

rbd bench- ceph performance tool

Revision history

Version	Date	Changes	Author
0.1	May 26, 2020	initial draft (with requirements and templates for rbd bench performance tool)	Chetan Chib TGI/OLN/CISS/OI

1. Objective

rd bench is ceph performance tool with a set of cli commands to get performance metrics for read and write io types with sequential and random io patterns.

2. SUT

- Redhat Storage Ceph Cluster "ceph version 12.2.12-101.el7cp luminous (stable)"

3. Test setup

- Redhat 7.7 server running rbd bench client
- Add Redhat 7.7 server as client to existing storage cluster
- Ceph backend provides pool/image, rbd_RP/rbdimagecompvol1 of size 100GB

4. Test plan

- 100GB total IO size with below IO patterns at different block sizes
- Sequential writes
 - Random writes
 - Sequential reads
 - Random reads

5. Template syntax for IO patterns

```
rd -p <pool_name> bench <image_name> -io-type <read | write> [-io-size size-in-B/K/M/G/T] [-io-threads num-ios-in-flight] [-io-total size-in-B/K/M/G/T] [-io-pattern <seq | rand>]
```

Generate a series of IOs to the image and measure the IO throughput. If no suffix is given, unit B is assumed for both -io-size and -io-total.

6. Template for IO patterns

6.1 Write Sequential with 8k/16k/32k block size with 32 IO threads

1) io-type write io-size 8K io-threads 32 io-total 100G io-pattern sequential

```
#rbd -p rbd_RP bench rbdimagecompv011 --io-type write --io-size 8K --io-threads
32 --io-total 100G --io-pattern seq
bench type write io_size 8192 io_threads 32 bytes 107374182400 pattern
sequential
...

elapsed: 306 ops: 13107200 ops/sec: 42809.57 bytes/sec: 350696015.81
```

2) io-type write io-size 16K io-threads 32 io-total 100G io-pattern sequential

```
#rbd -p rbd_RP bench rbdimagecompv011 --io-type write --io-size 16K --io-
threads 32 --io-total 100G --io-pattern seq
bench type write io_size 16384 io_threads 32 bytes 107374182400 pattern
sequential
...

elapsed: 181 ops: 6553600 ops/sec: 36107.99 bytes/sec: 591593229.32
```

3) io-type write io-size 32K io-threads 32 io-total 100G io-pattern sequential

```
#rbd -p rbd_RP bench rbdimagecompv011 --io-type write --io-size 32K --io-
threads 32 --io-total 100G --io-pattern seq
bench type write io_size 32768 io_threads 32 bytes 107374182400 pattern
sequential
...

elapsed: 151 ops: 3276800 ops/sec: 21595.92 bytes/sec: 707654997.90
```

6.2 Write Random with 8k/16k/32k block size with 32 IO threads

1) io-type write io-size 8K io-threads 32 io-total 100G io-pattern random

```
#rbd -p rbd_RP bench rbdimagecompv011 --io-type write --io-size 8K --io-threads
32 --io-total 100G --io-pattern rand
bench type write io_size 8192 io_threads 32 bytes 107374182400 pattern random
...

elapsed: 654 ops: 13107200 ops/sec: 20017.87 bytes/sec: 163986384.77
```

2) io-type write io-size 16K io-threads 32 io-total 100G io-pattern random

```
#rbd -p rbd_RP bench rbdimagecompv011 --io-type write --io-size 16K --io-threads 32 --io-total 100G --io-pattern rand
bench type write io_size 16384 io_threads 32 bytes 107374182400 pattern random
...

elapsed: 322 ops: 6553600 ops/sec: 20301.88 bytes/sec: 332626029.51
```

3) io-type write io-size 32K io-threads 32 io-total 100G io-pattern random

```
#rbd -p rbd_RP bench rbdimagecompv011 --io-type write --io-size 32K --io-threads 32 --io-total 100G --io-pattern rand
bench type write io_size 32768 io_threads 32 bytes 107374182400 pattern random
...

elapsed: 162 ops: 3276800 ops/sec: 20144.60 bytes/sec: 660098409.25
```

6.3 Read Sequential with 8k/16k/32k block size with 32 IO threads

1) io-type read io-size 8K io-threads 32 io-total 100G io-pattern sequential

```
#rbd -p rbd_RP bench rbdimagecompv011 --io-type read --io-size 8K --io-threads 32 --io-total 100G --io-pattern seq
bench type read io_size 8192 io_threads 32 bytes 107374182400 pattern sequential
...

elapsed: 2058 ops: 13107200 ops/sec: 6368.04 bytes/sec: 52166971.98
```

2) io-type read io-size 16K io-threads 32 io-total 100G io-pattern sequential

```
#rbd -p rbd_RP bench rbdimagecompv011 --io-type read --io-size 16K --io-threads 32 --io-total 100G --io-pattern seq
bench type read io_size 16384 io_threads 32 bytes 107374182400 pattern sequential
...

elapsed: 1946 ops: 6553600 ops/sec: 3366.73 bytes/sec: 55160475.51
```

3) io-type read io-size 32K io-threads 32 io-total 100G io-pattern sequential

```
#rbd -p rbd_RP bench rbdimagecompv011 --io-type read --io-size 32K --io-threads
32 --io-total 100G --io-pattern seq
bench type read io_size 32768 io_threads 32 bytes 107374182400 pattern
sequential
...
elapsed: 1152 ops: 3276800 ops/sec: 2842.87 bytes/sec: 93155043.55
```

6.4 Read Random with 8k/16k/32k block size with 32 IO threads

1) io-type read io-size 8K io-threads 32 io-total 100G io-pattern random

```
#rbd -p rbd_RP bench rbdimagecompv011 --io-type read --io-size 8K --io-threads
32 --io-total 100G --io-pattern rand
bench type read io_size 8192 io_threads 32 bytes 107374182400 pattern random
...
elapsed: 605 ops: 13107200 ops/sec: 21653.91 bytes/sec: 177388834.09
```

2) io-type read io-size 16K io-threads 32 io-total 100G io-pattern random

```
#rbd -p rbd_RP bench rbdimagecompv011 --io-type read --io-size 16K --io-threads
32 --io-total 100G --io-pattern rand
bench type read io_size 16384 io_threads 32 bytes 107374182400 pattern random
...
elapsed: 325 ops: 6553600 ops/sec: 20145.10 bytes/sec: 330057359.37
```

3) io-type read io-size 32K io-threads 32 io-total 100G io-pattern random

```
#rbd -p rbd_RP bench rbdimagecompv011 --io-type read --io-size 32K --io-threads
32 --io-total 100G --io-pattern rand
bench type read io_size 32768 io_threads 32 bytes 107374182400 pattern random
...
elapsed: 145 ops: 3276800 ops/sec: 22565.84 bytes/sec: 739437338.08
```

7. Quantitative results for templates

We see the following metrics for the given total IO size, OPS - OPS/SEC - BYTES/SEC, for given IO block size.

Template Definition (over IO total size 100GB)	Block Size	OPS	OPS/SEC	BYTES/SEC
RandRead	8k	13107200	21653.91	177388834.1
RandRead	16k	6553600	20145.1	330057359.4
RandRead	32k	3276800	22565.84	739437338.1
RandWrite	8k	13107200	20017.87	163986384.8
RandWrite	16k	6553600	20301.88	332626029.5
RandWrite	32k	3276800	20144.6	660098409.3
SeqRead	8k	13107200	6368.04	52166971.98
SeqRead	16k	6553600	3366.73	55160475.51
SeqRead	32k	3276800	2842.87	93155043.55
SeqWrite	8k	13107200	42809.57	350696015.8
SeqWrite	16k	6553600	36107.99	591593229.3
SeqWrite	32k	3276800	21595.92	707654997.9

8. Appendix - Requirements for client (pre -requisite)

- 1) Client server must have appropriate Operating System same as storage cluster nodes.
- 2) Client server must be having network interface on public network configured in storage cluster nodes.
- 3) Add Client to the existing storage cluster, with RW permissions on mentioned images/pool.
- 4) Map images/pool from storage cluster to Client.
- 5) Create file system on mapped block device and mount it to a directory path in Client