Whamcloud

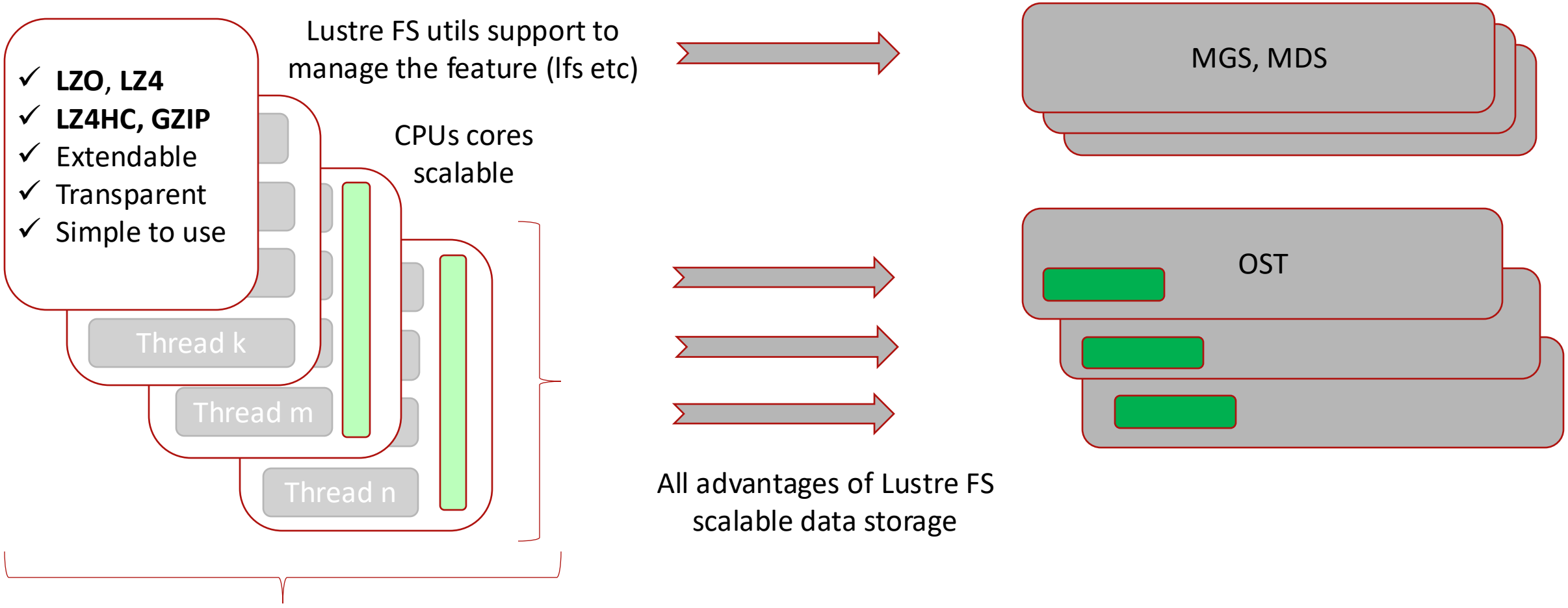
Client-Side Data Compression Best Practices

Artem Blagodarenko

Where's the feature?

- Delays in the **2.17 feature window** opening mean it is available in EXAScaler first
- EXAScaler **6.3.0** had CSDC **technical preview**
- EXAScaler **6.3.1** just released with **CSDC v1**
- EXAScaler **6.3.2** will have **CSDC v2** updates
  - aarch64 clients
  - Better compressibility heuristics
  - Persistent statistics

Client-Side Data Compression



Data is uncompressed in the client's cache and compressed on the OSTs

The worst case

Whamcloud

```
# ls -l /exafs/rawdata -h total
422G
-rw-r--r-- 1 root root 78G Jan 27 08:58 20201028_CCDG_14151_B01_GRM_WGS_2020-08-05_chr10.recalibrated_variants.vcf.gz
-rw-r--r-- 1 root root 75G Jan 27 08:59 20201028_CCDG_14151_B01_GRM_WGS_2020-08-05_chr11.recalibrated_variants.vcf.gz
-rw-r--r-- 1 root root 74G Jan 27 09:00
…
```

Copy files to non-compressed directory by 16 processes DCP on a single client
finished in 66 sec

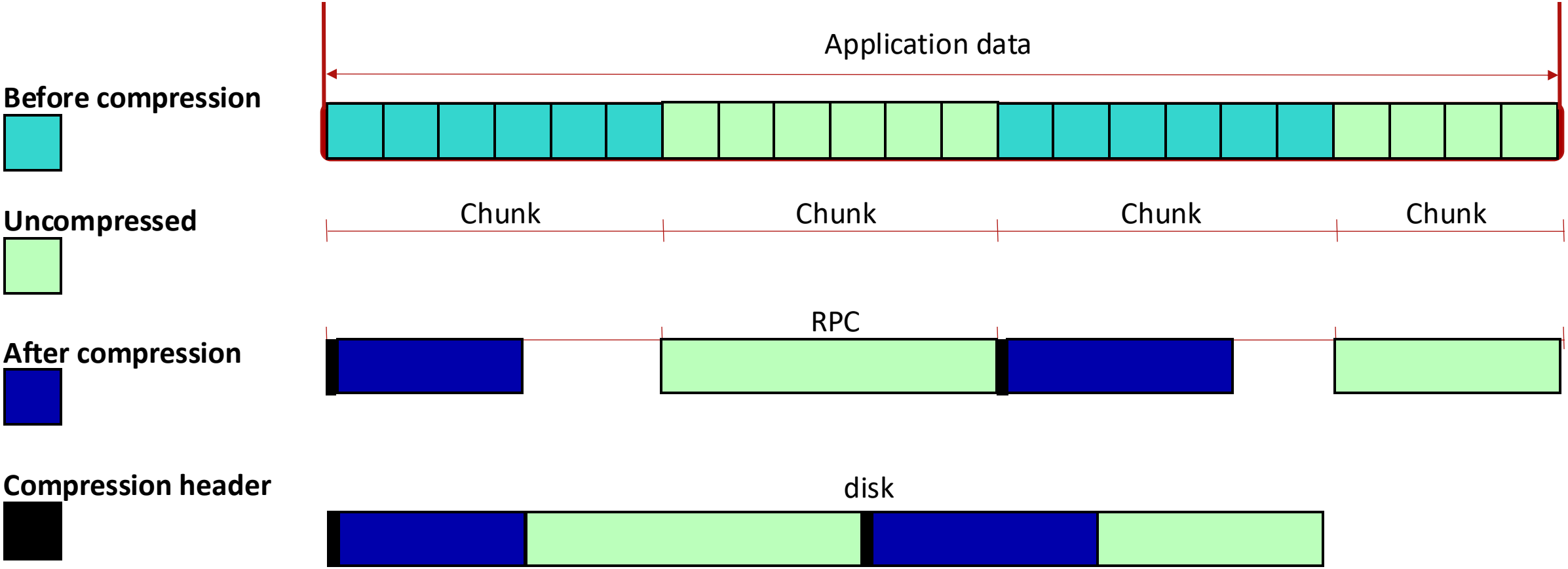
The same copy files to compressed dir (lz4:6)

Transfer speed was just 750MB/sec against 6.5GB/sec (compression off)

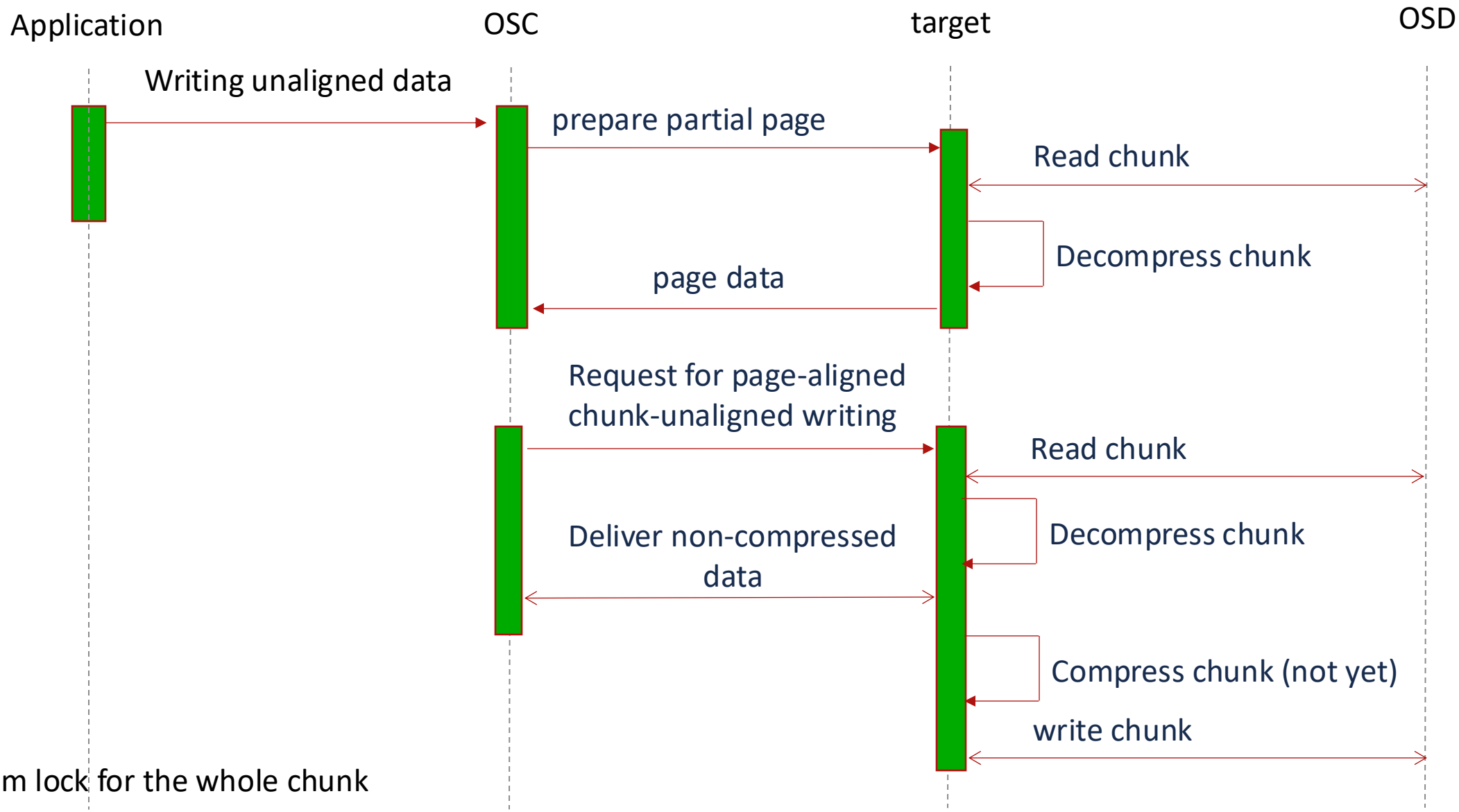
Slower copy speed under lz4 compression directory

Let me explain this case and some others

Data Compression Scheme



Partial Chunk Rewrite Server-Side Solution



*ldlm lock for the whole chunk

Persistent server-side counters

Whamcloud

- OFD saves compression stats every 10 minutes and at unmount
- Load the saved stats at mount time
- Presents overall compression usage
- The file is called "compr_stats“
- Stored in the root directory of underlying fs and uses the YAML format

```
- { name: uncompressed_objects, count:0, min:92, max:0, sum:0 }
- { name: compressed_objects, count:1, min:92233, max:0, sum:0 }
- { name: compr_chunks_read_disk, count:0, min:9223, max:0, sum:0 }
- { name: compr_bytes_read_disk, count:0, min:92233, max:0, sum:0 }
- { name: compr_bytes_read_user, count:0, min:92233, max:0, sum:0 }
- { name: compr_chunks_read_decompress, count:0, min:9, max:0, sum:0 }
- { name: compr_bytes_read_decompresse, count:0, min:9, max:0, sum:0 }
- { name: compr_chunks_write_disk, count:4, min:1, max:8, sum:11 }
- { name: compr_bytes_write_disk, count:4, min:8192, max:1310, sum:20}
- { name: compr_bytes_write_user, count:4, min:8192, max:10486, sum:1}
```

Persistent server-side counters (cont.)

Whamcloud

```
[root@tmp ~]# lctl set_param  obdfilter.*.stats_compr=clear
obdfilter.lustre-OST0001.stats_compr=clear

[root@tmp ~]# dd if=/dev/urandom of=/mnt/lustre/f0 bs=1M count=1 conv=notrunc,fdatasync
[  696.652810] LustreError: 5482:0:(lustre_compr.c:345:compress_chunk()) lustre-OST0001-osc-ffff8c63fc7d5000:
Compression failed, type 5, lvl 8, -22
[  696.653195] LustreError: 5482:0:(lustre_compr.c:345:compress_chunk()) Skipped 15 previous similar messages
1+0 records in
1+0 records out
1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.018989 s, 55.2 MB/s

[root@tmp ~]# du /mnt/lustre/f0
1220      /mnt/lustre/f0
[root@tmp ~]#

[root@tmp ~]# lctl get_param  obdfilter.*.stats_compr
obdfilter.lustre-OST0001.stats_compr=
snapshot_time             1726232644.825403334 secs.nsecs
start_time                1726232637.038225417 secs.nsecs
elapsed_time              7.787177917 secs.nsecs
compr_chunks_write_disk   1 samples [reqs] 16 16 16
compr_bytes_write_disk    1 samples [bytes] 1048576 1048576 1048576
compr_bytes_write_user    1 samples [bytes] 1048576 1048576 1048576
compr_chunks_write_compressed 1 samples [reqs] 0 0 0
compr_chunks_rmw          1 samples [reqs] 0 0 0
compr_bytes_rmw           1 samples [bytes] 0 0 0
compr_bytes_rmw_user      1 samples [bytes] 0 0 0
```

Client-side statistics


```
# lctl set_param osc.*.stats_compr=0

osc.lustre-OST0000-osc-ffff565948a4a000.stats_compr=0
osc.lustre-OST0001-osc-ffff565948a4a000.stats_compr=0

# lfs setstripe -i 0 -c 1 -E -1 -Z lz4:4 --compress-chunk=128k /mnt/lustre/testfile

# cp lustre/tests/AMSR_E_L3_DailyOcean_V05_20111003.hdf /mnt/lustre/testfile

# lctl get_param osc.lustre-OST0000*.stats_compr
osc.lustre-OST0000-osc-ffff565948a4a000.stats_compr=
write_pages_compr: 3066
write_pages_uncompr: 0
write_chunks_compr: 96
write_chunks_no_compr: 0
write_bytes_compr: 5322563
write_bytes_raw: 12558336
write_bytes_incompr: 0
read_pages_compr: 0
read_chunks_compr: 0
read_bytes_compr: 0
read_bytes_raw: 0
```

Preferred case - compressible data into a new file


```
# lctl set_param obdfilter.*.stats_compr=clear
obdfilter.lustre-OST0000.stats_compr=clear
obdfilter.lustre-OST0001.stats_compr=clear

# lctl set_param osc.*.stats_compr=0
osc.lustre-OST0000-osc-ffff565948a4a000.stats_compr=0
osc.lustre-OST0001-osc-ffff565948a4a000.stats_compr=0

# lfs setstripe -i 0 -c 1 -E -1 -Z lz4:6 --compress-chunk=128k /mnt/lustre/best

# dd if=/dev/zero of=/mnt/lustre/best bs=1M count=4 conv=fdatasync
4+0 records in
4+0 records out
4194304 bytes (4.2 MB, 4.0 MiB) copied, 0.0310476 s, 135 MB/s

# du -h /mnt/lustre/best
132K /mnt/lustre/best
```

Preferred case – client-side stats

Whamcloud

```
# lctl get_param osc.lustre-OST0000*.stats_compr
osc.lustre-OST0000-osc-ffff565948a4a000.stats_compr=
write_pages_compr: 1024
write_pages_uncompr: 0
write_chunks_compr: 32
write_chunks_no_compr: 0
write_bytes_compr: 17792
write_bytes_raw: 4194304
write_bytes_incompr: 0
read_pages_compr: 0
read_chunks_compr: 0
read_bytes_compr: 0
read_bytes_raw: 0
```

Preferred case – server-side stats

Whamcloud

```
# lctl get_param obdfilter.*.stats_compr
obdfilter.lustre-OST0000.stats_compr=
snapshot_time           1726794434.509994619 secs.nsecs
start_time              1726794364.042782240 secs.nsecs
elapsed_time            70.467212379 secs.nsecs
compressed_objects      1 samples [reqs]
compr_chunks_write_disk  1 samples [reqs] 32 32 32
compr_bytes_write_disk   1 samples [bytes] 131072 131072 131072
compr_bytes_write_user   1 samples [bytes] 4194304 4194304 4194304
compr_chunks_write_compressed 1 samples [reqs] 32 32 32
compr_chunks_rmw        1 samples [reqs] 0 0 0
compr_bytes_rmw         1 samples [bytes] 0 0 0
compr_bytes_rmw_user    1 samples [bytes] 0 0 0
obdfilter.lustre-OST0001.stats_compr=
snapshot_time           1726794434.510092255 secs.nsecs
start_time              1726794364.042811671 secs.nsecs
elapsed_time            70.467280584 secs.nsecs
```

Less effective usage - partial overwrite

Whamcloud

```
# lctl set_param obdfilter.*.stats_compr=clear
# lctl set_param osc.*.stats_compr=0

# dd if=/dev/random of=/mnt/lustre/best bs=4k count=1 conv=notrunc,fdatasync

# lctl get_param osc.lustre-OST0000*.stats_compr
osc.lustre-OST0000-osc-ffff565948a4a000.stats_compr=
write_pages_compr: 0
write_pages_uncompr: 1
write_chunks_compr: 0
write_chunks_no_compr: 1
write_bytes_compr: 0
write_bytes_raw: 4096
write_bytes_incompr: 4096
read_pages_compr: 0
read_chunks_compr: 0
read_bytes_compr: 0
read_bytes_raw: 0
```

Less effective cases – server-side stats

Whamcloud

```
# lctl get_param obdfilter.*.stats_compr

obdfilter.lustre-OST0000.stats_compr=
snapshot_time           1726796991.989065749 secs.nsecs
start_time              1726796881.696720064 secs.nsecs
elapsed_time            110.292345685 secs.nsecs
compr_chunks_write_disk   1 samples [reqs] 1 1 1
compr_bytes_write_disk    1 samples [bytes] 131072 131072 131072
compr_bytes_write_user    1 samples [bytes] 4096 4096 4096
compr_chunks_write_compressed 1 samples [reqs] 0 0 0
compr_chunks_rmw          1 samples [reqs] 1 1 1
compr_bytes_rmw           1 samples [bytes] 131072 131072 131072
compr_bytes_rmw_user      1 samples [bytes] 4096 4096 4096
decompression_time        1 samples [usecs] 351 351 351
obdfilter.lustre-OST0001.stats_compr=
snapshot_time           1726796991.989145871 secs.nsecs
start_time              1726796881.696756581 secs.nsecs
elapsed_time            110.292389290 secs.nsecs
```

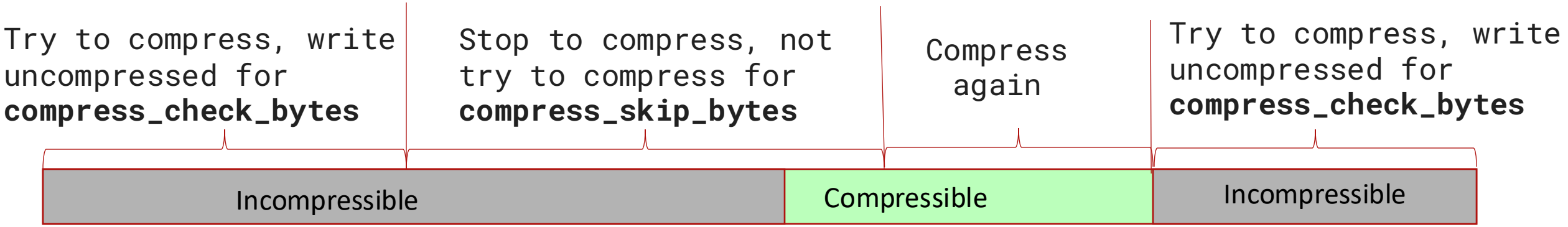
Detecting and avoiding incompressible data

The **reduced_ratio** (original_size/compress_reduced_size) represents the minimum fraction of pages that are compressed out of each chunk

Compressed chunk needs to shrink by at least 1/reduced_ratio blocks for it to be "compressible".

After every **compress_check_bytes** of compressed data written, the file's compressibility would be re-calculated.

If not compressing, recheck after **compress_skip_bytes * factor**
If data still incompressible, double **factor** before next check.



Incompressible data heuristics in operation

Whamcloud

```
# lctl set_param osc.lustre*.compress_check_bytes=1M
# lctl set_param osc.lustre*.compress_skip_bytes=8M
# lctl set_param osc.lustre*.compress_reduced_ratio=16
# lfs setstripe -E-1 -c1 -Z lz4:4 --compress-chunk=64 /mnt/lustre/foofile
# lctl set_param osc.*.stats_compr=0
# cp -a lustre/tests/AMSR_E_L3_DailyOcean_V05_20111003.hdf.bz2 /mnt/lustre/foofile
# lctl get_param osc.lustre*.stats_compr

osc.lustre-OST0000-osc-ffff565948a4a000.stats_compr=
write_pages_compr: 0
write_pages_uncompr: 861
write_chunks_compr: 0
write_chunks_no_compr: 54
write_bytes_compr: 0
write_bytes_raw: 3526254
write_bytes_incompr: 3526254
read_pages_compr: 0
read_chunks_compr: 0
read_bytes_compr: 0
read_bytes_raw: 0
```

```
#du -h /mnt/lustre/foofile
3.4M /mnt/lustre/foofile

#du -h lustre/tests/AMSR_E_L3_DailyOcean_V05_20111003.hdf.bz2
3.4M lustre/tests/AMSR_E_L3_DailyOcean_V05_20111003.hdf.bz2
```

Incompressible data heuristics in operation (cont.)

Whamcloud

```
# lctl set_param osc.*.stats_compr=0
# write compressiable data
# cat lustre/tests/AMSR_E_L3_DailyOcean_V05_20111003.hdf >> /mnt/lustre/foofile
# lctl get_param osc.lustre*.stats_compr

$ sudo lctl get_param osc.lustre*.stats_compr
osc.lustre-OST0000-osc-ffff565948a4a000.stats_compr=
write_pages_compr: 2080
write_pages_uncompr: 1492
write_chunks_compr: 130
write_chunks_no_compr: 93
write_bytes_compr: 3665644
write_bytes_raw: 14629144
write_bytes_incompr: 6111232
read_pages_compr: 0
read_chunks_compr: 0
read_bytes_compr: 0
read_bytes_raw: 0
```

```
$ du -h /mnt/lustre/foofile
13M /mnt/lustre/foofile
```

Recompressing files in-place with **lfs migrate**

```
#   lctl set_param osc.*.stats_compr=0
# lctl set_param obdfilter.*.stats_compr=clear
#   mkdir /mnt/lustre/dir
# lfs setstripe -E eof -Z lz4:1 /mnt/lustre/dir
# cp /mnt/lustre/tests/AMSR_E_L3_DailyOcean_V05_20111003.hdf /mnt/lustre/dir
```

```
#lctl get_param obdfilter.*.stats_compr
obdfilter.lustre-OST0000.stats_compr=
snapshot_time          1726831571.279181285 secs.nsecs
start_time             1726831495.725330980 secs.nsecs
elapsed_time           75.553850305 secs.nsecs
obdfilter.lustre-OST0001.stats_compr=
snapshot_time          1726831571.279318914 secs.nsecs
start_time             1726831495.725413810 secs.nsecs
elapsed_time           75.553905104 secs.nsecs
compressed_objects     1 samples [reqs]
compr_chunks_write_disk   4 samples [reqs] 33 64 225
compr_bytes_write_disk    4 samples [bytes] 1277952 2240512 7331840
compr_bytes_write_user    4 samples [bytes] 2067114 4194304 14625450
compr_chunks_write_compressed 4 samples [reqs] 32 64 220
compr_chunks_rmw       4 samples [reqs] 0 1 1
compr_bytes_rmw        4 samples [bytes] 0 65536 65536
compr_bytes_rmw_user   4 samples [bytes] 0 24576 24576
decompression_time     1 samples [usecs] 271 271 271
```

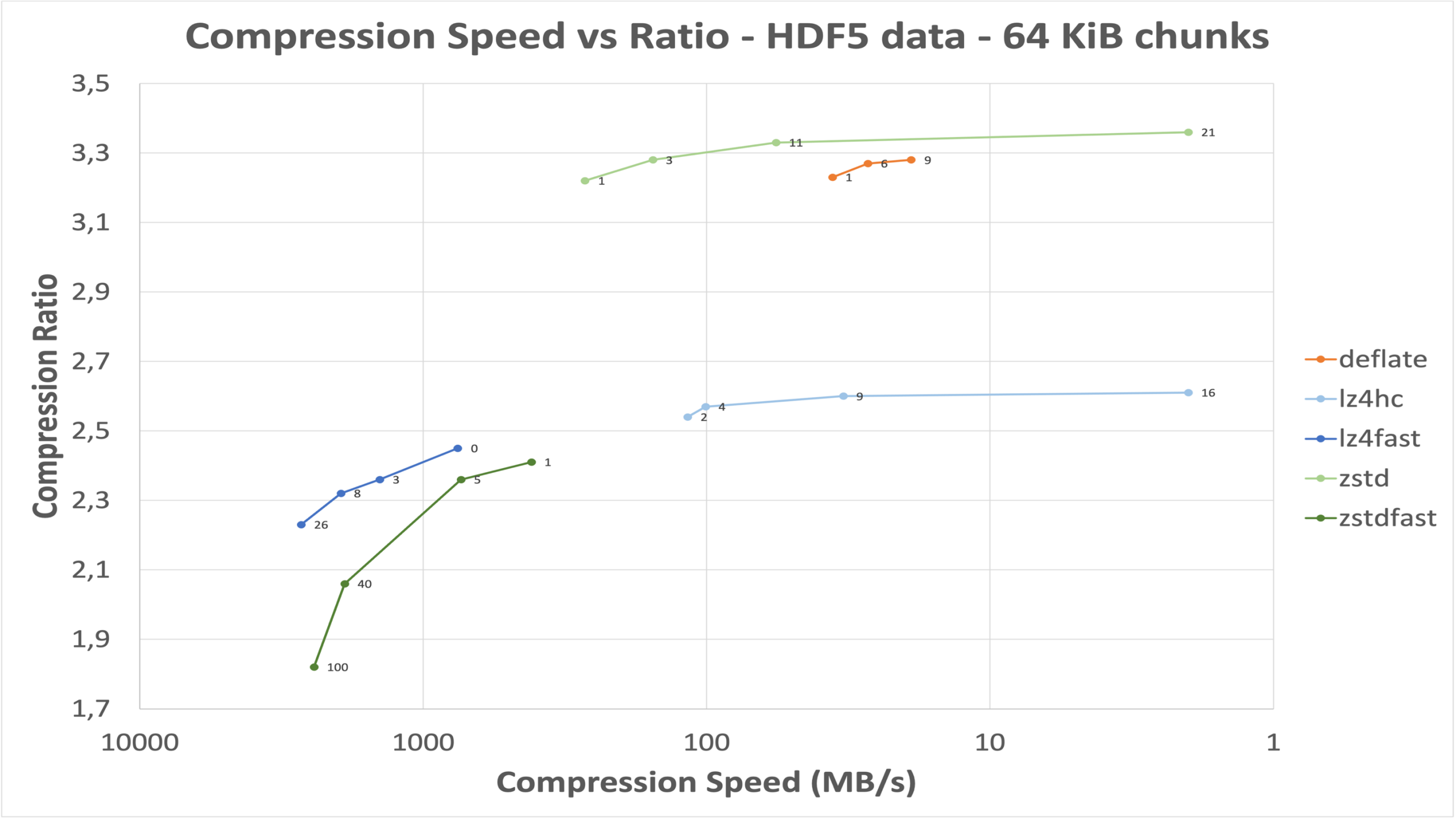
```
$ du -shc /mnt/lustre/dir
7.0M /mnt/lustre/dir
7.0M total
```

Migrating with recompression

sudo find /mnt/lustre/dir/ -type f -exec lfs migrate -E eof -Z lz4:6 {} \;

```
#lctl get_param obdfilter.*.stats_compr
obdfilter.lustre-OST0000.stats_compr=
snapshot_time           1726831760.064316575 secs.nsecs
start_time              1726831642.360612654 secs.nsecs
elapsed_time            117.703703921 secs.nsecs
compressed_objects      1 samples [reqs]
compr_chunks_write_disk 4 samples [reqs] 33 64 225
compr_bytes_write_disk  4 samples [bytes] 851968 2195456 6709248
compr_bytes_write_user  4 samples [bytes] 2067114 4194304 14625450
compr_chunks_write_compressed 4 samples [reqs] 32 64 222
compr_chunks_rmw        4 samples [reqs] 0 1 1
compr_bytes_rmw         4 samples [bytes] 0 65536 65536
compr_bytes_rmw_user    4 samples [bytes] 0 24576 24576
decompression_time      1 samples [usecs] 208 208 208
obdfilter.lustre-OST0001.stats_compr=
snapshot_time           1726831760.064482687 secs.nsecs
start_time              1726831642.360690383 secs.nsecs
elapsed_time            117.703792304 secs.nsecs
compr_chunks_read_disk  4 samples [bytes] 32 64 224
compr_bytes_read_disk   4 samples [bytes] 1212416 2240512 7327744
compr_bytes_read_user   4 samples [bytes] 2043904 4194304 14626816
[ablagodarenko@ubuntu-utm lustre-test]$
```

```
$ du -shc /mnt/lustre/dir
6.5M /mnt/lustre/dir
6.5M total
```


Compression Speed vs Ratio - HDF5 data - 64 KiB chunks
Compression Ratio
3,5
3,3
3,1
2,9
2,7
2,5
2,3
2,1
1,9
1,7
10000
1000
100
10
1
Compression Speed (MB/s)
deflate
lz4hc
lz4fast
zstd
zstdfast
1
3
11
21
1
6
9
2
4
9
16
0
3
8
26
1
5
40
100

Progressive layout

Whamcloud

- Large chunks are compressed better but have more overheads on
- rewriting
- Smaller chunks require fewer overheads, but compression is worse
- The progressive layout allows different compression parameters for different parts of a file.
- Assume that smaller files are accessed (write/read/rewrite) more frequently with small IO
- Large files typically written once with large IOs and stored for a longer time.

```
# lfs setstripe -E 512k -Z zstd:1 --compress-chunk=64k -E 1024k -Z zstd:3 --compress-chunk=1M -E -1 -Z zstd:6 --compress-chunk=1M /mnt/lustre/compressed
```

This example shows how compression level and chunk size are being incremented with incrementing of file size.

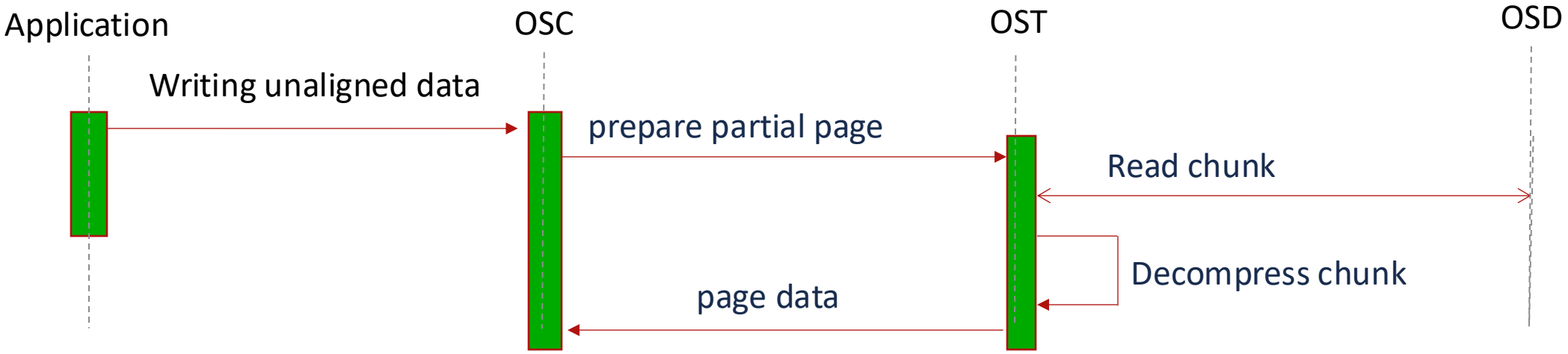
Memory page size and chunk size

Whamcloud

dd if=$src of=$dst bs=4k seek=2 count=1 conv=notrunc,fdatasync

- aarch64 client can have 64k memory page
- Any read/write operation with block < 64k leads to partial read/write
- **Infiniband** rounds up to page size. It can not send smaller amount so even if data compressed full chunk is sent.

```
compr_chunks_write_disk   1 samples [reqs] 1 1 1
compr_bytes_write_disk    1 samples [bytes] 8192 8192 8192
compr_bytes_write_user    1 samples [bytes] 65536 65536 65536
compr_chunks_write_compressed 1 samples [reqs] 1 1 1
compr_chunks_rmw          1 samples [reqs] 0 0 0
compr_bytes_rmw           1 samples [bytes] 0 0 0
compr_bytes_rmw_user      1 samples [bytes] 0 0 0
```



Things for consideration

Whamcloud

- All DIO operations with compressed files will be switched to buffer mode to compress data.
- CSDC uses sparse files to store data on the OST. It is not useful to preallocate space for them. Depending on client parameter either disable **fallocate** (default)or disable compression for file.
- No sparse file read support in LNet, so no reduction of read **network traffic** (will be fixed)
- No **recompression** of data on read-modify-write to avoid overhead, so if a file is updated at small sizes, the updated chunks become uncompressed (can migrate file to recompress afterward).
- No **DoM** support yet (will be fixed)
- No **encryption** support yet (will be fixed)
- **HDD** not recommended (performance is poor due to seeking)
- **T10PI** not integrated yet (server decompression/rewrite changes checksum)
- **GPU Direct** not possible with compression since CPU cannot access data pages in GPU RAM. This is the same with encrypted files and GDS must disabled for those files

Conclusion

Whamcloud

- Compress data that **can be compressed**
- Write **full chunks**
- Adjust **compressibility heuristics** parameters if needed
- Use **buffered** operations
- Use **flash-based** OST storage
- Consider which **other Lustre FS features are** used with CSDC
- Use compression during **migration**
- Use **progressive layout** to enable compression for large files
- Balance client CPU usage by using different compression **algorithms** and **levels**

Whamcloud

Thank You!
Questions?