SupremeRAID™ BeeGFS™ Performance with GIGABYTE Servers

SupremeRAID™ SR-1010 | GIGABYTE S183-SH0 | BeeGFS 7.3.3

BeeGFS®

Executive Summary

SupremeRAID™ by Graid Technology uses GPU-based acceleration to deliver extremely high RAID performance. Using SupremeRAID™ avoids the inherent performance limitations in other RAID products, including ASIC-based hardware RAID and CPU-based software RAID.

This paper explores how SupremeRAID™ enhances the performance of BeeGFS, a parallel file system, developed and optimized for high-performance computing (HPC). Performance measurements occurred using StorageBench and IOzone. StorageBench is a BeeGFS benchmark that measures the streaming throughput of the underlying file system and devices independent of the network performance. IOzone tests a wide range of IO operations to simulate real-world workloads and is designed to find performance bottlenecks in the overall system.

Testing occurred using GIGABYTE servers operating as two storage nodes and four client nodes. Findings show exceptional storage and BeeGFS performance, as summarized in the following pages, demonstrating that choosing SupremeRAID™ for data protection is a highly effective way to maximize performance.

Performance Benchmark	Theoretical Performance	SupremeRAID™ 5 Reads	SupremeRAID™ 5 Writes
StorageBench	140 GB/s[1]	130.35 GB/s	70 GB/s
IOzone	50 GB/s[2]	45.10 GB/s	42.97 GB/s

1. Two sets of twelve 7 GB/s SSDs configured as four RAID 5 groups. 2. Four 100G Ethernet links for a total of 400G.

The BeeGFS StorageBench benchmark, designed to measure raw storage performance, demonstrates impressive SupremeRAID™ 5 read speeds of 130.35 GB/s and write speeds of 70 GB/s. Also, the StorageBench RAID 5 read performance approaches the theoretical performance limit, with read and write performance significantly higher than the network bottlenecked IOzone benchmark, showcasing the superior storage performance of SupremeRAID™.

The IOzone benchmark, designed to simulate real-world client workloads that include network transmission overhead, is similarly impressive. Read and write speeds reach 45.10 GB/s and 42.97 GB/s, respectively, with 256 threads. Importantly, these figures approach the theoretical limit of a 400G network (50 GB/s), suggesting SupremeRAID™ can almost fully utilize a 400G network composed of four 100G network links.

Introduction

SupremeRAID™

SupremeRAID™ next-generation GPU-accelerated RAID eliminates the traditional RAID bottleneck to unlock the full performance and value of your NVMe SSDs. As the world's fastest RAID cards for PCIe Gen 3, 4, and 5 servers, SupremeRAID™ is designed to deliver superior performance while increasing scalability, improving flexibility, and lowering the total cost of ownership (TCO). A single SupremeRAID™ card blasts performance to up to 28M IOPS and 260 GB/s.

- Flexible & Future Ready – Unmatched flexibility with features added using software only.
- World Record Performance – Delivers the speed to power high-performance applications.
- Liberate CPU Resources – Offloads RAID computations to the SupremeRAID™ GPU card.
- Plug & Play Capability – Add into any open PCIe slot with no cabling re-layout required.
- Highly Scalable Applications – Easily manage up to 32 direct-attached NVMe SSDs.
- User-Friendly Management – Doesn't rely on memory caching to improve performance.

Visit https://www.graidtech.com for more information.

Giga Computing

Giga Computing, a spin-off from GIGABYTE, leverages hardware expertise in product design to operate as an independent entity that can drive innovation and accelerate investments in our core competencies. Their comprehensive product portfolio spans the entire spectrum of workloads, from traditional to emerging applications in HPC, AI, data analytics, 5G/edge, and cloud computing.

Visit https://www.gigacomputing.com for more information.

BeeGFS and StorageBench

The BeeGFS parallel file system, developed and designed by ThinkParQ®, delivers high performance, ease of use, and simple management for performance-oriented environments and workloads. Typical examples include high-performance computing, artificial intelligence, media and entertainment, oil and gas, and life sciences. BeeGFS is often considered to be easier to deploy and manage than other parallel file systems on the market and includes the StorageBench storage benchmarking tool.

Visit https://www.beegfs.io for more information.

IOzone

The IOzone synthetic benchmark tests for file system performance using various operations, including read, re-read, write, re-write, and random mix. Testing occurs depending on the options specified using a command line, with numerous types and combinations of test operations supported.

Visit https://www.iozone.org for more information.

About this Test

Testing Background

Hardware: Storage Nodes (two)

- Server: GIGABYTE S183-SH0-AAV1 x 1
- Processor: Intel® Xeon® Platinum 8468H 48C 2.1GHz x 2
- Memory: Micron MTC20F2085S1RC48BA1 DDR5 32GB 4800MHz x 16
- Network Card: ConnectX-5 Ex MCX556A-EDAT EDR x 1
- SSD: SAMSUNG MZTL23T8HCLS-00A07 3.84TB x 16
- RAID Controller: RAID Controller: SupremeRAID™ SR-1010 x 1

Hardware: Client Nodes (four)

- Server: GIGABYTE H242-Z10 x 4 (four-node system)
- Processor: AMD EPYC 7663 56C x 2
- Memory: Micron HMA82GR7CJR8N-XN DDR4 16GB 3200MHz x 16
- Network Card: ConnectX-5 Ex MCX556A-EDAT EDR x 1

Software: Storage Node

- Operating System: Red Hat Enterprise 8.8
- Kernel: 4.18.0-477.13.1.el8_8.x86_64
- BeeGFS: 7.3.3
- SupremeRAID™ Driver: 1.5.0
- OFED: 5.8-2.0.3.0

Software: Client Nodes

- Operating System: Red Hat Enterprise 8.8
- Kernel: 4.18.0-477.13.1.el8_8.x86_64
- BeeGFS: 7.3.3
- SupremeRAID™ Driver: 1.5.0
- OFED: 5.8-2.0.3.0
- IOzone: 3-506.x86_64

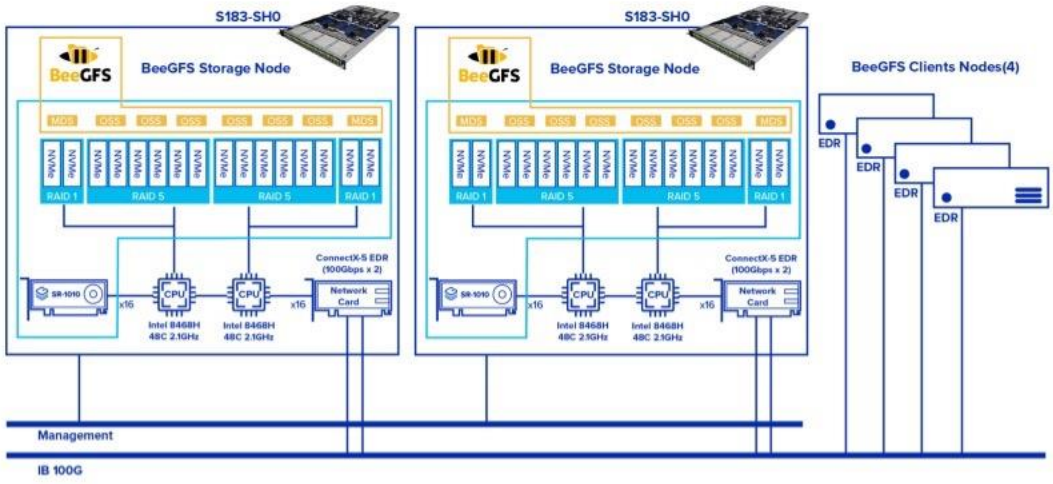
Cluster Architecture

Networking

Each storage node is equipped with a dual-port 100G network card, while each client node features a single-port 100G network card. All two storage nodes and four client nodes are interconnected using a 100G switch.

Storage

Each storage node is equipped with 16 NVMe drives, with eight located at CPU0 and the remaining eight at CPU1. A single SupremeRAID™ SR-1010 RAID controller, positioned at CPU0, manages all 16 NVMe drives. Two Metadata Services (MDS) are set up, each supported by a RAID1 group composed of two drives. Additionally, two RAID 5 groups were constructed, each incorporating six drives. Each RAID 5 group generated three virtual drives for three separate Object Storage Services (OSS). Altogether, the cluster consists of four MDSs and twelve OSSs.



Testing Profiles

BeeGFS StorageBench

Upon the successful construction of the cluster, we utilized the intrinsic BeeGFS StorageBench tool to gauge the performance of the NVMe drives and the RAID controller. The evaluation process commenced with a write test aimed at establishing the test file. This procedure involved a block size of 1M and employed 64 threads. Furthermore, to bypass potential influences from the VFS cache and subsequently reveal the genuine performance capabilities of the storage system, we incorporated the --odirect option.

```sh
$ sudo beegfs-ctl --storagebench --alltargets --write  --blocksize=1m --size=200G --threads=64 --odirect
```

Upon completion of the write test, we transitioned to the read test phase.

```sh
$ sudo beegfs-ctl --storagebench --alltargets --read --blocksize=1m --size=200G --threads=64 --odirect
```

IOzone

To evaluate the cluster's performance under realistic workloads, we utilized IOzone to generate I/O from four client nodes at various I/O depths. This involved conducting both read and write workloads with a 1M block size and a file size of 16GB for each thread. Additionally, the -I option was specified to allow for direct I/O.

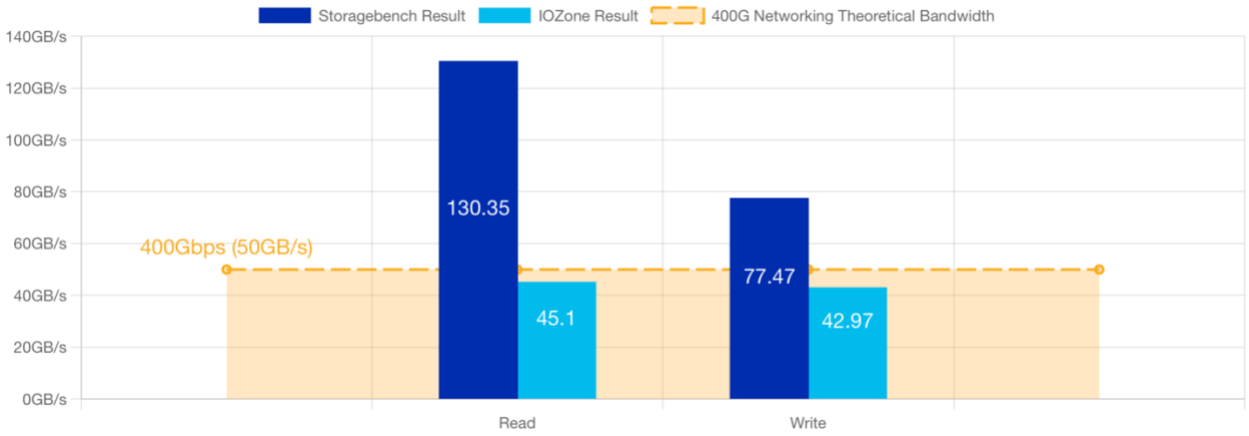
```sh
# $workload: 0(write),1(read)
# $threads: 1, 4, 8, 16, 32, 64, 128, 256
# $threadlist_file: A list for all IOZone workers across 4 client nodes
$ for workload in 0 1; do \
>   for threads in 1 4 8 16 32 64 128 256; do \
>       sudo /opt/iozone/bin/iozone -i "$workload" -MCcew -r 1m -I -s 16g -t "$threads" -+n -+u -+m "$threadlist_file" \
>   ;done \
> ;done
```

Testing Results

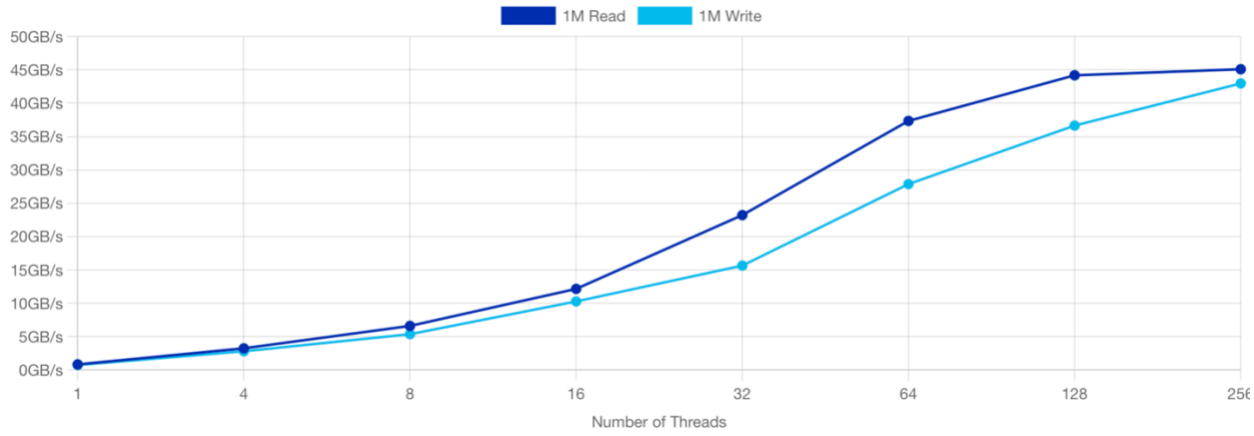
The BeeGFS StorageBench benchmark, designed to measure raw storage performance, demonstrates impressive results under a RAID 5 protected environment. The read and write speeds observed during this benchmark peak at 130.35 GB/s and 70 GB/s across four RAID 5 groups, respectively, as depicted in the chart titled "BeeGFS StorageBench Results vs. IOzone Results". StorageBench RAID 5 read performance approaches the theoretical performance limit, with read and write performance significantly higher than the network bottlenecked IOzone benchmark, showcasing the superior storage performance of SupremeRAID™.

In contrast, the IOzone benchmark simulates real-world client workloads, incorporating network transmission overhead. Although performance in this scenario is lower than the StorageBench results, it remains impressive. The read and write speeds reach 45.10 GB/s and 42.97 GB/s, respectively, with 256 threads. Importantly, these figures approach the theoretical limit of a 400G network (50 GB/s), suggesting SupremeRAID™ can almost fully utilize a 400G network (4 x 100G).

BeeGFS StorageBench Results vs. IOzone Results



IOzone read/write Performance Across Varied Thread Counts



Workloads/Threads	1	4	8	16	32	64	128
1M Read (GB/s)	0.85	3.26	6.62	12.17	23.23	37.35	44.19
1M Write (GB/s)	0.76	2.82	5.38	10.29	15.66	27.88	36.65

Summary

To summarize, SupremeRAID™ delivers high performance under raw storage and real-world workload scenarios. As demonstrated by the BeeGFS StorageBench results, SupremeRAID™ achieves remarkably high storage performance levels under RAID5 protection. Furthermore, the IOzone results reveal that SupremeRAID™ can efficiently handle real-world client workloads while optimally utilizing high-speed network infrastructure.

When coupled with GIGABYTE S183-SH0, we can provide an extremely dense and efficient parallel filesystem solution. The ability to offer up to 398.32TB per 1U when all 32 bays are fully populated makes it an ideal solution for High-Performance Computing (HPC) and Artificial Intelligence (AI) applications. SupremeRAID™, in conjunction with GIGABYTE S183-SH0, brings together exceptional performance and maximum storage efficiency, establishing it as a premier choice for HPC/AI scenarios.

Deployment Details

For all Nodes

Install the RHEL 8.8 on all servers.

Setup Networking

1. Install the OFED package on all servers.

```sh
$ wget   https://content.mellanox.com/ofed/MLNX_OFED-5.8-2.0.3.0/MLNX_OFED_LINUX-5.8-2.0.3.0-rhel8.7-x86_64.iso
$ mkdir /mnt/ofed_iso
$ mount MLNX_OFED_LINUX-5.8-2.0.3.0-rhel8.7-x86_64.iso /mnt/ofed_iso/
$ cd /mnt/ofed_iso/
$ sudo ./mlnxofedinstall  --add-kernel-support --with-nvmf --hpc --distro rhel8.7
```

2. Configure and start an InfiniBand subnet manager on a server.

```sh
$ yum install opensmd
$ /etc/init.d/opensmd start
```

3. Verify the InfiniBand (IB) status.

```sh
$ sudo mlxlink -d <PCIe Address> -p <Port#> -e -m
```

Storage Nodes

Install the SupremeRAID™ Drivers

1. Download the Pre-installer and Installer.

```sh
# Pre-installer
$ wget https://download.graidtech.com/driver/pre-install/graid-sr-pre-installer-1.3.0-43.run
# Installer
$ wget https://download.graidtech.com/driver/sr/linux/1.5.0/alpha/graid-sr-installer-1.5.0-010-567-5-x86_64.run
```

2. Run the pre-installer to install the necessary packages.

```sh
$ chmod +x ./graid-sr-pre-installer-1.3.0-43.run
$ ./graid-sr-pre-installer-1.3.0-43.run
```

3. Execute the installer to install the SupremeRAID™ driver.

```sh
$ chmod +x ./graid-sr-installer-1.5.0-010-567-5-x86_64.run
$ ./graid-sr-installer-1.5.0-010-567-5-x86_64.run
```

4. Apply the license key to activate the SupremeRAID™ service.

```sh
$ sudo graidctl apply license <license key>
```

Install the BeeGFS Packages

1. Add the BeeGFS repo on all servers.

```sh
$ wget -O /etc/yum.repos.d/beegfs_rhel8.repo https://www.beegfs.io/release/beegfs_7.3.3/dists/beegfs-rhel8.repo
```

2. Install the BeeGFS packages.

```sh
$ yum install beegfs-mgmtd beegfs-meta libbeegfs-ib beegfs-storage beegfs-helperd beegfs-utils beegfs-common
```

Setup the RAID Array for BeeGFS

1. Verify the SSD NUMA location. Ensure that eight drives are from NUMA0 and eight from NUMA1.

```sh
$ sudo graidctl list nvme_drive -n 0
$ sudo graidctl list nvme_drive -n 1
```

2. Create 16 NVMe drives as physical drives.

```sh
$ sudo graidctl create physical_drive /dev/nvme0-15
```

3. Construct two RAID1 groups and two RAID5 groups.

```sh
$ sudo graidctl create drive_group RAID1 0-1
$ sudo graidctl create drive_group RAID5 2-7
$ sudo graidctl create drive_group RAID1 8-9
$ sudo graidctl create drive_group RAID5 10-15
```

4. Generate virtual drives for 2 MDSs and 6 OSS

```sh
$ sudo graidctl create virtual_drive 0
$ for i in {0..2}; do sudo graidctl create virtual_drive 1 4TiB; sleep 1 ; done
$ sudo graidctl create virtual_drive 2
$ for i in {0..2}; do sudo graidctl create virtual_drive 3 4TiB; sleep 1 ; done
```

5. Format the virtual drives with the appropriate file systems for MDS (ext4) and OSS (xfs).

```sh
$ for k in 0 2; do sudo mkfs.ext4 /dev/gdg"$k"n1 -F ;done
$ for k in 1 3; do for i in {1..3}; do sudo mkfs.xfs /dev/gdg"$k"n"$i" -f ; done ;done
```

Setup the BeeGFS Management Service

```sh
$ sudo mkdir /data/beegfs/beegfs_mgmtd
$ sudo /opt/beegfs/sbin/beegfs-setup-mgmtd -p /data/beegfs/beegfs_mgmtd -f
```

Setup Multiple BeeGFS MDSs and OSSs in the Storage Node

1. Create 2 MDS folders and 6 OSS folders.

```sh
$ for i in {0..5}; do sudo mkdir /etc/beegfs/storage"$i".d; sudo mkdir -p /beegfs/storage"$i"; done
$ for i in {0..1}; do sudo mkdir /etc/beegfs/meta"$i".d; sudo mkdir -p /beegfs/meta"$i"; done
```

2. Copy the beegfs-meta config file to the MDS folder.

```sh
$ sudo cp beegfs-meta.conf /etc/beegfs/meta0.d/
$ sudo cp beegfs-meta.conf /etc/beegfs/meta1.d/
```

3. Modify the beegfs-meta TCP/UDP port for each MDS to prevent port conflict.

```sh
# /etc/beegfs/meta0.d/beegfs-meta.conf
connMetaPortTCP = 8005
connMetaPortUDP = 8005
# /etc/beegfs/meta0.d/beegfs-meta.conf
connMetaPortTCP = 8015
connMetaPortUDP = 8015
```

4. Copy the beegfs-storage config file to the OSS folder.

```sh
$ for i in {0..5}; do sudo cp beegfs-storage.conf /etc/beegfs/storage"$i".d/; done
```

5. Modify the BeeGFS-storage TCP/UDP port for each OSS to prevent port conflict.

```sh
# /etc/beegfs/storage0.d/beegfs-storage.conf
connStoragePortTCP = 8003
connStoragePortUDP = 8003
# /etc/beegfs/storage1.d/beegfs-storage.conf
connStoragePortTCP = 8013
connStoragePortUDP = 8013
# /etc/beegfs/storage2.d/beegfs-storage.conf
connStoragePortTCP = 8023
connStoragePortUDP = 8023
# /etc/beegfs/storage3.d/beegfs-storage.conf
connStoragePortTCP = 8033
connStoragePortUDP = 8033
# /etc/beegfs/storage4.d/beegfs-storage.conf
connStoragePortTCP = 8043
connStoragePortUDP = 8043
# /etc/beegfs/storage5.d/beegfs-storage.conf
connStoragePortTCP = 8053
connStoragePortUDP = 8053
```

6. Place the interface file in the /etc/beegfs folder.

```sh
$ sudo cp beegfs-storage.nic /etc/beegfs/
```

7. Set the BeeGFS mount point.

```sh
$ count=5;for i in $(seq 0 "$count");do if [ "$i" -ge "$((count/2+1 ))" ];then  sudo mount /dev/gdg3n"$((i-"$((count/
```

8. Initialize the MDSs and OSSs.

```sh
$ sudo /opt/beegfs/sbin/beegfs-setup-meta -f -c /etc/beegfs/meta0.d/beegfs-meta.conf -p /beegfs/meta0 -m 172.16.1.32
$ sudo /opt/beegfs/sbin/beegfs-setup-meta -f -c /etc/beegfs/meta1.d/beegfs-meta.conf -p /beegfs/meta1 -m 172.16.1.32
$ for i in {0..5}; do sudo /opt/beegfs/sbin/beegfs-setup-storage -r -f -c /etc/beegfs/storage"$i".d/beegfs-storage.co
```

9. Start the MDS and OSS services.

```sh
$ sudo systemctl start beegfs-mgmtd.service
$ for i in {0..1}; do sudo systemctl start beegfs-meta@meta"$i"; done
$ for i in {0..5}; do sudo systemctl start beegfs-storage@storage"$i"; done
$ sudo systemctl start beegfs-helperd
```

10. Open the firewall's port.

```sh
# management port
$ firewall-cmd --zone=public --add-port=8006/tcp --permanent
$ firewall-cmd --zone=public --add-port=8006/udp --permanent
$ firewall-cmd --zone=public --add-port=8008/tcp --permanent
$ firewall-cmd --zone=public --add-port=8008/udp --permanent
# metadata port
$ firewall-cmd --zone=public --add-port=8005/tcp --permanent
$ firewall-cmd --zone=public --add-port=8015/tcp --permanent
$ firewall-cmd --zone=public --add-port=8005/udp --permanent
$ firewall-cmd --zone=public --add-port=8015/udp --permanent
# storage port
$ for i in {3..5}0{3..8}; do firewall-cmd --zone=public --add-port=80"$i"/tcp --permanent; done
$ for i in {3..5}0{3..8}; do firewall-cmd --zone=public --add-port=80"$i"/udp --permanent; done
```

11. Reload the firewall service.

```sh
$ firewall-cmd --reload
```

Client Nodes

Install the BeeGFS Packages

1. Add the BeeGFS repo on all servers.

```sh
$ wget -O /etc/yum.repos.d/beegfs_rhel8.repo https://www.beegfs.io/release/beegfs_7.3.3/dists/beegfs-rhel8.repo
```

2. Install the BeeGFS client packages.

```sh
$ yum install beegfs-client beegfs-helperd beegfs-utils libbeegfs-ib beegfs-common
```

Setup the BeeGFS Client

1. Configure the build option on the client servers in the beegfs-client-autobuild.conf file.

```sh
$ buildArgs=-j8 OFED_INCLUDE_PATH=/usr/src/ofa_kernel/x86_64/4.18.0-477.13.1.el8_8.x86_64/include/
```

2. Enforce a rebuild of the client kernel modules.

```sh
$ sudo /etc/init.d/beegfs-client rebuild
```

3. Initialize the client service on the client servers.

```sh
$ sudo /opt/beegfs/sbin/beegfs-setup-client -m 172.17.2.32
```

4. Restart the BeeGFS client service.

```sh
$ sudo systemctl restart beegfs-helperd
$ sudo systemctl restart beegfs-client
```

BeeGFS Tuning

Object Storage Service

```sh
# beegfs-storage.conf
tuneFileReadSize = 1024k
tuneFileWriteSize = 1024k
tuneNumStreamListeners = 8
tuneNumWorkers = 64
tuneWorkerBufSize = 4m
connMaxInternodeNum = 128
tuneBindToNumaZone  = 0 #NIC NUMA Node
tuneUseAggressiveStreamPoll  = true
tuneUsePerUserMsgQueues = true
connRDMABufNum = 22
connRDMABufSize = 65536
```

Metadata Service

```sh
# beegfs-meta.conf
tuneBindToNumaZone = 0 #NIC NUMA Node
```

Client

```sh
# beegfs-client.conf
connMaxInternodeNum = 64
connRDMABufNum = 22
connRDMABufSize = 65536
```

Filesystem

```sh
$ beegfs-ctl --setpattern --numtargets=4 --chunksize=1m /mnt/beegfs
```

References

- BeeGFS Installation Guide
- BeeGFS Multi Modes Configuration Guide
- BeeGFS StorageBench
- IOzone

Conclusion

SupremeRAID™ by Graid Technology uses GPU-based acceleration to deliver extremely high RAID performance. Using SupremeRAID™ avoids the inherent performance limitations in other RAID products, including ASIC-based hardware RAID and CPU-based software RAID. The efficient utilization of SSD performance is improved by SupremeRAID™ software version 1.5, delivering significant gains.

StorageBench and IOzone benchmark testing confirms high storage and BeeGFS performance when using SupremeRAID™ and GIGABYTE servers. StorageBench results demonstrate storage performance matching the aggregate of sixteen SSDs and BeeGFS performance approaching the theoretical limits of 400G.

The performance benefits of using SupremeRAID™ and GIGABYTE for BeeGFS include:

- Up to 130.35 GB/s storage performance.
- Up to 45.10 GB/s BeeGFS performance.

About Graid Technology

Graid Technology, creator of SupremeRAID™ next-generation GPU-based RAID, is led by a team of experts in the storage industry and is headquartered in Silicon Valley, California with an R&D center in Taipei, Taiwan. Designed for performance-demanding workloads, SupremeRAID™ is the fastest NVMe and NVMeoF RAID solution for PCIe Gen 3, 4, and 5 servers. A single SupremeRAID™ card delivers up to 28 million IOPS and up to 260 GB/s and supports up to 32 native NVMe drives, delivering superior NVMe/NVMeoF performance while increasing scalability, improving flexibility, and lowering TCO. For more information on Graid Technology, visit graidtech.com or connect with us on Twitter or LinkedIn.