BeeGFS® Benchmarks on Huawei High-end Systems

Streaming Performance on ZFS with SSDs and HDDs

Ely de Oliveira, Bomin Song - November 2017 – v 1.2

Index

1 Overview

At-A-Glance

ZFS has become a popular technology for implementing software-based RAID

Over the past decade, ZFS has gained popularity in the HPC community as a technology for implementing software-based RAID volumes. It provides numerous features, such as data protection, data compression, snapshots, as well as powerful management tools, which make it a good option as the underlying file system of BeeGFS storage targets.

This document presents a brief summary of the results of a series of benchmarks executed on BeeGFS services running on high-end systems provided by Huawei Technologies, which use ZFS on the storage targets. It also presents the best system configuration identified during the experiment.

BeeGFS supports ZFS along with other Linux file systems like XFS and ext4

2 Systems Description

Two systems were used in the experiments: one based on SSDs and another one based on HDDs. Both systems had two servers and two client nodes, as illustrated in Figures 1 and 2, and described below.

The two client machines used in the experiment had the following configuration.

- CPU: 2 x Intel E5-2667 V4 processors (3.20GHz GHz).
- Memory: 256 GB RAM.
- Network card: 1 x 100 GB Mellanox EDR InfiniBand MT27700 Family [ConnectX-4].
- OS disks: 2 x 300 GB SAS.
- OS: CentOS 7.3, Linux Kernel 3.10.0-514.21.1.el7

The servers of the SSD-based system had the following configuration.

- CPU: 2 x Intel E5-2667 V4 processors (up to 3.20GHz GHz).
- Memory: 128 GB RAM.
- Network card: 1 x 100 GB Mellanox EDR InfiniBand MT27700 Family [ConnectX-4].
- OS disks: 2 x 300 GB SAS.
- OS: CentOS 7.3, Linux Kernel 3.10.0-514.21.1.el7
- Metadata targets: 4 x Huawei 1.8 TB SSD, forming a single RAID 1 volume.
- Storage targets: 50 x Huawei 1.8 TB SAS SSD, distributed across 2 JBODs with Huawei OceanStor V3 2U25 DAS, connected to both servers via 12 Gbps SAS connectors. Each server had 2 ZFS storage pools (RAID-Z2) of 12 SSDs each. The ZFS version used was 0.6.5.11.

The benchmarks were performed on two High-end systems provided by Huawei Technologies

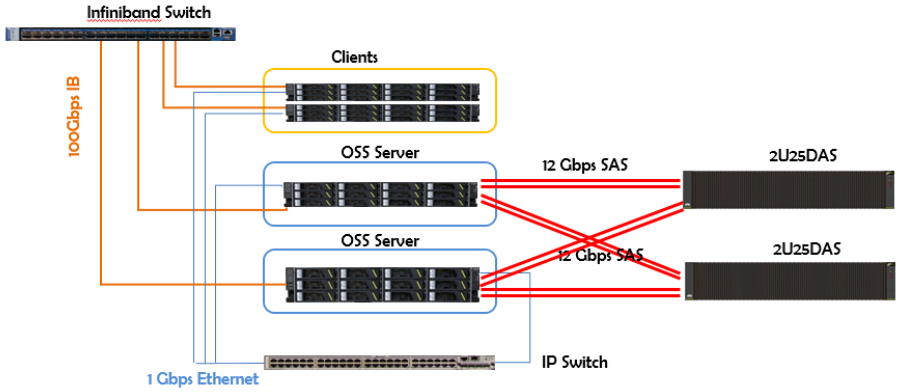

Figure 1- SSD-based System Overview

The servers of the HDD-based system had a slightly different configuration, as follows.

- CPU: 2 x Intel E5-2667 V4 processors (3.20GHz GHz).
- Memory: 128 GB RAM.
- Network card: 1 x 100 Gb Mellanox EDR InfiniBand MT27700 Family [ConnectX-4].
- OS disks: 2 x 300 GB SAS.
- OS: CentOS 7.3, Linux Kernel 3.10.0-514.21.1.el7
- Metadata targets: 4 x Huawei 1.8 TB SSD, forming a single RAID 1 volume.
- Storage targets: 150 x Huawei 6 TB NLSAS HDD, distributed across 2 JBODs with Huawei OceanStor V3 4U75 DAS, connected to both servers via 12 Gbps SAS connectors. Each server had 6 ZFS storage pools (RAID-Z2) of 12 HDDs each. The ZFS version used was 0.6.5.11.

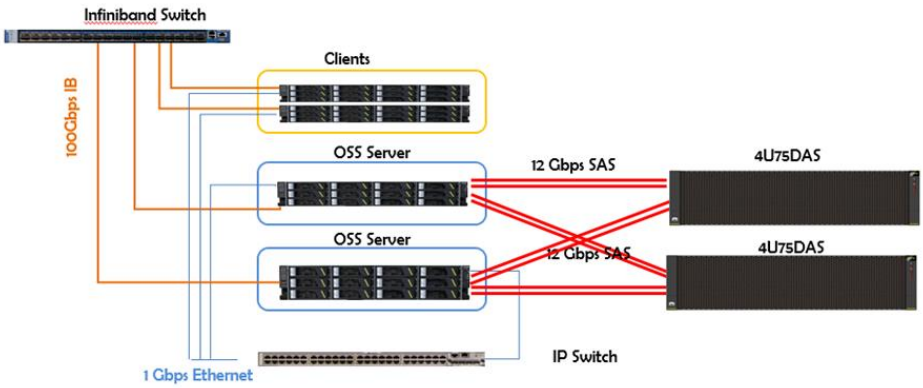

Figure 2: HDD-based System Overview

BeeGFS release 6.14 was used with the following configuration:

- The client services were installed on both client nodes.
- The metadata service was installed on each server and configured to use the RAID 1 of SSD array as metadata target.
- The storage service was also installed on each server and configured to use the ZFS storage pools as storage targets.
- The management service was installed on one of the servers. It used the OS disk as the management target.

IOR 3.0.1 was the benchmark tool used in the experiment to measure the sustained data streaming throughput of the BeeGFS storage service.

3 Tuning

The following tables provide the values of the system and service tuning options that led BeeGFS to achieve the highest performance during the experiment. For a detailed explanation of each option, please go to https://www.beegfs.io/wiki/TableOfContents.

4.1 BeeGFS Storage Service Tuning

The used tuning options are close to the general storage tuning recommendations on the BeeGFS website: Storage Tuning

ZFS Storage Pools Options	Values
Data protection	RAID-Z2
Partition alignment	yes (partitions using entire disks)
Compression (compression)	off
Extended Attributes (xattr)	off
Block size (ashift)	12 (4 KB sector disks)
Record size (recordsize)	4 MB

ZFS Module General Parameters	Values
Maximum record size (zfs_max_recordsize)	4194304
IO scheduler (zfs_vdev_scheduler)	deadline
Read chunk size (zfs_read_chunk_size)	1310720
Data prefetch (zfs_prefetch_disable)	1
Data aggregation limit (zfs_vdev_aggregation_limit)	6
Metadata compression (zfs_mdcomp_disable)	1

	Values	
ZFS Module IO Queues Parameters	**SSD System**	**HDD System**
Minimum scrub requests requests (zfs_vdev_scrub_min_active)	5	6
Maximum scrub requests requests (zfs_vdev_scrub_max_active)	64	36
Minimum sync write requests (zfs_vdev_sync_write_min_active)	24	20
Maximum sync write requests (zfs_vdev_sync_write_max_active)	128	48
Minimum synchronous read requests (zfs_vdev_sync_read_min_active)	24	20
Maximum synchronous read requests (zfs_vdev_sync_read_max_active)	64	48
Minimum asynchronous read requests (zfs_vdev_async_read_min_active)	24	6
Maximum asynchronous read requests (zfs_vdev_async_read_max_active)	64	32
Minimum asynchronous write requests (zfs_vdev_async_write_min_active)	12	6
Maximum asynchronous write requests (zfs_vdev_async_write_max_active)	64	32
Number of requests issued to a single vdev (zfs_vdev_max_active)	3000	3000
Dirty memory flush threshold (zfs_vdev_async_write_active_min_dirty_percent)	20	20

BeeGFS Storage Service Options	**Values**
Worker threads (tuneNumWorkers)	24
Requests in flight to the same server (connMaxInternodeNum)	64
NUMA Zone binding (tuneBindToNumaZone)	0

4.2 BeeGFS Metadata Service Tuning

The used tuning options are close to the general metadata tuning recommendations on the BeeGFS website: ***Metadata Tuning***

ext4 Formatting Options	**Values**
Disk array local Linux file system	ext4
Minimize access times for large directories	-Odir_index
Large inodes	-I 512
Number of inodes	-i 2048
Large journal	-J size=400
Extended attributes	user_xattr
Partition Alignment	yes (partitions using entire disks)

ext4 Mount Options	Values
Last File and Directory Access	noatime, nodiratime
Write Barriers	nobarrier

IO Scheduler Options	Values
Scheduler	deadline
Number of schedulable requests (nr_requests)	1024
Read-ahead data (read_ahead_kb)	4096
Max kilobytes per filesystem request (max_sectors_kb)	512

BeeGFS Meta Service Options	Values
Worker threads (tuneNumWorkers)	24
Requests in flight to the same server (connMaxInternodeNum)	64
NUMA Zone binding (tuneBindToNumaZone)	0

4.3 BeeGFS Client Tuning

Options	Values
Requests in flight to the same server (connMaxInternodeNum)	64
Number of available RDMA buffers (connRDMABufNum)	70
Maximum size of RDMA buffer (connRDMABufSize)	8192
Remote fsync (tuneRemoteFSync)	true

4.4 BeeGFS Striping Settings Tuning

Options	Values
Chunk size (beegfs-ctl pattern option: --chunksize)	512K
Storage targets per file (beegfs-ctl pattern option: --numtargets)	1

The options above had the effect of disabling data striping in the system, as parallelism loses importance in such test scenarios, where all disks are accessed simultaneously. In production systems however, this setting must be enabled.

4 Results

In the data streaming throughput benchmarks, 320 GiB was written and read in each IOR execution. This amount of data is equivalent to 2.5 times the combined RAM size of the servers, with the intention of reducing the impact of data caching on the measurements. Each execution started a different number of processes, ranging from 2 to 128.

4.5 Sustained Data Streaming Throughput – SSD-based System

With 4 SSD-based storage targets, the system delivered 10,756 MiB/s write and 17,046 MiB/s read throughput

Figure 2 and Table 1 show the write and read throughput observed in the SSD-based system. The results show that data streaming throughput scales approximately linearly and stabilize around the maximum values: 10,756 MiB/s write and 17,046 MiB/s read throughput.

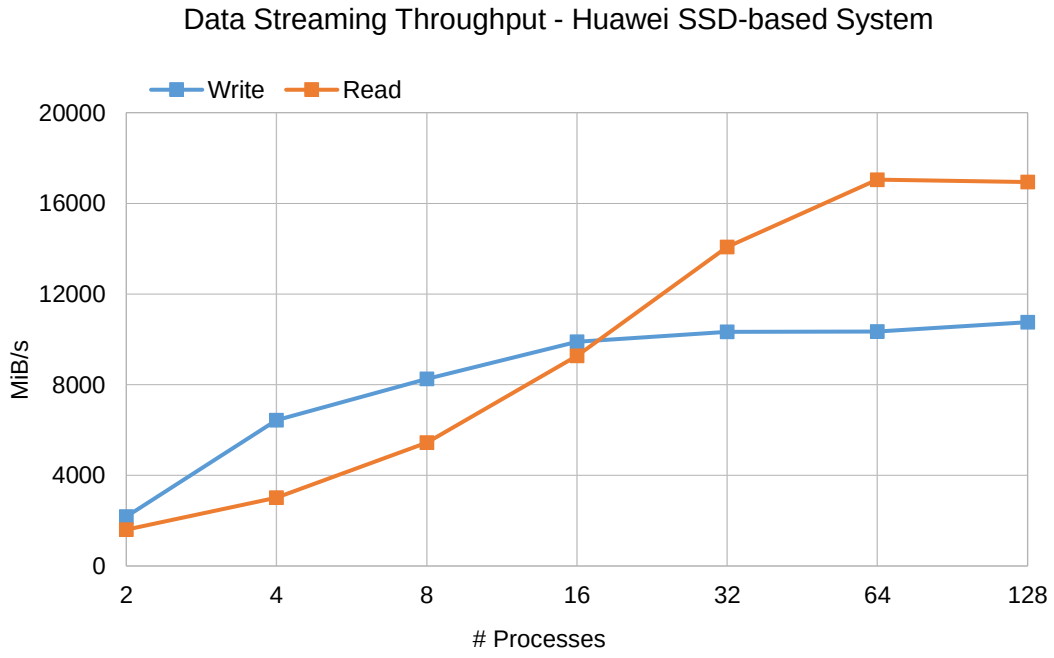

Figure 3 - Read and Write Data Streaming Throughput
Huawei SSD-based System

The hardware capabilities of the Huawei systems, such as powerful CPUs, a good amount of RAM, multiple disk controllers, low-latency network, and flash storage devices played an important role to determine the system throughput.

However, the tuning options used were also very important. As software RAID technologies like ZFS demand more CPU power than hardware-based ones, the tuning options that most

impacted the results were related to the use of CPU. For example, the storage pool record size is the unit that ZFS validates data by calculating checksums. Larger data blocks can reduce the frequency in which checksums are calculated, and speed up write operations.

In addition, pinning all BeeGFS daemons on CPU sockets where both the network interface cards and the RAID cards were connected (option tuneBindToNumaZone) helped to optimize the flow of storage data on the machines, avoiding the CPU sockets interconnect that can sometimes function as a bottleneck.

Processes	Processes / Node	Write (MiB/s)	Read (MiB/s)
2	1	2177	1602
4	2	6431	3011
8	4	8248	5448
16	8	9894	9275
32	16	10333	14068
64	32	10346	17046
128	64	10756	16940

Table 1 - Read and Write Data Streaming Throughput Measurements
Huawei SSD-based System

4.6 Sustained Data Streaming Throughput – HDD-based System

Figure 4 and Table 2 show the write and read throughput observed in the HDD-based system. The results show that data streaming throughput scales approximately linearly and stabilize around the maximum values: 11,444 MiB/s write and 16,625 MiB/s read throughput.

Processes	Processes / Node	Write (MiB/s)	Read (MiB/s)
2	1	2570	2060
4	2	4624	3865
8	4	7669	6994
16	8	10540	10997
32	16	10933	14411
64	32	11302	15903
128	64	11444	16625

Table 2 - Read and Write Data Streaming Throughput Measurements
Huawei HDD-based system

These results look very similar to the ones observed for the SSD-based system, but such initial impression is easily dismissed after we take into consideration that this system had three times more storage devices than the SSD-based system.

With 12 HDD-based storage targets, the system delivered 11,444 MiB/s write and 16,625 MiB/s read throughput

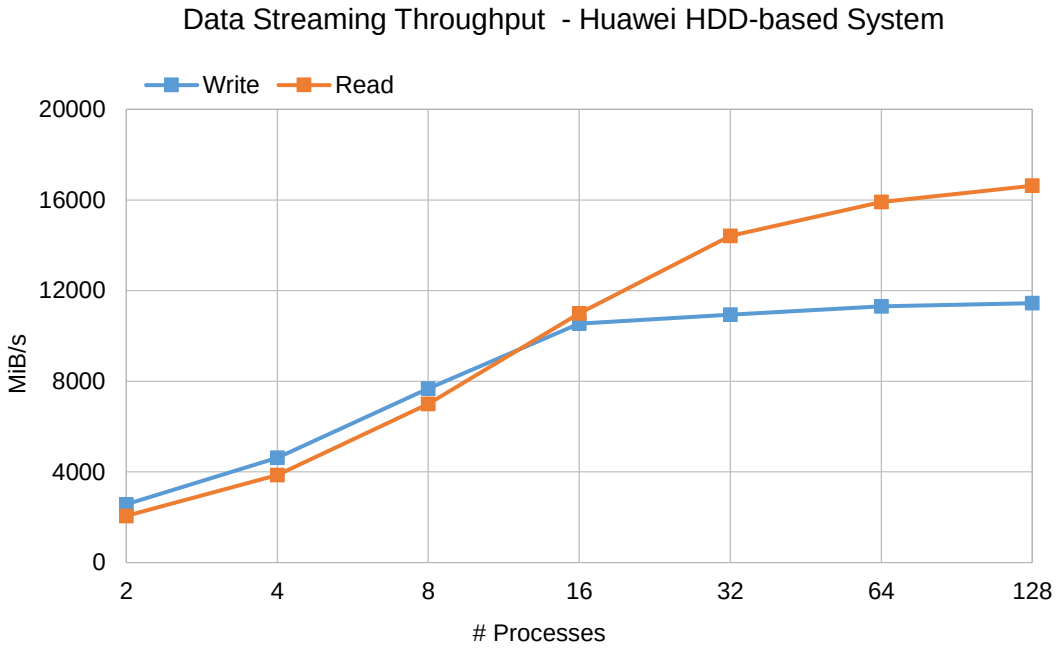

Figure 4 - Read and Write Data Streaming Throughput
Huawei HDD-based System

5 Conclusion

BeeGFS on the Huawei systems showed excellent benchmark results for storage data streaming in both SSD and HDD-based systems.

The highest performance was observed on the SSD-based system, even though it had 3 times less devices than the HDD-based system. This highlights the superiority of flash devices over traditional spinning disks, making it the ideal technology for high-end services and for workloads characterized by non-sequential IO patterns.

6 Commands

This section shows the script that was used on the client nodes to run the data streaming benchmarks.

```
#!/bin/bash

work_dir=~/work
nodes_file=~/nodeslist
results_file="${work_dir}/results.log"
num_procs_array=( 2 4 8 16 32 64 96 128 )

for num_procs in "${num_procs_array[@]}"; do
      rm -rf /mnt/beegfs/*;
      mpirun -ppn ${numproc} -hosts oss02,oss05 \
         /opt/ior -o /mnt/beegfs/file -F -e -g -b \
         $((320/${numproc}))g -t 4M -w -r | tee -a \
         ${results_file}
done
```