

Jary Chen

[chenj63@rpi.edu](mailto:chenj63@rpi.edu)

662049331

IBM Repo: <https://github.com/ceph/ceph>

Forked Repo with ibmceph\_Project2 Branch to work on Project 2 with two Parts: <https://github.com/CJary/ceph>

## Onboarding/Startup ~ Took A While

1. Deciding Between 3 Projects
  - a. Project 1 : Expose cluster config options to Telemetry (Python)
  - b. Project 2 : Add RocksDB metric collection to telemetry (C++ and Python)
    - i. Section 1 : C++ Unit Test (I am currently working on this)
    - ii. Section 2 : Connect the RocksDB Metrics to the telemetry module
2. System Requirement
  - a. Local Set up
  - b. VPN Virtual Machine ~This is the most common as IBM Project will take up too much disk space for most students including myself
    - i. Setting up tunnelblick
    - ii. Creating IBM pad account ~waiting for confirmation
    - iii. Create ssh keys and other information necessary to put on pad and wait for IBM admin to verify.
3. Troubleshooting:
  - a. ~Certain files in the config when setting up the tunnelblick wasn't correctly copied to the parent directory
  - b. Granting access to certain files to let the vpn connect to the tunnelblick
  - c. SSH key was generated in the incorrect way therefore leading to delay in IBM Admin granting access to VR machine

## Additional Links And Information for VPN Mac

- Step 1: Download [Tunnelblick](#)
- Step 2: Once Tunnelblick is downloaded and setup open terminal and run the following commands

```
mkdir /etc/openvpn
cd /etc/openvpn
wget https://filedump.ceph.com/sepia-vpn-client.tar.gz
sudo tar zxvf sepia-vpn-client.tar.gz

# Generate client credentials
# USER should be your desired username and HOST should describe your
# workstation
# e.g., dgalloway@thinkpad

sudo ./sepia/new-client USER@HOST

# Submit the output of this command in your ticket
```
- Step 3: Replace the line `auth-user-pass sepia/secret` with just `auth-user-pass` in `client.conf`, you may have to use `sudo` to edit it.
- Step 4: Follow [Tunnelblick's instructions](#) for adding the config

- Step 5: When logging into the vpn you will be prompted for user/pass, enter username USER@HOST as above, and for password use the secret contents of the file `/etc/openssh/sepia/secret.`, you won't be able to access the vpn yet as you still have to submit the ticket.
- 

### **Once credentials are generated:**

Submit a ticket answering the following questions and wait for approval

1) Desired Username (can be your rcsID id):

2) E-mail address:

3) If you don't already have an established history of code contributions to Ceph, is there an existing community or core developer you've worked with who has reviewed your work and can vouch for your access request?

4) Paste your SSH public key:

Generate one using these steps:

1. Open Terminal

2. Paste the text below, substituting in your GitHub email address.

```
`ssh-keygen -t rsa -C "your_email@example.com"``
```

3. When you're prompted to "Enter a file in which to save the key," press Enter. This accepts the default file location.

```
> Enter a file in which to save the key (/home/you/.ssh/id_rsa): [Press enter]
```

4. At the prompt, type a secure passphrase. you can press Enter if choose to not use passphrase.

5. SSH key is now generated. Copy the contents of `"cat ~/.ssh/id_rsa.pub"` to this tracker.

5) Paste your hashed VPN credentials (Format: user@hostname 22CharacterSalt 65CharacterHashedPassword)

### **Once in the vpn:**

- Step 1: go to terminal and type

```
ssh desiredUsername@smithi061.front.sepia.ceph.com
```

OR

```
ssh desiredUsername@smithi124.front.sepia.ceph.com
```

- (Optional) Step 2: You can use the remote-ssh extension in visual studio code to code in there if you don't want to use VIM.

# RocksDB Metrics File Notes

Flow Control:

Start Of Program:

class RocksDBStore::SocketHook reads the command lines from user

Ex. "dump\_rocksdb\_stats <'telemetry'/'objectstore'/'debug'/'all'> ; dump RocksDB stats");

^Call Method: extracts the correct level provided by the user so in our case it collects telemetry

then mapped to the correct ENUM db\_stats\_mode value which then will be set to telemetry.

Runs db\_stats(db).print(f, mode); afterwards which starts the program to give the RocksDB metrics

HELPER CODE:

```
static constexpr uint32_t deb = 1;          //< db_stats_debug
static constexpr uint32_t obs = 2;          //< db_stats_objectstore
static constexpr uint32_t tel = 4;          //< db_stats_telemetry
static constexpr uint32_t alw = deb + obs + tel; //< always capture
struct db_ticker
{
    uint32_t mode;          //< filter that determines which mode includes this counter
    const char *name;       //< name associated with performance counter
    uint32_t ticker_index = 0; //< RocksDB id to retrieve data
};
```

Helper Function:

Convert the enum mode to the corresponding const int in the helper code

uint32\_t get\_mode\_bits(db\_stats\_mode mode)

```
{
    switch (mode)
    {
        case all:
            return alw;
        case debug:
            return deb;
        case objectstore:
            return obs;
        case telemetry:
            return tel;
        default:
            ceph_assert(false);
    }
}
```

METRICS Call the 3 major metrics main\_stats, histograms, cfstats

class RocksDBStore::db\_stats:

```
run each Print depending on mode:
void print(ceph::Formatter *f, db_stats_mode mode)
{
    f->open_object_section("stats");
    print_main_stats(f, mode);
    print_histograms(f, mode);
    print_cfstats(f, mode);
    f->close_section();
}
```

Unit Test Notes:

-use C++ Framework with Google Test and Google Mock , gtest && gmock

~1)create the unit test -> have to add the file to the executable CMAKEList.txt -> ninja the executable -> find it in the ceph/build/bin to run the file

-meeting commands: ~

1529 git status

1530 cd "/home/jarychen/ceph/src/test/" && g++ perf\_counters.cc -o perf\_counters && "/home/jarychen/ceph/src/test/"perf\_counters

1531 cd ..

1532 cd build

1533 clear

```
1534 ls
1535 ./bin/ceph -s
1536 clear
1537 ninja unittest_perf_counters
1538 clear
1539 ls
1540 ls bin
1541 ./unittest_perf_counters
1542 cd bin
1543 ./unittest_perf_counters
1544 clear
1545 ls
1546 ./unittest_perf_counters
1547 git history
1548 history
```

-Created skeleton and then improve the skeleton to become more in depth by creating Test cases and other helper function similar structure to the perf counter that was provided as a reference to the unit test:

<https://github.com/ceph/ceph/pull/45346>

[https://github.com/ceph/ceph/blob/cadfc96393e3189109a7a314ce3078d02896c5af/src/test/perf\\_counters.cc#L66-L81](https://github.com/ceph/ceph/blob/cadfc96393e3189109a7a314ce3078d02896c5af/src/test/perf_counters.cc#L66-L81)

#### Additional Information For Clarification Related to Unit Test

1. Might have to create tmp/database files
  - a. RocksDB Metrics operates on the aspect of the key/value pair mapping. Storing data on a disk
    - i. To accurately test this might have to create a tmp database to perform write and read operations to make sure it works correctly in Real world situation

## Output for telemetry

```
jarychen@smithi061:~/ceph/build$ ./bin/ceph tell osd.* dump_rocksdb_stats
telemetry
*** DEVELOPER MODE: setting PATH, PYTHONPATH and LD_LIBRARY_PATH
***
2024-10-21T15:23:59.409+0000 7f8a140e9640 -1 WARNING: all dangerous
and experimental features are enabled.
2024-10-21T15:23:59.425+0000 7f8a140e9640 -1 WARNING: all dangerous
and experimental features are enabled.
{
  "main": {
    "rocksdb.blobdb.num.get": 0,
    "rocksdb.blobdb.num.keys.read": 0,
    "rocksdb.blobdb.num.keys.written": 0,
    "rocksdb.blobdb.num.multiget": 0,
    "rocksdb.blobdb.num.next": 0,
    "rocksdb.blobdb.num.prev": 0,
    "rocksdb.blobdb.num.put": 0,
    "rocksdb.blobdb.num.seek": 0,
    "rocksdb.blobdb.num.write": 0,
    "rocksdb.blobdb.write.blob": 0,
    "rocksdb.block.cache.data.hit": 2,
    "rocksdb.block.cache.data.miss": 4,
    "rocksdb.block.cache.filter.add": 4,
    "rocksdb.block.cache.filter.miss": 4,
    "rocksdb.block.cache.hit": 61,
    "rocksdb.block.cache.miss": 12,
    "rocksdb.bytes.read": 671,
    "rocksdb.bytes.written": 15576905,
    "rocksdb.memtable.hit": 0,
    "rocksdb.memtable.miss": 136
  },
  "histogram": {
    "rocksdb.compaction.times.cpu_micros": {
      "p50": 0,
      "p95": 0,
      "p99": 0,
      "avg": 0,
      "std": 0,
      "min": -9223372036854775808,
      "max": 0,
      "count": 0,
      "sum": 0
    },
    "rocksdb.compaction.times.micros": {
      "p50": 0,
      "p95": 0,
      "p99": 0,
      "avg": 0,
      "std": 0,
```

"min": -9223372036854775808,

"max": 0,

"count": 0,

"sum": 0

},

"rocksdb.db.get.micros": {

"p50": 45.724138,

"p95": 96.400000,

"p99": 120,

"avg": 50.161765,

"std": 20.069079,

"min": 8,

"max": 120,

"count": 136,

"sum": 6822

},

"rocksdb.db.write.micros": {

"p50": 138.570774,

"p95": 45856.250000,

"p99": 234066.666667,

"avg": 8238.105421,

"std": 40373.066978,

"min": 38,

"max": 425551,

"count": 3339,

"sum": 27507034

},

"rocksdb.sst.read.micros": {

"p50": 3,

"p95": 9.840000,

"p99": 26,

"avg": 5.375000,

"std": 4.626126,

"min": 3,

"max": 26,

"count": 24,

"sum": 129

},

"rocksdb.wal.file.sync.micros": {

"p50": 124210.526316,

"p95": 371062.500000,

"p99": 423316,

"avg": 155723.381503,

"std": 90683.241709,

"min": 32924,

"max": 423316,

"count": 173,

"sum": 26940145

}

	<pre>    },     "columns": {       "all_columns": {         "sum": {           "KeyDrop": 0,           "KeyIn": 0,           "NumFiles": 4         },         "total_slowdown": 0,         "total_stop": 0       }     }   } }</pre>
Output for all	

## Output for debug

```
jarychen@smithi061:~/ceph/build$ ./bin/ceph tell osd.* dump_rocksdb_stats
debug
*** DEVELOPER MODE: setting PATH, PYTHONPATH and LD_LIBRARY_PATH
***
2024-10-21T15:27:11.310+0000 7f55fe67b640 -1 WARNING: all dangerous
and experimental features are enabled.
2024-10-21T15:27:11.322+0000 7f55fe67b640 -1 WARNING: all dangerous
and experimental features are enabled.
{
  "main": {
    "rocksdb.blobdb.num.get": 0,
    "rocksdb.blobdb.num.keys.read": 0,
    "rocksdb.blobdb.num.keys.written": 0,
    "rocksdb.blobdb.num.multiget": 0,
    "rocksdb.blobdb.num.next": 0,
    "rocksdb.blobdb.num.prev": 0,
    "rocksdb.blobdb.num.put": 0,
    "rocksdb.blobdb.num.seek": 0,
    "rocksdb.blobdb.num.write": 0,
    "rocksdb.blobdb.write.blob": 0,
    "rocksdb.block.cache.bytes.read": 7576,
    "rocksdb.block.cache.bytes.write": 2640,
    "rocksdb.block.cache.data.add": 4,
    "rocksdb.block.cache.data.add.redundant": 0,
    "rocksdb.block.cache.data.bytes.insert": 1664,
    "rocksdb.block.cache.data.hit": 2,
    "rocksdb.block.cache.data.miss": 4,
    "rocksdb.block.cache.filter.add": 4,
    "rocksdb.block.cache.filter.add.redundant": 0,
    "rocksdb.block.cache.filter.bytes.evict": 0,
    "rocksdb.block.cache.filter.bytes.insert": 480,
    "rocksdb.block.cache.filter.hit": 45,
    "rocksdb.block.cache.filter.miss": 4,
    "rocksdb.block.cache.hit": 61,
    "rocksdb.block.cache.index.add": 4,
    "rocksdb.block.cache.index.add.redundant": 0,
    "rocksdb.block.cache.index.bytes.evict": 0,
    "rocksdb.block.cache.index.bytes.insert": 496,
    "rocksdb.block.cache.index.hit": 14,
    "rocksdb.block.cache.index.miss": 4,
    "rocksdb.block.cache.miss": 12,
    "rocksdb.bloom.filter.full.positive": 3,
    "rocksdb.bloom.filter.full.true.positive": 3,
    "rocksdb.bloom.filter.micros": 0,
    "rocksdb.bloom.filter.prefix.checked": 0,
    "rocksdb.bloom.filter.prefix.useful": 0,
    "rocksdb.bloom.filter.useful": 42,
    "rocksdb.bytes.read": 671,
    "rocksdb.bytes.written": 15576905,
```



```
"rocksdb.compaction.cancelled": 0,
"rocksdb.compaction.key.drop.new": 0,
"rocksdb.compaction.key.drop.obsolete": 0,
"rocksdb.compaction.key.drop.range_del": 0,
"rocksdb.compaction.key.drop.user": 0,
"rocksdb.compaction.optimized.del.drop.obsolete": 0,
"rocksdb.compaction.range_del.drop.obsolete": 0,
"rocksdb.compact.read.bytes": 0,
"rocksdb.compact.read.marked.bytes": 0,
"rocksdb.compact.read.periodic.bytes": 0,
"rocksdb.compact.read.ttl.bytes": 0,
"rocksdb.compact.write.bytes": 4192,
"rocksdb.compact.write.marked.bytes": 0,
"rocksdb.compact.write.periodic.bytes": 0,
"rocksdb.compact.write.ttl.bytes": 0,
  "rocksdb.l0.hit": 3,
  "rocksdb.l0.num.files.stall.micros": 0,
  "rocksdb.l0.slowdown.micros": 0,
  "rocksdb.l1.hit": 0,
  "rocksdb.l2andup.hit": 0,
  "rocksdb.memtable.compaction.micros": 0,
  "rocksdb.memtable.hit": 0,
  "rocksdb.memtable.miss": 136,
  "rocksdb.merge.operation.time.nanos": 0,
  "rocksdb.no.file.closes": 0,
  "rocksdb.no.file.errors": 0,
  "rocksdb.no.file.opens": 4,
  "rocksdb.wal.bytes": 15576905,
  "rocksdb.wal.synced": 173
},
"histogram": {
  "rocksdb.compaction.times.cpu_micros": {
    "p50": 0,
    "p95": 0,
    "p99": 0,
    "avg": 0,
    "std": 0,
    "min": -9223372036854775808,
    "max": 0,
    "count": 0,
    "sum": 0
  },
  "rocksdb.compaction.times.micros": {
    "p50": 0,
    "p95": 0,
    "p99": 0,
    "avg": 0,
    "std": 0,
    "min": -9223372036854775808,
```

```
"max": 0,
"count": 0,
"sum": 0
},
"rocksdb.db.flush.micros": {
  "p50": 0,
  "p95": 0,
  "p99": 0,
  "avg": 0,
  "std": 0,
  "min": -9223372036854775808,
  "max": 0,
  "count": 0,
  "sum": 0
},
"rocksdb.db.get.micros": {
  "p50": 45.724138,
  "p95": 96.400000,
  "p99": 120,
  "avg": 50.161765,
  "std": 20.069079,
  "min": 8,
  "max": 120,
  "count": 136,
  "sum": 6822
},
"rocksdb.db.multiget.micros": {
  "p50": 0,
  "p95": 0,
  "p99": 0,
  "avg": 0,
  "std": 0,
  "min": -9223372036854775808,
  "max": 0,
  "count": 0,
  "sum": 0
},
"rocksdb.db.seek.micros": {
  "p50": 10,
  "p95": 171,
  "p99": 171,
  "avg": 31.750000,
  "std": 53.072474,
  "min": 4,
  "max": 171,
  "count": 8,
  "sum": 254
},
"rocksdb.db.write.micros": {
```

```
"p50": 138.570774,  
"p95": 45856.250000,  
"p99": 234066.666667,  
"avg": 8238.105421,  
"std": 40373.066978,  
"min": 38,  
"max": 425551,  
"count": 3339,  
"sum": 27507034  
},  
"rocksdb.db.write.stall": {  
  "p50": 0,  
  "p95": 0,  
  "p99": 0,  
  "avg": 0,  
  "std": 0,  
  "min": -9223372036854775808,  
  "max": 0,  
  "count": 0,  
  "sum": 0  
},  
"rocksdb.sst.batch.size": {  
  "p50": 0,  
  "p95": 0,  
  "p99": 0,  
  "avg": 0,  
  "std": 0,  
  "min": -9223372036854775808,  
  "max": 0,  
  "count": 0,  
  "sum": 0  
},  
"rocksdb.sst.read.micros": {  
  "p50": 3,  
  "p95": 9.840000,  
  "p99": 26,  
  "avg": 5.375000,  
  "std": 4.626126,  
  "min": 3,  
  "max": 26,  
  "count": 24,  
  "sum": 129  
},  
"rocksdb.wal.file.sync.micros": {  
  "p50": 124210.526316,  
  "p95": 371062.500000,  
  "p99": 423316,  
  "avg": 155723.381503,  
  "std": 90683.241709,
```

```
"min": 32924,  
"max": 423316,  
"count": 173,  
"sum": 26940145  
}  
,  
"columns": {  
  "default": {  
    "sum": {  
      "CompCount": 1,  
      "CompMergeCPU": 0,  
      "KeyDrop": 0,  
      "KeyIn": 0,  
      "NumFiles": 2,  
      "ReadGB": 0,  
      "ReadMBps": 0,  
      "WriteAmp": 1,  
      "WriteGB": 0.000001,  
      "WriteMBps": 0.012640  
    },  
    "IO": {  
      "CompCount": 1,  
      "CompMergeCPU": 0,  
      "KeyDrop": 0,  
      "KeyIn": 0,  
      "NumFiles": 2,  
      "ReadGB": 0,  
      "ReadMBps": 0,  
      "WriteAmp": 1,  
      "WriteGB": 0.000001,  
      "WriteMBps": 0.012640  
    },  
    "level0_numfiles": 0,  
    "level0_numfiles_with_compaction": 0,  
    "level0_slowdown": 0,  
    "level0_slowdown_with_compaction": 0,  
    "slowdown_for_pending_compaction_bytes": 0,  
    "stop_for_pending_compaction_bytes": 0,  
    "total_slowdown": 0,  
    "total_stop": 0  
  },  
  "p": {  
    "sum": {  
      "CompCount": 1,  
      "CompMergeCPU": 0,  
      "KeyDrop": 0,  
      "KeyIn": 0,  
      "NumFiles": 1,  
      "ReadGB": 0,
```

```
"ReadMBps": 0,
"WriteAmp": 1,
"WriteGB": 0.000002,
"WriteMBps": 0.014785
},
"l0": {
  "CompCount": 3,
  "CompMergeCPU": 0,
  "KeyDrop": 0,
  "KeyIn": 0,
  "NumFiles": 3,
  "ReadGB": 0,
  "ReadMBps": 0,
  "WriteAmp": 3,
  "WriteGB": 0.000006,
  "WriteMBps": 0.014785
},
"level0_numfiles": 0,
"level0_numfiles_with_compaction": 0,
"level0_slowdown": 0,
"level0_slowdown_with_compaction": 0,
"slowdown_for_pending_compaction_bytes": 0,
"stop_for_pending_compaction_bytes": 0,
"total_slowdown": 0,
"total_stop": 0
},
"m": {
  "sum": {
    "CompCount": 0,
    "CompMergeCPU": 0,
    "KeyDrop": 0,
    "KeyIn": 0,
    "NumFiles": 0,
    "ReadGB": 0,
    "ReadMBps": 0,
    "WriteAmp": 0,
    "WriteGB": 0,
    "WriteMBps": 0
  },
  "level0_numfiles": 0,
  "level0_numfiles_with_compaction": 0,
  "level0_slowdown": 0,
  "level0_slowdown_with_compaction": 0,
  "slowdown_for_pending_compaction_bytes": 0,
  "stop_for_pending_compaction_bytes": 0,
  "total_slowdown": 0,
  "total_stop": 0
},
"p": {
```

```
"sum": {
  "CompCount": 0,
  "CompMergeCPU": 0,
  "KeyDrop": 0,
  "KeyIn": 0,
  "NumFiles": 0,
  "ReadGB": 0,
  "ReadMBps": 0,
  "WriteAmp": 0,
  "WriteGB": 0,
  "WriteMBps": 0
},
"level0_numfiles": 0,
"level0_numfiles_with_compaction": 0,
"level0_slowdown": 0,
"level0_slowdown_with_compaction": 0,
"slowdown_for_pending_compaction_bytes": 0,
"stop_for_pending_compaction_bytes": 0,
"total_slowdown": 0,
"total_stop": 0
},
"O": {
  "sum": {
    "CompCount": 1,
    "CompMergeCPU": 0,
    "KeyDrop": 0,
    "KeyIn": 0,
    "NumFiles": 1,
    "ReadGB": 0,
    "ReadMBps": 0,
    "WriteAmp": 1,
    "WriteGB": 0.000001,
    "WriteMBps": 0.018514
  },
  "IO": {
    "CompCount": 1,
    "CompMergeCPU": 0,
    "KeyDrop": 0,
    "KeyIn": 0,
    "NumFiles": 1,
    "ReadGB": 0,
    "ReadMBps": 0,
    "WriteAmp": 1,
    "WriteGB": 0.000001,
    "WriteMBps": 0.018514
  },
  "level0_numfiles": 0,
  "level0_numfiles_with_compaction": 0,
  "level0_slowdown": 0,
```

```
    "level0_slowdown_with_compaction": 0,  
    "slowdown_for_pending_compaction_bytes": 0,  
    "stop_for_pending_compaction_bytes": 0,  
    "total_slowdown": 0,  
    "total_stop": 0  
  },  
  "L": {  
    "sum": {  
      "CompCount": 1,  
      "CompMergeCPU": 0,  
      "KeyDrop": 0,  
      "KeyIn": 0,  
      "NumFiles": 0,  
      "ReadGB": 0,  
      "ReadMBps": 0,  
      "WriteAmp": 0,  
      "WriteGB": 0,  
      "WriteMBps": 0  
    },  
    "IO": {  
      "CompCount": 1,  
      "CompMergeCPU": 0,  
      "KeyDrop": 0,  
      "KeyIn": 0,  
      "NumFiles": 0,  
      "ReadGB": 0,  
      "ReadMBps": 0,  
      "WriteAmp": 0,  
      "WriteGB": 0,  
      "WriteMBps": 0  
    },  
    "level0_numfiles": 0,  
    "level0_numfiles_with_compaction": 0,  
    "level0_slowdown": 0,  
    "level0_slowdown_with_compaction": 0,  
    "slowdown_for_pending_compaction_bytes": 0,  
    "stop_for_pending_compaction_bytes": 0,  
    "total_slowdown": 0,  
    "total_stop": 0  
  }  
}
```

## Output for objectstore

```
jarychen@smithi061:~/ceph/build$ ./bin/ceph tell osd.* dump_rocksdb_stats
objectstore
*** DEVELOPER MODE: setting PATH, PYTHONPATH and LD_LIBRARY_PATH
***
2024-10-21T15:27:39.981+0000 7f5ebde3d640 -1 WARNING: all dangerous
and experimental features are enabled.
2024-10-21T15:27:39.993+0000 7f5ebde3d640 -1 WARNING: all dangerous
and experimental features are enabled.
{
  "main": {
    "rocksdb.blobdb.num.get": 0,
    "rocksdb.blobdb.num.keys.read": 0,
    "rocksdb.blobdb.num.keys.written": 0,
    "rocksdb.blobdb.num.multiget": 0,
    "rocksdb.blobdb.num.next": 0,
    "rocksdb.blobdb.num.prev": 0,
    "rocksdb.blobdb.num.put": 0,
    "rocksdb.blobdb.num.seek": 0,
    "rocksdb.blobdb.num.write": 0,
    "rocksdb.blobdb.write.blob": 0,
    "rocksdb.block.cache.data.hit": 2,
    "rocksdb.block.cache.data.miss": 4,
    "rocksdb.block.cache.filter.add": 4,
    "rocksdb.block.cache.filter.miss": 4,
    "rocksdb.block.cache.hit": 61,
    "rocksdb.block.cache.miss": 12,
    "rocksdb.bytes.read": 671,
    "rocksdb.bytes.written": 15576905,
    "rocksdb.memtable.hit": 0,
    "rocksdb.memtable.miss": 136
  },
  "histogram": {
    "rocksdb.compaction.times.cpu_micros": {
      "p50": 0,
      "p95": 0,
      "p99": 0,
      "avg": 0,
      "std": 0,
      "min": -9223372036854775808,
      "max": 0,
      "count": 0,
      "sum": 0
    },
    "rocksdb.compaction.times.micros": {
      "p50": 0,
      "p95": 0,
      "p99": 0,
      "avg": 0,
      "std": 0,
```



"min": -9223372036854775808,

"max": 0,

"count": 0,

"sum": 0

},

"rocksdb.db.get.micros": {

"p50": 45.724138,

"p95": 96.400000,

"p99": 120,

"avg": 50.161765,

"std": 20.069079,

"min": 8,

"max": 120,

"count": 136,

"sum": 6822

},

"rocksdb.db.write.micros": {

"p50": 138.570774,

"p95": 45856.250000,

"p99": 234066.666667,

"avg": 8238.105421,

"std": 40373.066978,

"min": 38,

"max": 425551,

"count": 3339,

"sum": 27507034

},

"rocksdb.sst.read.micros": {

"p50": 3,

"p95": 9.840000,

"p99": 26,

"avg": 5.375000,

"std": 4.626126,

"min": 3,

"max": 26,

"count": 24,

"sum": 129

},

"rocksdb.wal.file.sync.micros": {

"p50": 124210.526316,

"p95": 371062.500000,

"p99": 423316,

"avg": 155723.381503,

"std": 90683.241709,

"min": 32924,

"max": 423316,

"count": 173,

"sum": 26940145

}

```
},
"columns": {
"default": {
"sum": {
"KeyDrop": 0,
"KeyIn": 0,
"NumFiles": 2
},
"IO": {
"KeyDrop": 0,
"KeyIn": 0,
"NumFiles": 2
},
},
"total_slowdown": 0,
"total_stop": 0
},
"p": {
"sum": {
"KeyDrop": 0,
"KeyIn": 0,
"NumFiles": 1
},
"IO": {
"KeyDrop": 0,
"KeyIn": 0,
"NumFiles": 3
},
},
"total_slowdown": 0,
"total_stop": 0
},
"m": {
"sum": {
"KeyDrop": 0,
"KeyIn": 0,
"NumFiles": 0
},
},
"total_slowdown": 0,
"total_stop": 0
},
"P": {
"sum": {
"KeyDrop": 0,
"KeyIn": 0,
"NumFiles": 0
},
},
"total_slowdown": 0,
"total_stop": 0
},
"O": {
```

```
"sum": {  
  "KeyDrop": 0,  
  "KeyIn": 0,  
  "NumFiles": 1  
},  
"l0": {  
  "KeyDrop": 0,  
  "KeyIn": 0,  
  "NumFiles": 1  
},  
"total_slowdown": 0,  
"total_stop": 0  
},  
"L": {  
  "sum": {  
    "KeyDrop": 0,  
    "KeyIn": 0,  
    "NumFiles": 0  
  },  
  "l0": {  
    "KeyDrop": 0,  
    "KeyIn": 0,  
    "NumFiles": 0  
  },  
  "total_slowdown": 0,  
  "total_stop": 0  
}  
}
```