Benchmark details

Read/Write data stream by cat command

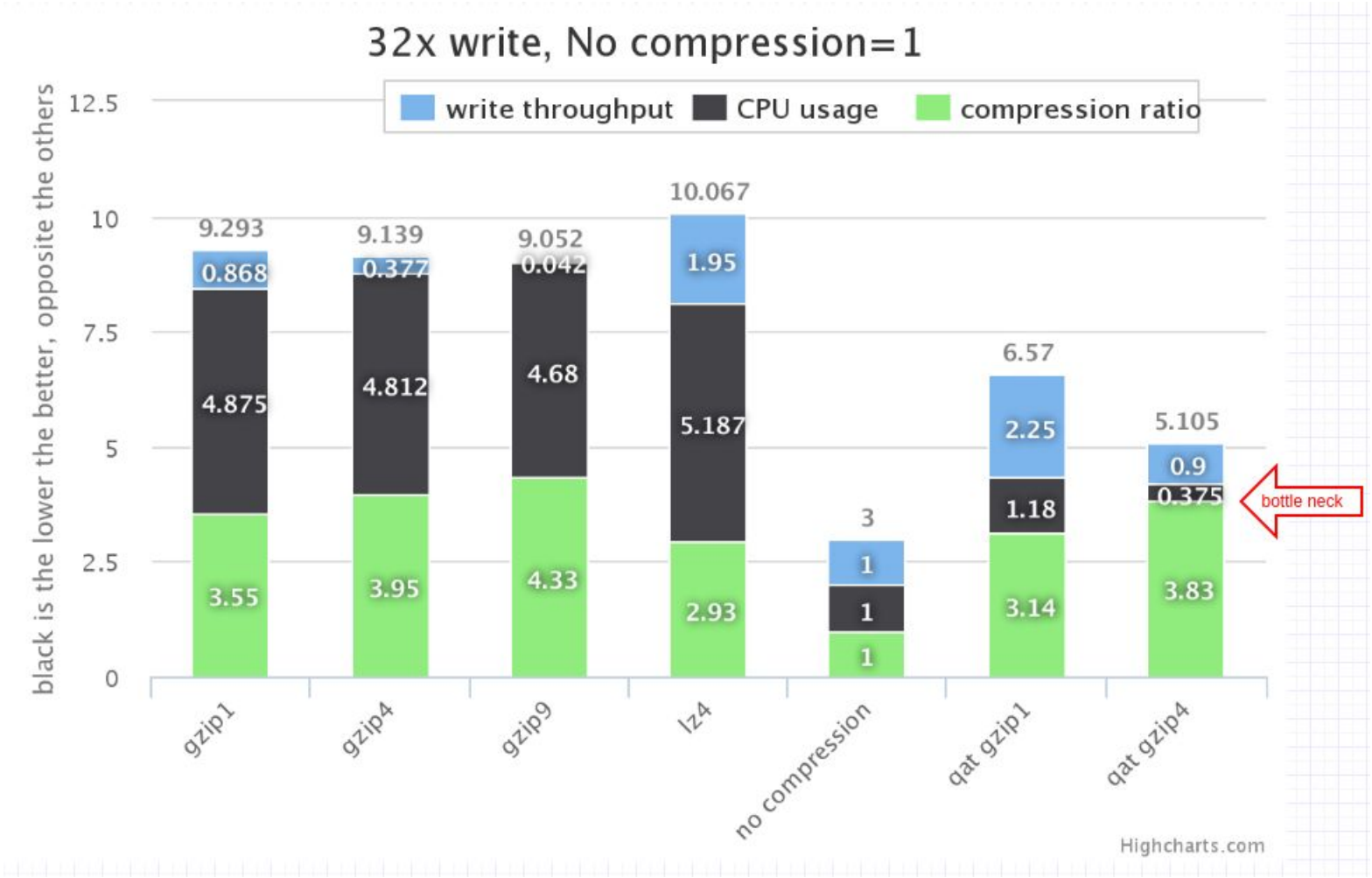
Hardware info

CPU	Produce	Cores and DIMMS	Storage
2 x Xeon E5 2620v3	Dell R430	4GB 2133MHz x8	3x intel DCS 3510 800G
1 x Intel QuickAssist	Adapter 8950 (QAT)		

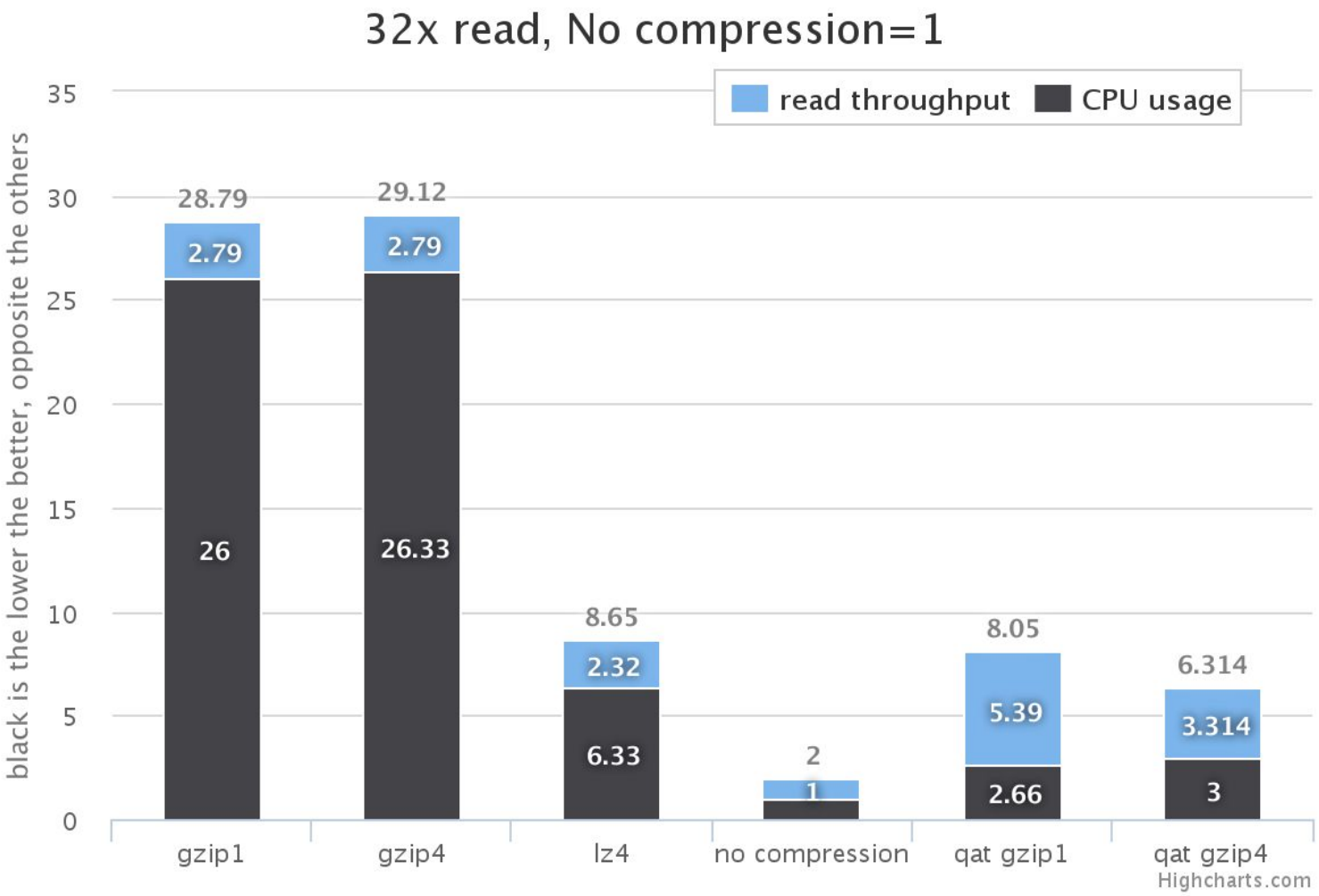
Software info

cat	Cat /dev/shm/xxx > /tank/xx.xx My script will create a processes pool for limit process number 32x means there are 16 processes in processes pool
kernel	2.6.32-642.3.1.el6.x86_64
ZOL	0.6.5.3-1
QAT driver	1.6
Sample file	SRR622461_1.filt.fastq

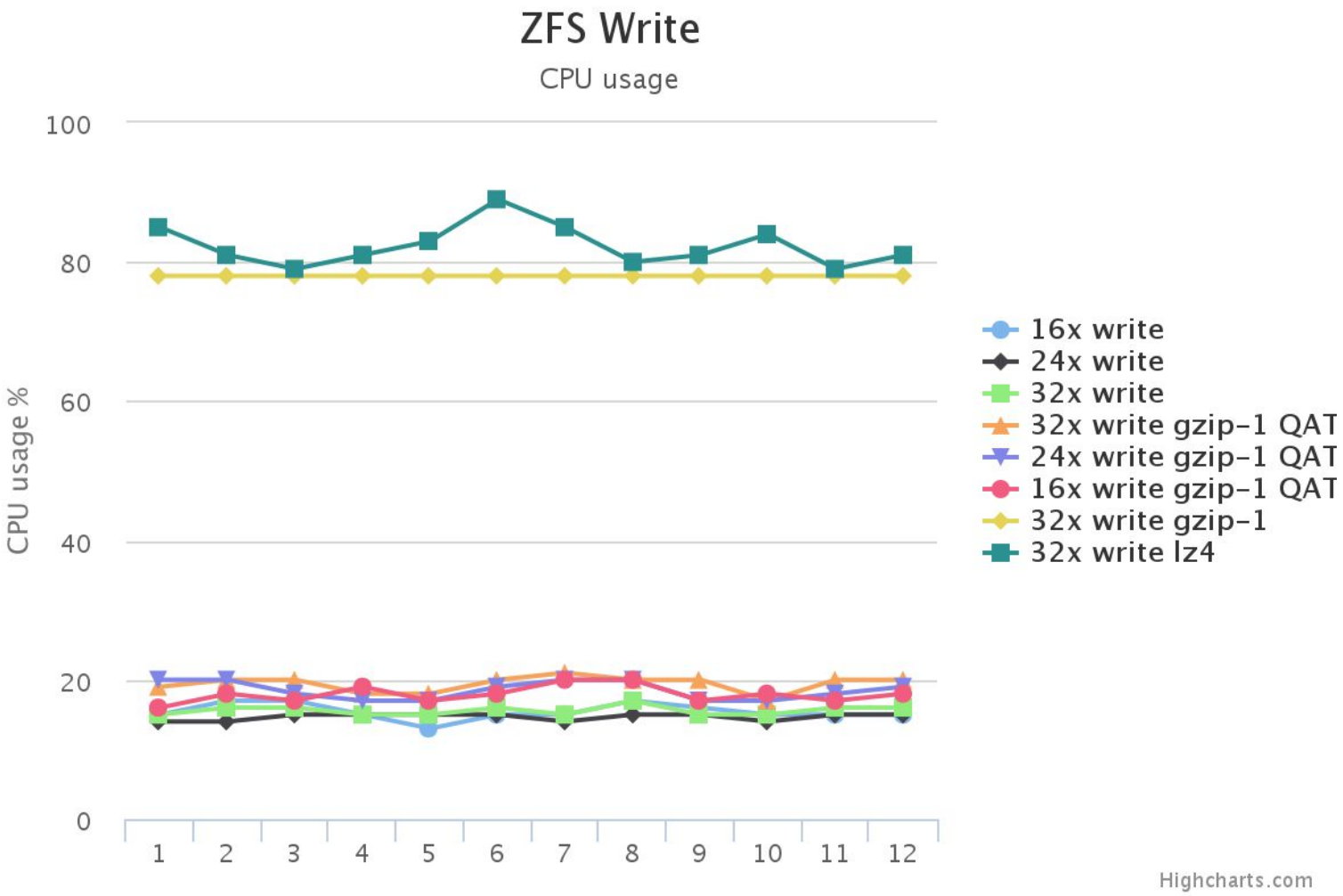
Compare with Write throughput & cpu usage & compression ratio



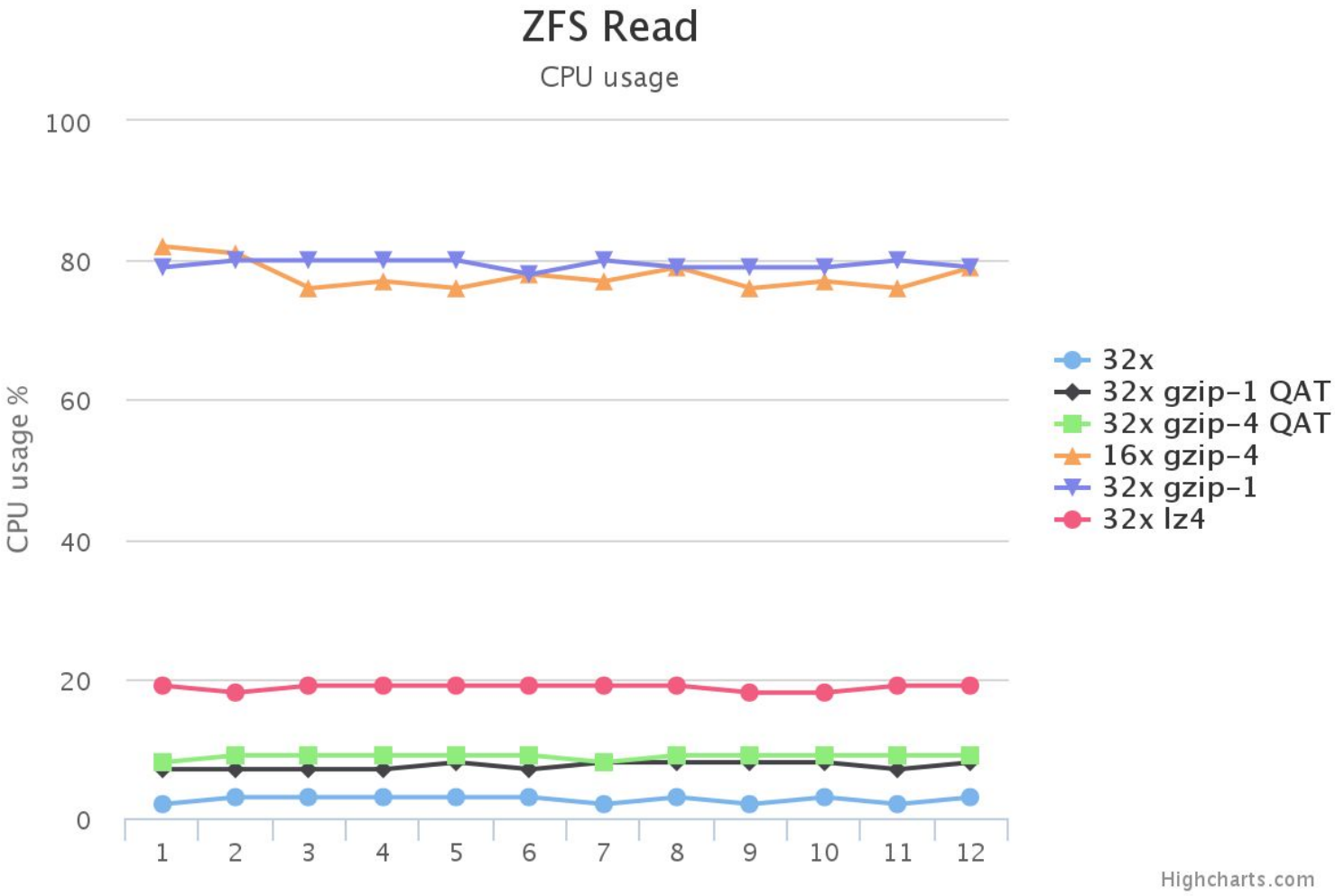
Compare with Read throughput & cpu usage



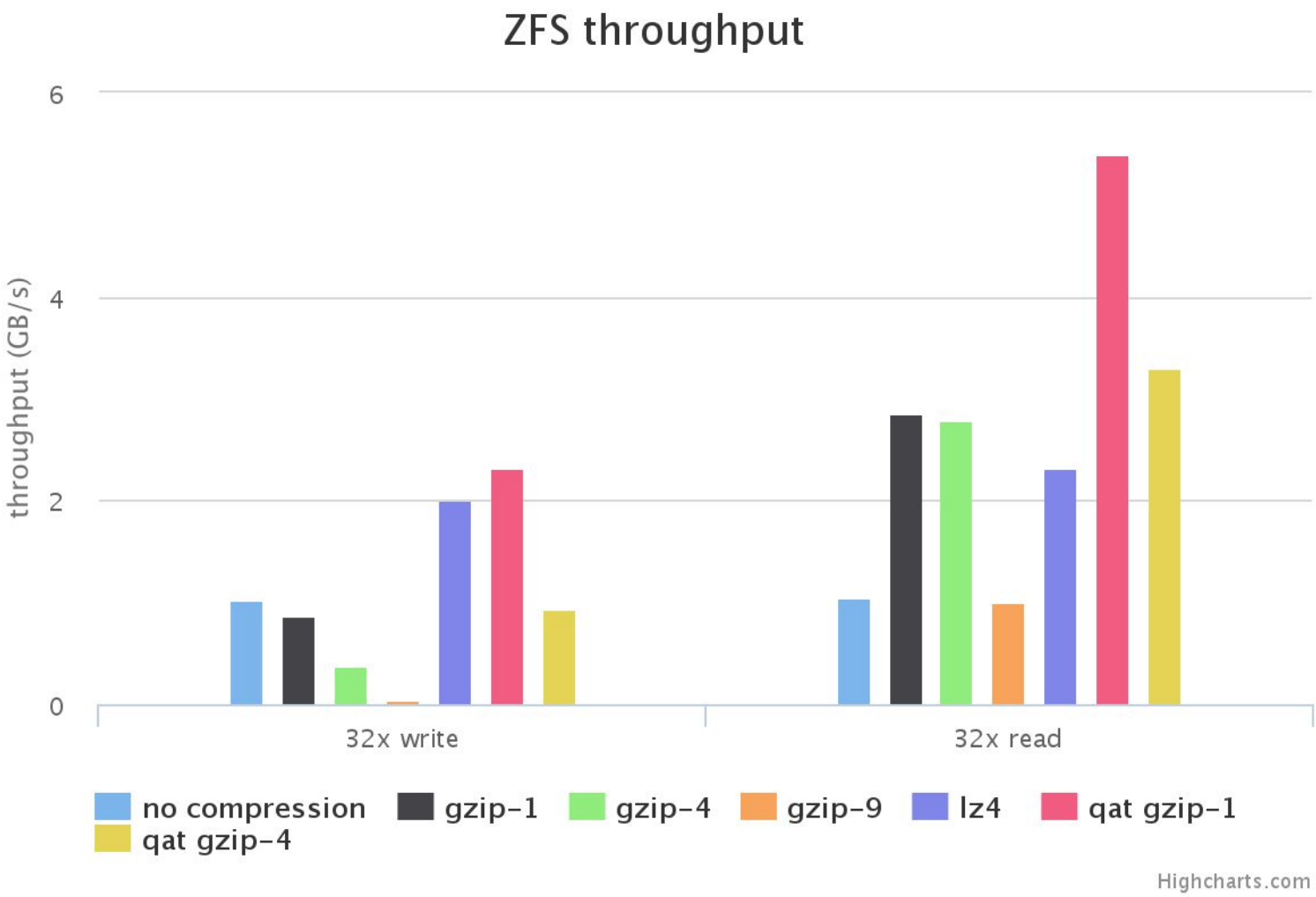
ZFS write CPU usage



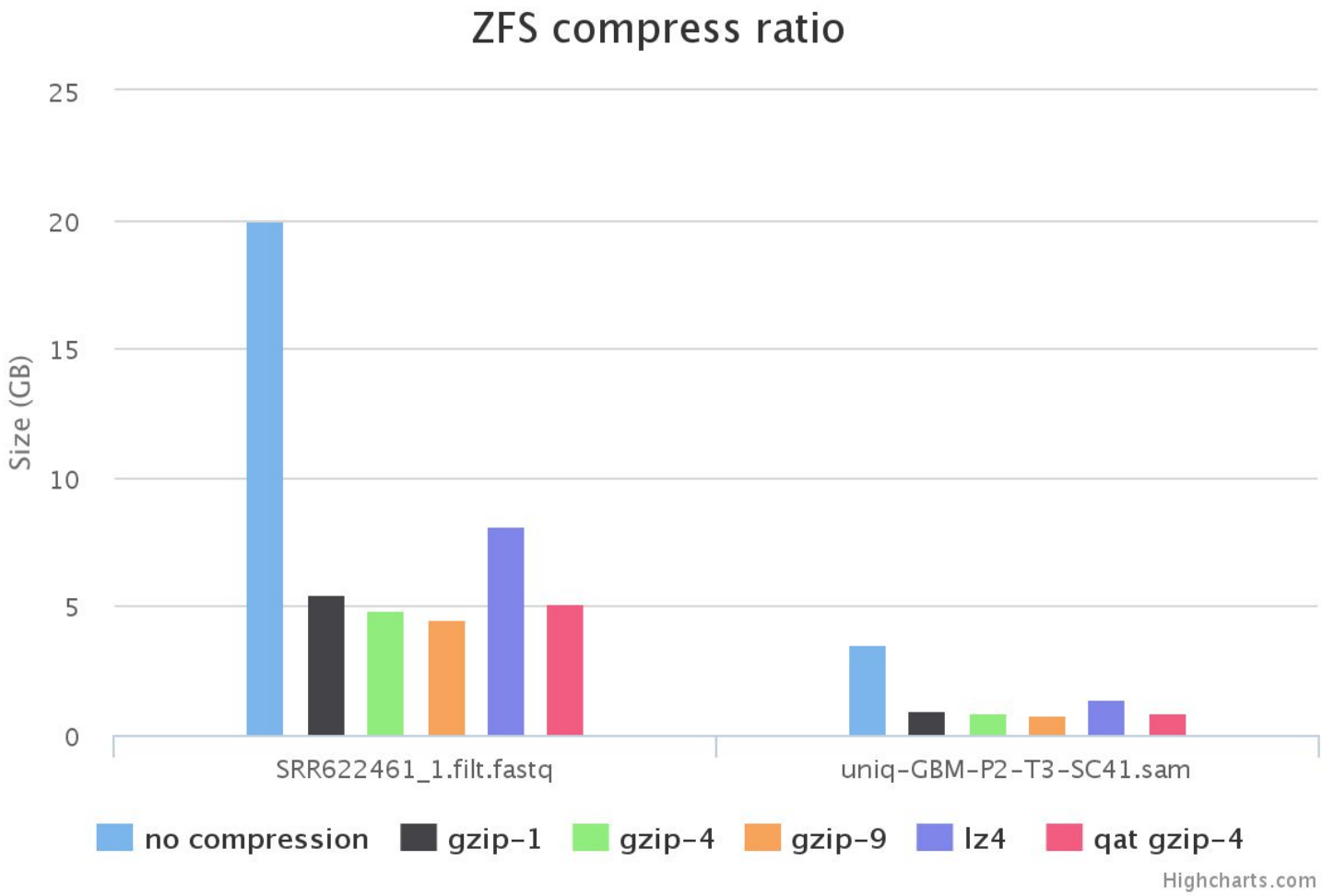
ZFS read CPU usage



ZFS throughput



ZFS compression ratio



No accelerator

```
39.38%  [kernel]                [k] longest_match
29.10%  [kernel]                [k] deflate_slow
10.75%  [kernel]                [k] fill_window
10.11%  [kernel]                [k] compress_block
 3.58%  [kernel]                [k] zlib_tr_tally
 0.76%  [kernel]                [k] copy_user_generic_string
 0.64%  [kernel]                [k] copy_user_enhanced_fast_string
 0.51%  [kernel]                [k] _spin_lock_irqsave
 0.50%  [kernel]                [k] pqdownheap
 0.41%  [kernel]                [k] memcpy
 0.33%  [kernel]                [k] build_tree
 0.24%  [kernel]                [k] fletcher_4_native
 0.11%  [kernel]                [k] mutex_lock
 0.10%  [kernel]                [k] _spin_lock
 0.10%  [kernel]                [k] send_tree
 0.08%  [kernel]                [k] taskq_member
 0.08%  [kernel]                [k] memset
 0.07%  [kernel]                [k] scan_tree
```

With accelerator

```
26.97%  [kernel]                [k] copy_user_generic_string
15.44%  [kernel]                [k] copy_user_enhanced_fast_string
 6.87%  [kernel]                [k] _spin_lock_irqsave
 5.11%  [kernel]                [k] fletcher_4_native
 2.36%  [kernel]                [k] memmove
 1.88%  [kernel]                [k] _spin_lock
 1.69%  [kernel]                [k] mutex_lock
 1.39%  [kernel]                [k] taskq_member
 0.80%  [kernel]                [k] mutex_unlock
 0.78%  [kernel]                [k] __vmalloc_node
 0.76%  [kernel]                [k] list_del
 0.72%  [kernel]                [k] schedule
 0.70%  [kernel]                [k] remove_vm_area
 0.59%  [kernel]                [k] kfree
 0.59%  [kernel]                [k] kmem_cache_alloc_node_trace
 0.55%  [kernel]                [k] _spin_unlock_irqrestore
 0.53%  [kernel]                [k] find_get_page
 0.51%  [kernel]                [k] __wake_up_bit
 0.48%  [kernel]                [k] kmem_cache_free
```

In my opinion, The compression function could in some of cases:

1. Atom/Xeon D cpu + Intel QuickAssist adapter = archive storage solution (posix/object)
2. OpenZFS + SSD (space saving) + Intel QuickAssist adapter = High performance storage solution (eg: lustre,glusterfs,some of object storage base openzfs)
3. Compress database(postgresql/mariadb/elasticsearch)

A problem:

1. It will increase network traffic