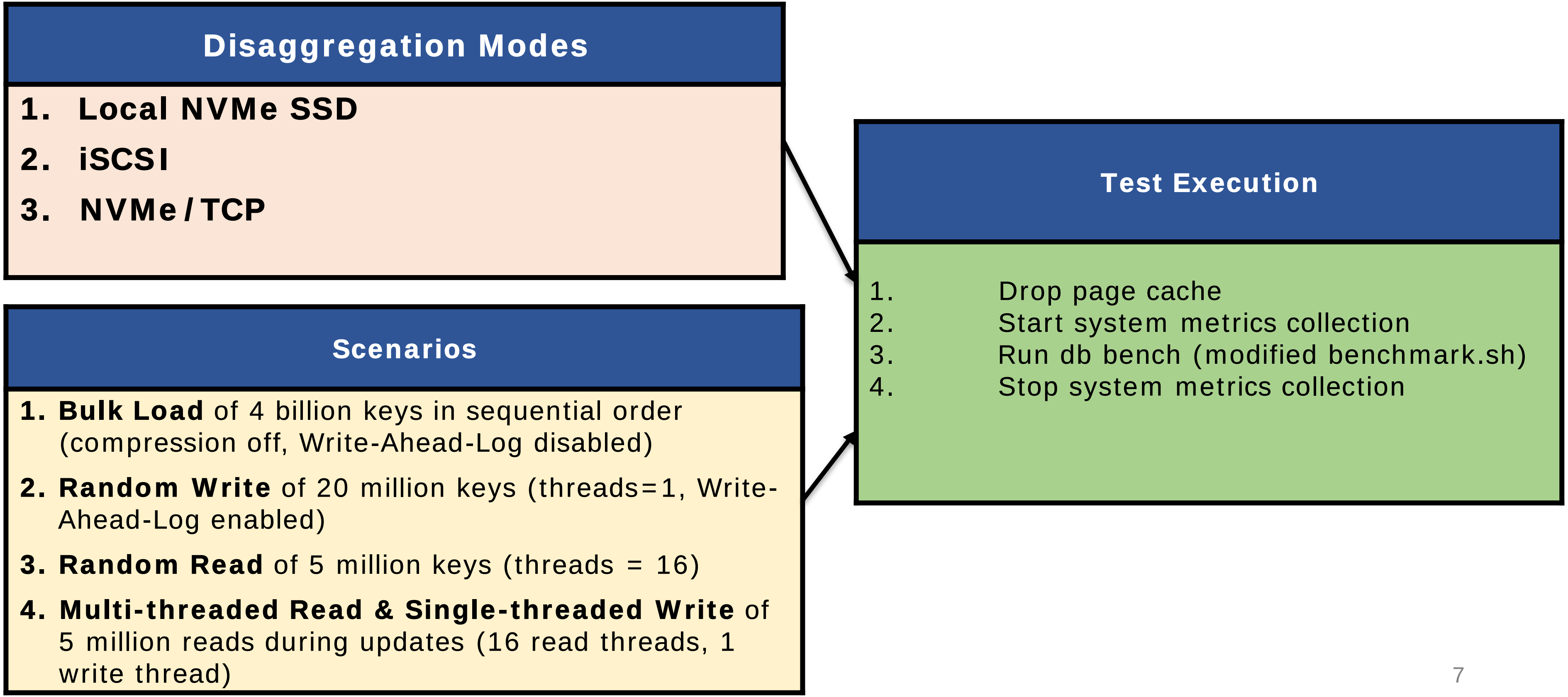
Database & **Write-Ahead-Log** co-located on the same drive

Jemalloc memory allocator

Direct IO for flush_and_compaction, reads

Flash Memory Summit

Test Methodology



Disaggregation Modes

1. **Local NVMe SSD**
2. **iSCSI**
3. **NVMe/TCP**

Scenarios

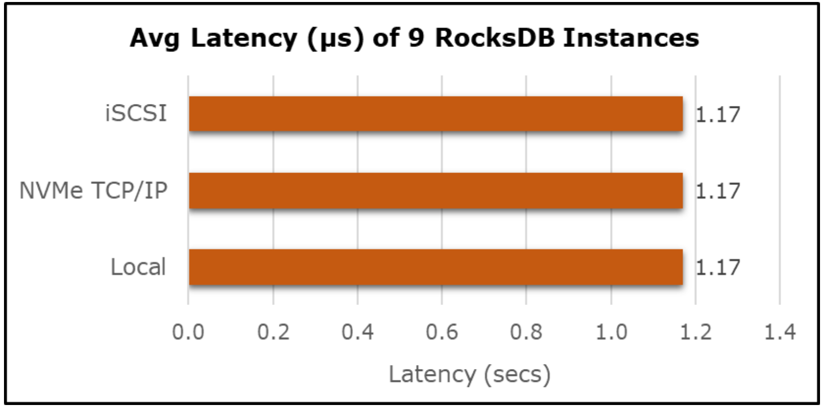
1. **Bulk Load** of 4 billion keys in sequential order (compression off, Write-Ahead-Log disabled)
2. **Random Write** of 20 million keys (threads=1, Write-Ahead-Log enabled)
3. **Random Read** of 5 million keys (threads = 16)
4. **Multi-threaded Read & Single-threaded Write** of 5 million reads during updates (16 read threads, 1 write thread)

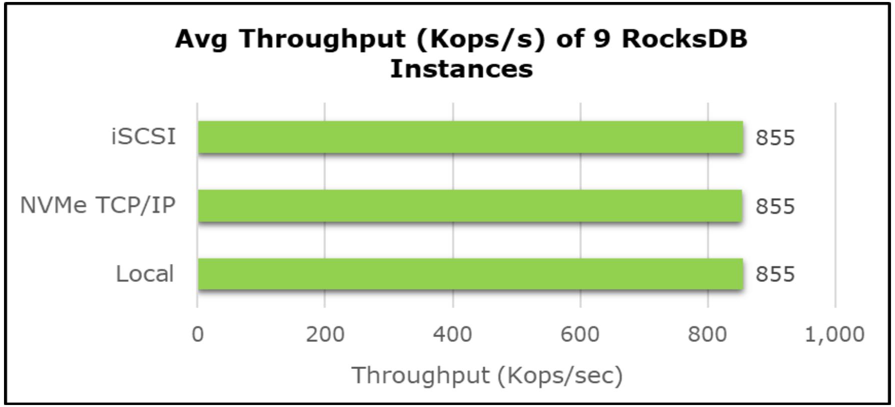
Test Execution

1. Drop page cache
2. Start system metrics collection
3. Run db bench (modified benchmark.sh)
4. Stop system metrics collection

Performance Comparison: Bulk Load

32 threads per RocksDB instance

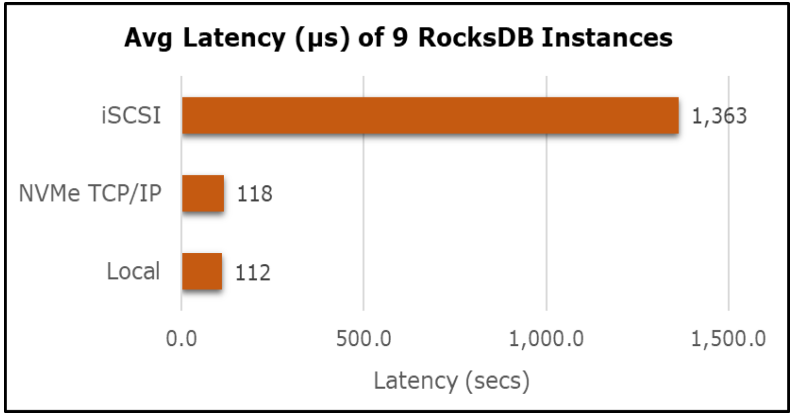


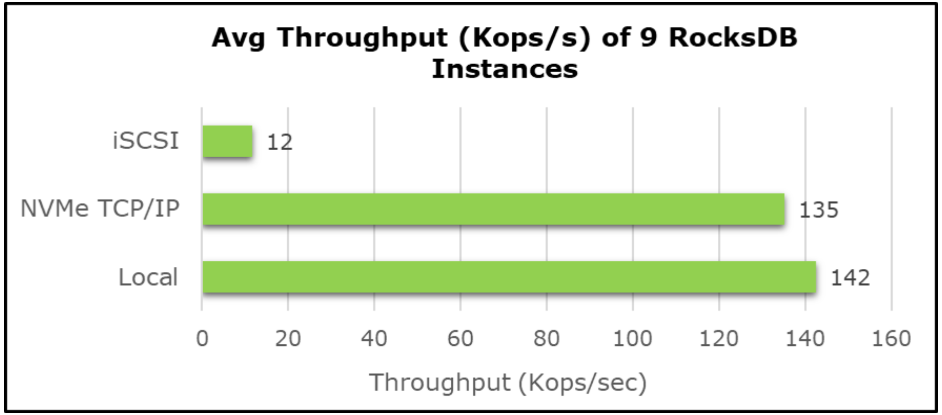


Comparable Performance between local and network attached config (Sequential IO)

Performance Comparison: Random Read

16 threads per RocksDB instance



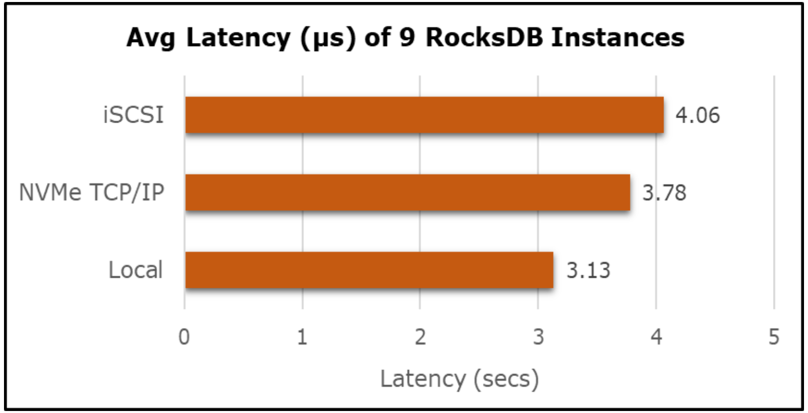


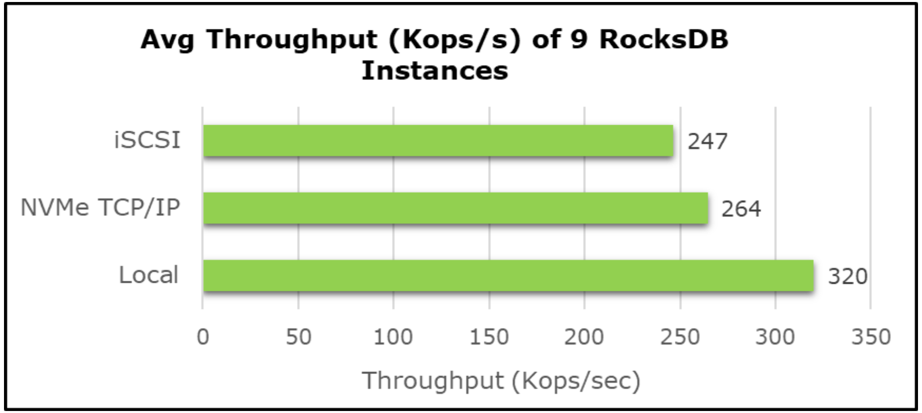
Minimal Performance overhead with NVMe over TCP/IP

Flash Memory Summit

Performance Comparison: Random Write

1 thread per RocksDB instance

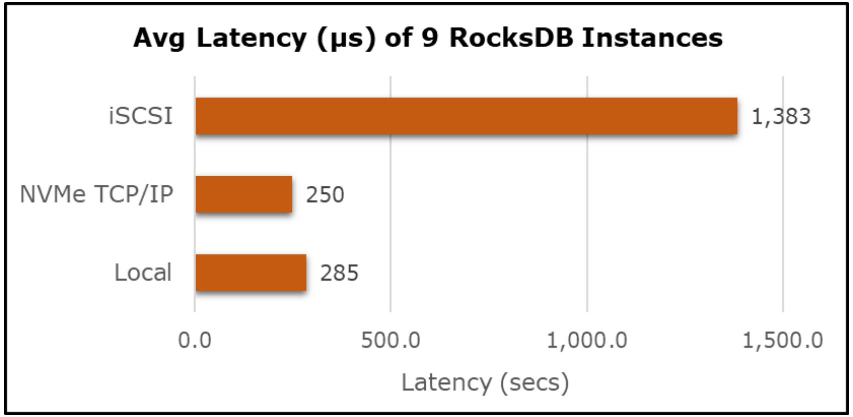


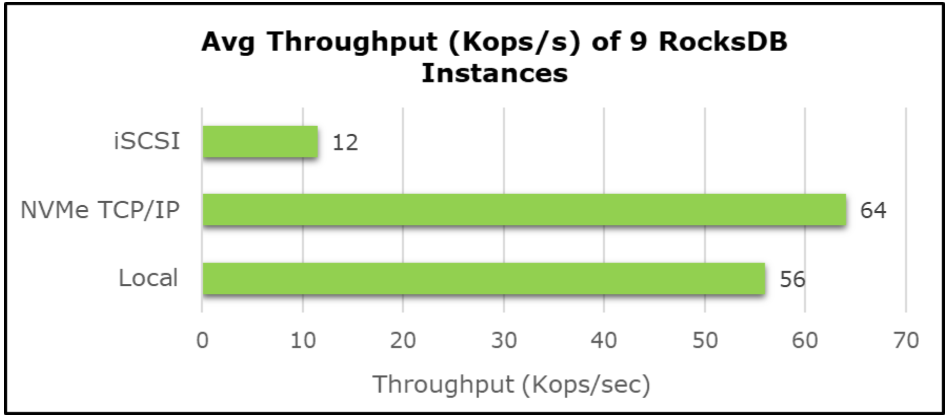


NVMe over TCP/IP performance is better than iSCSI

Perf Comparison: ReadwhileWrite

16 Read Threads, 1 rate limited write thread (2Mbps) per RocksDB instance



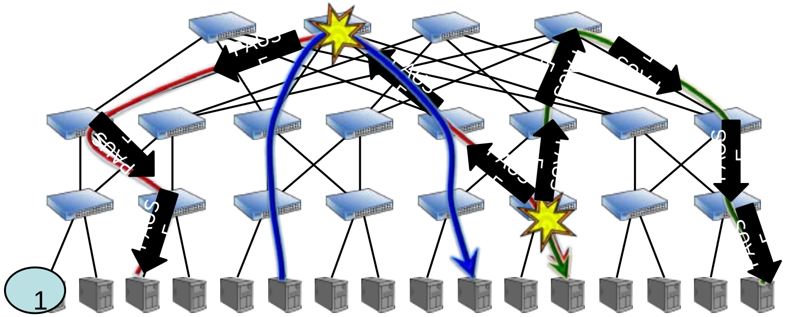


NVMe over TCP/IP scales better as number of clients increase

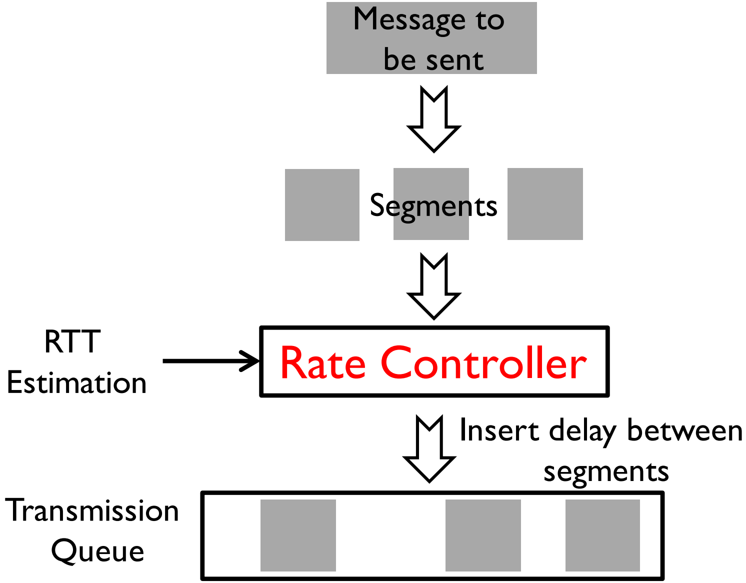
Challenges of NVMeoF

- Increased Network Traffic leads to Network Congestion
- QoS for Applications

Congestion Spreading in Lossless Networks - Existing



- ➢ Port-based congestion control incurs congestion spreading
- ➢ DCQCN: incorporating explicit congestion notification to support flow-based congestion control



Message to be sent

Segments

RTT Estimation

Rate Controller

Insert delay between segments

Transmission Queue

➢ Target rate is determined by segment size and delay between segments

RTT based rate control better than window based congestion

Congestion control

Algorithms to prevent that the sender overloads the network. Detection and Recovery. Congestion is detected with

  - an explicit congestion notification from a packet switch
  - packet loss: in wired networks, the main reason for packet loss is congested buffers.

- The mechanism is implemented at the sender. The sender has two parameters:
  - Congestion Window (cwnd)
  - Slow-start threshhold Value (ssthresh) Initial value is the advertised window size
- Congestion control works in two modes:
  - slow start (cwnd < ssthresh)
  - congestion avoidance (cwnd >= ssthresh)
  - Congestion avoidance phase is started if cwnd has reached the slow-start threshold value
- RTT based window adjustment:
  - Calculate the propagation delay and influence to RTT
  - Adjust cwnd based on delay

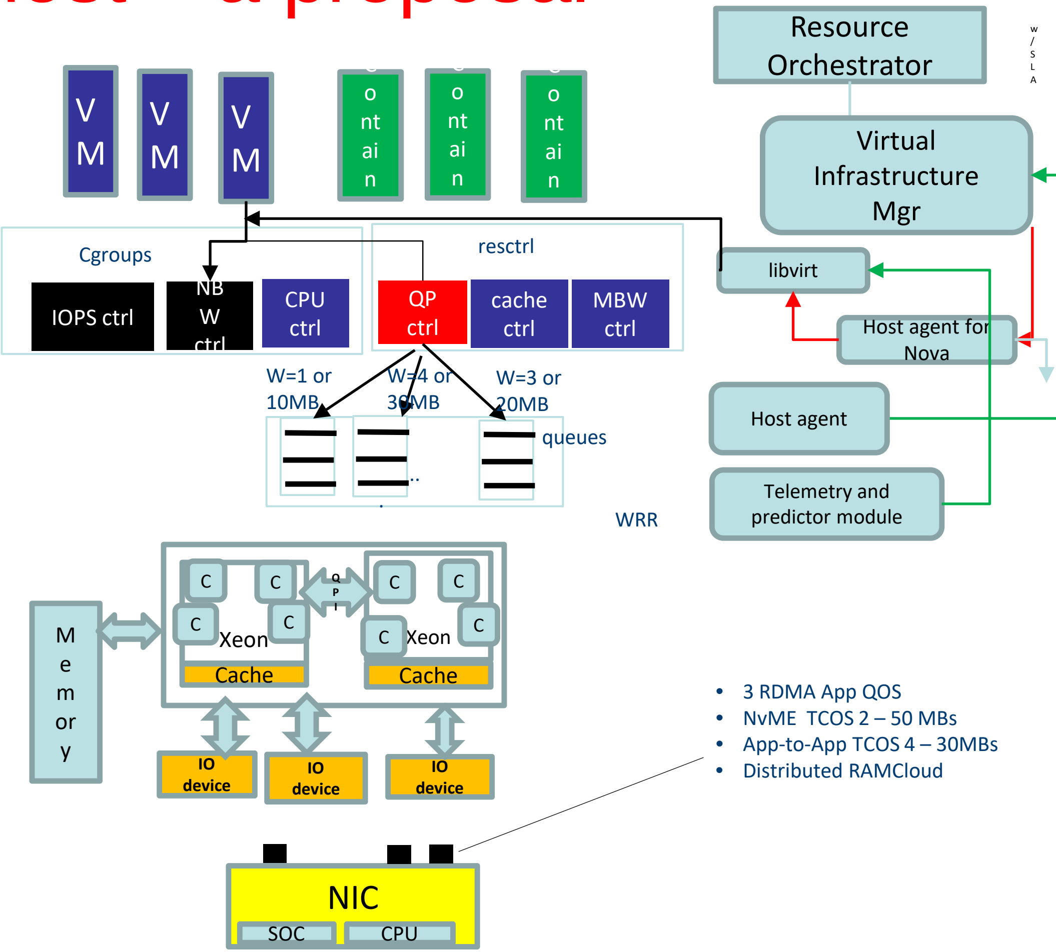
Get RTT based window adjustment for RDMA

Ack: http://www.ece.virginia.edu

Flash Memory Summit

RDMA QOS at host – a proposal

- VM Resource Orchestrator sets QOS as flavors derived from SLAs.
- Orchestrator maps those to filters: pciepassthrufilter, NUMTopologyfilter, RamFilter, ComputeFilter,TrustedFilter, QPConfigFilter
- Scenario: VM on same port and same PF and TCs but on different scheduling pattern on the RDMA work scheduler. Packet scheduler does rate limits and BW allocation.
- At the switch/NOS - DSCP – priority TOS to COS value will be set at the port in Linux.
- PF netdev has the TC classes.
    - DCB netlink and queues belonging to a TC gets configured when driver gets configured.
    - This is advertised to OS.
    - netprio cgroups can be created for ROCE UDP port. Network driver and libvirt APIs.
- A service table to map the QP. Sub-TCs for RDMA, port number per service ID. Sub-TC TOS values are passed onto verbs call. This is enabled via libvirt.



Resource Orchestrator

w/SLA

Virtual Infrastructure Mgr

VM

VM

VM

ontain

ontain

ontain

Cgroups

IOPS ctrl

NBW ctrl

CPU ctrl

resctrl

QP ctrl

cache ctrl

MBW ctrl

libvirt

Host agent for Nova

Host agent

Telemetry and predictor module

W=1 or 10MB

W=4 or 30MB

W=3 or 20MB

queues

WRR

Memory

C

C

C

C

Xeon

Cache

QPI

C

C

C

C

Xeon

Cache

IO device

IO device

IO device

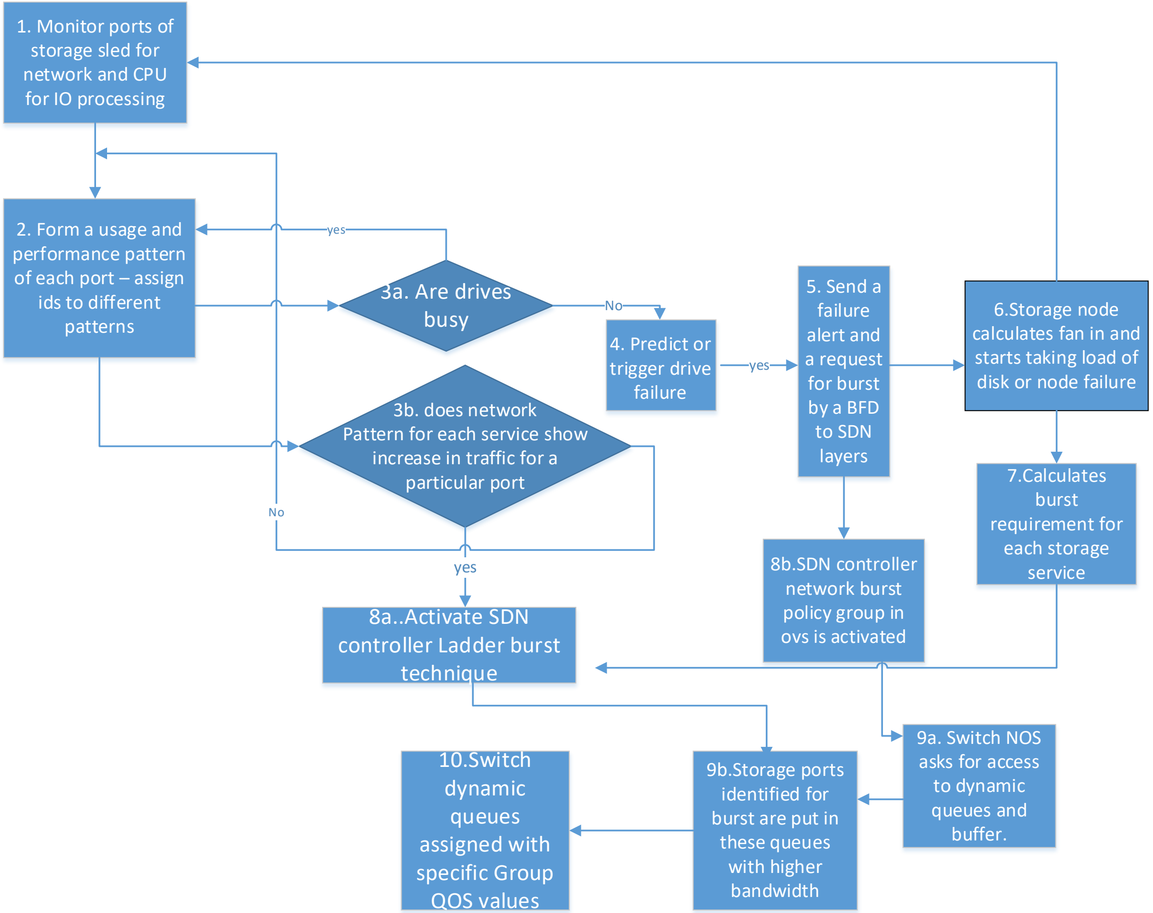
NIC

SOC

CPU

- 3 RDMA App QOS
- NvME TCOS 2 – 50 MBs
- App-to-App TCOS 4 – 30MBs
- Distributed RAMCloud

Group QOS for burst management



1. Monitor ports of storage sled for network and CPU for IO processing

2. Form a usage and performance pattern of each port – assign ids to different patterns

3a. Are drives busy

3b. does network Pattern for each service show increase in traffic for a particular port

4. Predict or trigger drive failure

5. Send a failure alert and a request for burst by a BFD to SDN layers

6.Storage node calculates fan in and starts taking load of disk or node failure

7.Calculates burst requirement for each storage service

8a..Activate SDN controller Ladder burst technique

8b.SDN controller network burst policy group in ovs is activated

9a. Switch NOS asks for access to dynamic queues and buffer.

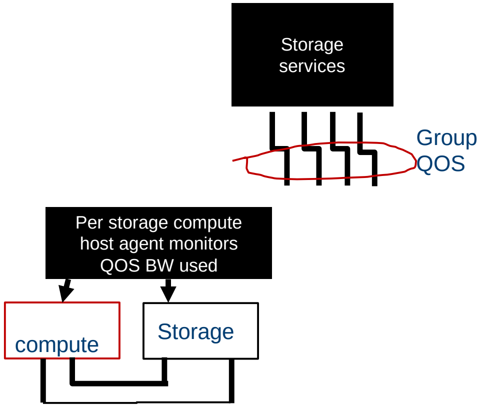
9b.Storage ports identified for burst are put in these queues with higher bandwidth

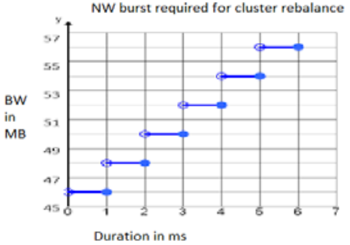
10.Switch dynamic queues assigned with specific Group QOS values

VxLAN extensions and Step function for QOS

- Group QOS for Storage burst and re-balance
  - Add to vxlan header for group QOS presence bit
  - Add bits in VxLAN Group based policy

1





example

5<t<10

t=1

amount.

Step function f(y) = {3t-a, 11<t<ψ

{ t+a,

{ 10,

Where a is the burst MB step

3

LADDER technique or step function:

1. VxLAN groups QOS extensions – presence bit

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|R|R|C|R|I|R|R|R|  QoS |              Reserved                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                VXLAN Network Identifier (VNI) |    Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

2. VxLAN GBP enhanced to include Burst

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|G|R|R|R|I|R|R|R|R|D|R|R|A|R|R|B|        Group Policy ID        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          VXLAN Network Identifier (VNI)       |    Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

2

Summary

- Locally attached SSDs result in stranded flash capacity and increased TCO
- Disaggregating flash storage enables independent scaling of compute and storage
- NVMe over TCP enables disaggregation without changes to network infrastructure
- RocksDB using NVMeoF delivers scalability while delivering comparable performance to local storage
- Manage Network Congestion
- QOS at the host with RTT and VXLAN group QOS for mitigation

Backup

Flash Memory Summit

BIOS Setup

Profiles

- CPU Power and Performance Policy: Performance
- Workload Configuration: Balanced
- Memory RAS Configuration: Maximum Performance
- Fan Profile: Performance

Enabled

- Hyper-Threading
- Enhanced Intel SpeedStep® Tech
- Intel® Turbo Boost Technology
- Uncore Frequency Scaling
- Performance P-Limit

Disabled

- Cluster on Die
- Early Snoop
- CPU C States
- Energy Efficient Turbo

Test Setup (Linux OS)

/etc/sysctl.conf

```
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
net.core.netdev_max_backlog = 250000
```

/etc/security/limits.conf

```
* soft nofile 65536
* hard nofile 1048576
* soft nproc 65536
* hard nproc unlimited
* hard memlock unlimited
```

CPU Profile

```
echo performance> /sys/devices/system/cpu/cpu{0..n}/cpufreq/scaling_governor
```

Huge Page

```
echo never> /sys/kernel/mm/transparent_hugepage/defrag
echo never> /sys/kernel/mm/transparent_hugepage/enabled
```

Network

```
ifconfig <eth> mtu 9000
ifconfig <eth> txqueuelen 1000
```

Test Setup (RocksDB)

```
                      Options.error_if_exists: 0
                     Options.create_if_missing: 0
                       Options.paranoid_checks: 1
                                   Options.env: 0x56126fe7b240
                              Options.info_log: 0x561270c35d90
              Options.max_file_opening_threads: 16
                            Options.statistics: (nil)
                             Options.use_fsync: 0
                     Options.max_log_file_size: 0
                Options.max_manifest_file_size: 1073741824
                 Options.log_file_time_to_roll: 0
                     Options.keep_log_file_num: 1000
                  Options.recycle_log_file_num: 0
                       Options.allow_fallocate: 1
                      Options.allow_mmap_reads: 0
                     Options.allow_mmap_writes: 0
                      Options.use_direct_reads: 1
                      Options.use_direct_io_for_flush_and_compaction: 1
        Options.create_missing_column_families: 0
                            Options.db_log_dir:
                               Options.wal_dir: /mnt/nvme2n1/wal
             Options.table_cache_numshardbits: 6
                    Options.max_subcompactions: 4
                Options.max_background_flushes: 7
                       Options.WAL_ttl_seconds: 0
                     Options.WAL_size_limit_MB: 0
           Options.manifest_preallocation_size: 4194304
                   Options.is_fd_close_on_exec: 1
                 Options.advise_random_on_open: 1
                  Options.db_write_buffer_size: 0
                  Options.write_buffer_manager: 0x561270c3de90
       Options.access_hint_on_compaction_start: 1
Options.new_table_reader_for_compaction_inputs: 1
         Options.random_access_max_buffer_size: 1048576
                    Options.use_adaptive_mutex: 0
                          Options.rate_limiter: 0x561270c35860
```

```
  Options.sst_file_manager.rate_bytes_per_sec: 0
                     Options.wal_recovery_mode: 2
                Options.enable_thread_tracking: 0
                Options.enable_pipelined_write: 1
       Options.allow_concurrent_memtable_write: 1
    Options.enable_write_thread_adaptive_yield: 1
           Options.write_thread_max_yield_usec: 100
          Options.write_thread_slow_yield_usec: 3
                             Options.row_cache: None
                            Options.wal_filter: None
           Options.avoid_flush_during_recovery: 0
           Options.allow_ingest_behind: 0
           Options.preserve_deletes: 0
           Options.two_write_queues: 0
           Options.manual_wal_flush: 0
           Options.max_background_jobs: 8
           Options.max_background_compactions: 16
           Options.avoid_flush_during_shutdown: 0
         Options.writable_file_max_buffer_size: 1048576
           Options.delayed_write_rate : 8388608
           Options.max_total_wal_size: 17179869184
           Options.delete_obsolete_files_period_micros: 21600000000
                 Options.stats_dump_period_sec: 600
                        Options.max_open_files: -1
                        Options.bytes_per_sync: 8388608
                    Options.wal_bytes_per_sync: 8388608
     Options.compaction_readahead_size: 0
Compression algorithms supported:
    kZSTDNotFinalCompression supported: 0
    kZSTD supported: 0
    kXpressCompression supported: 0
    kLZ4HCCompression supported: 0
    kLZ4Compression supported: 0
    kBZip2Compression supported: 0
    kZlibCompression supported: 1
    kSnappyCompression supported: 0
```

Test Setup (RocksDB)

```
Fast CRC32 supported: Supported on x86
Options for column family [default]:
             Options.comparator: leveldb.BytewiseComparator
         Options.merge_operator: PutOperator
      Options.compaction_filter: None
      Options.compaction_filter_factory: None
       Options.memtable_factory: SkipListFactory
          Options.table_factory: BlockBasedTable
          table_factory options:   flush_block_policy_factory:
FlushBlockBySizePolicyFactory (0x561270c2cb20)
  cache_index_and_filter_blocks: 1
  cache_index_and_filter_blocks_with_high_priority: 0
  pin_l0_filter_and_index_blocks_in_cache: 1
  pin_top_level_index_and_filter: 0
  index_type: 0
  hash_index_allow_collision: 1
  checksum: 1
  no_block_cache: 0
  block_cache: 0x561270c2caa0
  block_cache_name: LRUCache
  block_cache_options:
    capacity : 34359738368
    num_shard_bits : 6
    strict_capacity_limit : 0
    memory_allocator : None
    high_pri_pool_ratio: 0.000
  block_cache_compressed: (nil)
  persistent_cache: (nil)
  block_size: 16384
  block_size_deviation: 10
  block_restart_interval: 16
  index_block_restart_interval: 1
  metadata_block_size: 4096
  partition_filters: 0
  use_delta_encoding: 1
```

```
filter_policy: rocksdb.BuiltinBloomFilter
  whole_key_filtering: 1
  verify_compression: 0
  read_amp_bytes_per_bit: 0
  format_version: 2
  enable_index_compression: 1
  block_align: 0
      Options.write_buffer_size: 134217728
Options.max_write_buffer_number: 8
        Options.compression: NoCompression
                Options.bottommost_compression: Disabled
     Options.prefix_extractor: nullptr
 Options.memtable_insert_with_hint_prefix_extractor: nullptr
           Options.num_levels: 6
      Options.min_write_buffer_number_to_merge: 1
   Options.max_write_buffer_number_to_maintain: 0
          Options.bottommost_compression_opts.window_bits: -14
                Options.bottommost_compression_opts.level: 32767
             Options.bottommost_compression_opts.strategy: 0
       Options.bottommost_compression_opts.max_dict_bytes: 0
       Options.bottommost_compression_opts.zstd_max_train_bytes: 0
                Options.bottommost_compression_opts.enabled: false
          Options.compression_opts.window_bits: -14
                Options.compression_opts.level: 32767
             Options.compression_opts.strategy: 0
       Options.compression_opts.max_dict_bytes: 0
       Options.compression_opts.zstd_max_train_bytes: 0
                Options.compression_opts.enabled: false
    Options.level0_file_num_compaction_trigger: 4
        Options.level0_slowdown_writes_trigger: 20
            Options.level0_stop_writes_trigger: 20
                 Options.target_file_size_base: 134217728
           Options.target_file_size_multiplier: 1
              Options.max_bytes_for_level_base: 1073741824
Options.level_compaction_dynamic_level_bytes: 1
        Options.max_bytes_for_level_multiplier: 8.000000
```

Test Setup (RocksDB)

```
Options.max_bytes_for_level_multiplier_addtl[0]: 1
Options.max_bytes_for_level_multiplier_addtl[1]: 1
Options.max_bytes_for_level_multiplier_addtl[2]: 1
Options.max_bytes_for_level_multiplier_addtl[3]: 1
Options.max_bytes_for_level_multiplier_addtl[4]: 1
Options.max_bytes_for_level_multiplier_addtl[5]: 1
Options.max_bytes_for_level_multiplier_addtl[6]: 1
     Options.max_sequential_skip_in_iterations: 8
                  Options.max_compaction_bytes: 3355443200
                      Options.arena_block_size: 16777216
 Options.soft_pending_compaction_bytes_limit: 0
 Options.hard_pending_compaction_bytes_limit: 0
     Options.rate_limit_delay_max_milliseconds: 1000000
              Options.disable_auto_compactions: 0
                      Options.compaction_style: kCompactionStyleLevel
                        Options.compaction_pri: kMinOverlappingRatio
Options.compaction_options_universal.size_ratio: 1
Options.compaction_options_universal.min_merge_width: 2
Options.compaction_options_universal.max_merge_width: 4294967295
Options.compaction_options_universal.max_size_amplification_percent: 200
Options.compaction_options_universal.compression_size_percent: -1
Options.compaction_options_universal.stop_style:
kCompactionStopStyleTotalSize
Options.compaction_options_fifo.max_table_files_size: 0
Options.compaction_options_fifo.allow_compaction: 1
Options.compaction_options_fifo.ttl: 0
                 Options.table_properties_collectors:
                 Options.inplace_update_support: 0
               Options.inplace_update_num_locks: 10000
             Options.memtable_prefix_bloom_size_ratio: 0.000000
```

```
Options.memtable_huge_page_size: 0
                        Options.bloom_locality: 0
                 Options.max_successive_merges: 0
             Options.optimize_filters_for_hits: 1
             Options.paranoid_file_checks: 0
             Options.force_consistency_checks: 0
             Options.report_bg_io_stats: 0
                            Options.ttl: 0
```