Overview and requirements

BeeGFS on NetApp with E-Series Storage

NetApp
March 21, 2024

Table of Contents

Overview and requirements

Solution overview

The BeeGFS on NetApp solution combines the BeeGFS parallel file system with NetApp EF600 storage systems for a reliable, scalable, and cost-effective infrastructure that keeps pace with demanding workloads.

This design takes advantage of the performance density delivered by the latest enterprise server and storage hardware and network speeds, requiring file nodes that feature dual AMD EPYC 7003 “Milan” processors and support for PCIe 4.0 with direct connects using 200Gb (HDR) InfiniBand to block nodes that provide end-to-end NVMe and NVMeOF using the NVMe/IB protocol.

NVA program

The BeeGFS on NetApp solution is part of the NetApp Verified Architecture (NVA) program, which provides customers with reference configurations and sizing guidance for specific workloads and use cases. NVA solutions are thoroughly tested and designed to minimize deployment risks and to accelerate time to market.

Use cases

The following use cases apply to the BeeGFS on NetApp solution:

- Artificial Intelligence (AI) including machine learning (ML), deep learning (DL), large-scale natural language processing (NLP), and natural language understanding (NLU). For more information, see BeeGFS for AI: Fact versus fiction.
- High-performance computing (HPC) including applications accelerated by MPI (message passing interface) and other distributed computing techniques. For more information, see Why BeeGFS goes beyond HPC.
- Application workloads characterized by:
  - Reading or writing to files larger than 1GB
  - Reading or writing to the same file by multiple clients (10s, 100s, and 1000s)
- Multi-terabyte or multi-petabyte datasets.
- Environments that need a single storage namespace optimizable for a mix of large and small files.

Benefits

The key benefits of using BeeGFS on NetApp include:

- Availability of verified hardware designs providing full integration of hardware and software components to ensure predictable performance and reliability.
- Deployment and management using Ansible for simplicity and consistency at scale.
- Monitoring and observability provided using the E-Series Performance Analyzer and BeeGFS plugin. For more information, see Introducing a Framework to Monitor NetApp E-Series Solutions.
- High availability featuring a shared-disk architecture that provides data durability and availability.
- Support for modern workload management and orchestration using containers and Kubernetes. For more information, see Kubernetes meet BeeGFS: A tale of future-proof investment.

HA architecture

BeeGFS on NetApp expands the functionality of the BeeGFS enterprise edition by creating a fully integrated solution with NetApp hardware that enables a shared-disk high availability (HA) architecture.

> While the BeeGFS community edition can be used free of charge, the enterprise edition requires purchasing a professional support subscription contract from a partner like NetApp. The enterprise edition allows use of several additional features including resiliency, quota enforcement, and storage pools.

The following figure compares the shared-nothing and shared-disk HA architectures.

[beegfs design image1]

For more information, see Announcing High Availability for BeeGFS Supported by NetApp.

Ansible

BeeGFS on NetApp is delivered and deployed using Ansible automation, which is hosted on GitHub and Ansible Galaxy (the BeeGFS collection is available from Ansible Galaxy and NetApp's E-Series GitHub). Although Ansible is primarily tested with the hardware used to assemble the BeeGFS building blocks, you can configure it to run on virtually any x86-based server using a supported Linux distribution.

For more information, see Deploying BeeGFS with E-Series Storage.

Design generations

The BeeGFS on NetApp solution is currently in its second generational design.

Both the first and second generation include a base architecture that incorporates a BeeGFS file system and an NVMe EF600 storage system. However, the second generation builds on the first to include these additional benefits:

- Double the performance and capacity while adding only 2U of rack space
- High availability (HA) based on a shared-disk, two-tier hardware design
- External qualification for NVIDIA's DGX A100 SuperPOD and NVIDIA BasePOD Architectures

Second generational design

The second generation of BeeGFS on NetApp is optimized to meet the performance requirements of demanding workloads including high-performance computing (HPC) and HPC-style machine learning (ML), deep learning (DL), and similar artificial intelligence (AI) techniques. By incorporating a shared-disk high-availability (HA) architecture, the BeeGFS on NetApp solution also meets the data durability and availability requirements of enterprises and other organizations that cannot afford downtime or data loss as they look for storage that can scale to keep up with their workloads and use cases. This solution has not only been verified by NetApp, but it also passed external qualification as a storage option for the NVIDIA DGX SuperPOD and DGX BasePOD.

First generational design

The first generation of BeeGFS on NetApp was designed for machine learning (ML) and artificial intelligence (AI) workloads using NetApp EF600 NVMe storage systems, the BeeGFS parallel file system, NVIDIA DGX™

A100 systems, and NVIDIA® Mellanox® Quantum™ QM8700 200Gbps IB switches. This design also features 200Gbps InfiniBand (IB) for the storage and compute cluster interconnect fabric to provide a completely IB-based architecture for high-performance workloads.

For more information on the first generation, see NetApp EF-Series AI with NVIDIA DGX A100 Systems and BeeGFS.

Architecture overview

The BeeGFS on NetApp solution includes architectural design considerations used to determine the specific equipment, cabling, and configurations required to support validated workloads.

Building block architecture

The BeeGFS file system can be deployed and scaled in different ways depending on the storage requirements. For example, use cases primarily featuring numerous small files will benefit from extra metadata performance and capacity, whereas use cases featuring fewer large files might favor more storage capacity and performance for actual file contents. These multiple considerations impact different dimensions of the parallel file system deployment, which adds complexity to designing and deploying the file system.

To address these challenges, NetApp has designed a standard building block architecture that is used to scale out each of these dimensions. Typically, BeeGFS building blocks are deployed in one of three configuration profiles:

- A single base building block, including BeeGFS management, metadata, and storage services
- A BeeGFS metadata plus storage building block
- A BeeGFS storage only building block

The only hardware change between these three options is the use of smaller drives for BeeGFS metadata. Otherwise, all configuration changes are applied through software. And with Ansible as the deployment engine, setting up the desired profile for a particular building block makes configuration tasks straightforward.

For further details, see Verified hardware design.

File system services

The BeeGFS file system includes the following main services:

- **Management service.** Registers and monitors all other services.
- **Storage service.** Stores the distributed user file contents known as data chunk files.
- **Metadata service.** Keeps track of the file system layout, directory, file attributes, and so on.
- **Client service.** Mounts the file system to access the stored data.

The following figure shows BeeGFS solution components and relationships used with NetApp E-Series systems.

[beegfs components]

As a parallel file system, BeeGFS stripes its files over multiple server nodes to maximize read/write performance and scalability. The server nodes work together to deliver a single file system that can be

simultaneously mounted and accessed by other server nodes, commonly known as *clients*. These clients can see and consume the distributed file system similarly to a local file system such as NTFS, XFS, or ext4.

The four main services run on a wide range of supported Linux distributions and communicate via any TCP/IP or RDMA-capable network, including InfiniBand (IB), Omni-Path (OPA), and RDMA over Converged Ethernet (RoCE). The BeeGFS server services (management, storage, and metadata) are user space daemons, while the client is a native kernel module (patchless). All components can be installed or updated without rebooting, and you can run any combination of services on the same node.

Verified nodes

The BeeGFS on NetApp solution includes the following verified nodes: the NetApp EF600 storage system (block node) and the Lenovo ThinkSystem SR665 Server (file node).

Block node: EF600 storage system

The NetApp EF600 all-flash array provides consistent, near real-time access to data while supporting any number of workloads simultaneously. To enable fast, continuous feeding of data to AI and HPC applications, EF600 storage systems deliver up to two million cached read IOPS, response times of under 100 microseconds, and 42GBps sequential read bandwidth in one enclosure.

File node: Lenovo ThinkSystem SR665 Server

The SR665 is a two-socket 2U server featuring PCIe 4.0. When configured to meet the requirements of this solution, it provides ample performance to run BeeGFS file services in a configuration well balanced with the availability of throughput and IOPs provided by the direct attached E-Series nodes.

For more information about the Lenovo SR665, see Lenovo's website.

Verified hardware design

The solution's building blocks (shown in the following figure) uses two dual socket PCIe 4.0-capable servers for the BeeGFS file layer and two EF600 storage systems as the block layer.

[beegfs design image2 small]

> Because each building block includes two BeeGFS file nodes, a minimum of two building blocks is required to establish quorum in the failover cluster. While you can configure a two-node cluster, this configuration has limits that might prevent a successful failover to occur. If you require a two-node cluster, you can incorporate a third device as a tiebreaker (however, that design is not covered in this site).

Each building block delivers high availability through a two-tier hardware design that separates fault domains for the file and block layers. Each tier can independently fail over, providing increased resiliency and reducing the risk of cascading failures. Using HDR InfiniBand in conjunction with NVMeOF provides high throughput and minimal latency between file and block nodes, with full redundancy and sufficient link oversubscription to avoid the disaggregated design becoming a bottleneck, even when the system is partially degraded.

The BeeGFS on NetApp solution runs across all building blocks in the deployment. The first building block deployed must run BeeGFS management, metadata, and storage services (referred to as the base building block). All subsequent building blocks are configured through software to run BeeGFS metadata and storage services, or only storage services. The availability of different configuration profiles for each building block enables scaling of file system metadata or storage capacity and performance using the same underlying hardware platforms and building block design.

Up to five building blocks are combined into a standalone Linux HA cluster, ensuring a reasonable number of resources per cluster resource manager (Pacemaker), and reducing the messaging overhead required to keep cluster members in sync (Corosync). A minimum of two building blocks per cluster is recommended to allow enough members to establish quorum. One or more of these standalone BeeGFS HA clusters are combined to create a BeeGFS file system (shown in the following figure) that is accessible to clients as a single storage namespace.

[beegfs design image3]

Although ultimately the number of building blocks per rack depends on the power and cooling requirements for a given site, the solution was designed so that up to five building blocks can be deployed in a single 42U rack while still providing room for two 1U InfiniBand switches used for the storage/data network. Each building block requires eight IB ports (four per switch for redundancy), so five building blocks leaves half the ports on a 40-port HDR InfiniBand switch (like the NVIDIA QM8700) available to implement a fat-tree or similar nonblocking topology. This configuration ensures that the number of storage or compute/GPU racks can be scaled up without networking bottlenecks. Optionally, an oversubscribed storage fabric can be used at the recommendation of the storage fabric vendor.

The following image shows an 80-node fat-tree topology.

[beegfs design image4]

By using Ansible as the deployment engine to deploy BeeGFS on NetApp, administrators can maintain the entire environment using modern infrastructure as code practices. This drastically simplifies what would otherwise be a complex system, allowing administrators to define and adjust configuration all in one place, then ensure it is applied consistently regardless of how large the environment scales. The BeeGFS collection is available from Ansible Galaxy and NetApp's E-Series GitHub.

Technical requirements

To implement the BeeGFS on NetApp solution, make sure your environment meets the technology requirements.

Hardware requirements

The following table lists the hardware components that are required to implement a single second-generation building block design of the BeeGFS on NetApp solution.

> The hardware components used in any particular implementation of the solution might vary based on customer requirements.

Count	Hardware component	Requirements
2	BeeGFS file nodes.	Each file node should meet or exceed the following configuration to achieve expected performance. **Processors:** • 2x AMD EPYC 7343 16C 3.2 GHz. • Configured as two NUMA zones. **Memory:** • 256GB. • 16x 16GB TruDDR4 3200MHz (2Rx8 1.2V) RDIMM-A (prefer more smaller DIMMs over fewer larger DIMMs). • Populated to maximize memory bandwidth. **PCIe Expansion: Four PCE Gen4 x16 slots:** • Two slots per NUMA zone. • Each slot should provide enough power/cooling for the Mellanox MCX653106A-HDAT adapter. **Miscellaneous:** • Two 1TB 7.2K SATA drives (or comparable) configured in RAID 1 for the OS. • 10GbE OCP 3.0 adapter (or comparable) for in-band OS management. • 1GbE BMC with Redfish API for out-of-band server management. • Dual hot swap power supplies and performance fans. • Must support Mellanox optical InfiniBand cables if required to reach storage InfiniBand switches. **Lenovo SR665:** • A custom NetApp model includes the required version of the XClarity controller firmware needed to support dual-port Mellanox ConnectX-6 adapters. Contact NetApp for ordering details.
8	Mellanox ConnectX-6 HCAs (for file nodes).	• MCX653106A-HDAT Host Channel Adapters (HDR IB 200Gb, Dual-port QSFP56, PCIe4.0 x16).
8	1m HDR InfiniBand cables (for file/block node direct connects).	• MCP1650-H001E30 (1m Mellanox Passive Copper cable, IB HDR, up to 200Gbps, QSFP56, 30AWG). The length can be adjusted to account for longer distances between the file and block nodes if required.

Count	Hardware component	Requirements
8	HDR InfiniBand cables (for file node/storage switch connections)	Requires InfiniBand HDR cables (QSFP56 transceivers) of the appropriate length to connect file nodes to storage leaf switches. Possible options include: • MCP1650-H002E26 (2m Mellanox Passive Copper cable, IB HDR, up to 200Gb/s, QSFP56, 30AWG). • MFS1S00-H003E (3m Mellanox active fiber cable, IB HDR, up to 200Gb/s, QSFP56).
2	E-Series block nodes	Two EF600 controllers configured as follows: • Memory: 256GB (128GB per controller). • Adapter: 2-port 200Gb/HDR (NVMe/IB). • Drives: Configured to match desired capacity.

Software requirements

For predictable performance and reliability, releases of the BeeGFS on NetApp solution are tested with specific versions of the software components required to implement the solution.

Software deployment requirements

The following table lists the software requirements deployed automatically as part of the Ansible-based BeeGFS deployment.

Software	Version
BeeGFS	7.2.6
Corosync	3.1.5-1
Pacemaker	2.1.0-8
OpenSM	opensm-5.9.0 (from mlnx_ofed 5.4-1.0.3.0) ⓘ Only required for the direct connects to enable virtualization.

Ansible control node requirements

The BeeGFS on NetApp solution is deployed and managed from an Ansible control node. For more information, see the Ansible documentation.

The software requirements listed in the following tables are specific to the version of the NetApp BeeGFS Ansible collection listed below.

Software	Version
Ansible	2.11 When installed through pip: ansible-4.7.0 and ansible-core < 2.12,>=2.11.6
Python	3.9
Additional Python packages	Cryptography-35.0.0, netaddr-0.8.0
BeeGFS Ansible Collection	3.0.0

File node requirements

Software	Version
RedHat Enterprise Linux	RedHat 8.4 Server Physical with High Availability (2 socket). File nodes require a valid RedHat Enterprise Linux Server subscription and the Red Hat Enterprise Linux High Availability Add-On.
Linux Kernel	4.18.0-305.25.1.el8_4.x86_64
InfiniBand / RDMA Drivers	Inbox
ConnectX-6 HCA Firmware	FW: 20.31.1014
PXE: 3.6.0403	UEFI: 14.24.0013

EF600 block node requirements

Software	Version
SANtricity OS	11.70.2
NVSRAM	N6000-872834-D06.dlp
Drive Firmware	Latest available for the drive models in use.

Additional requirements

The equipment listed in the following table was used for the validation, but appropriate alternatives can be used as needed. In general, NetApp recommends running the latest software versions to avoid unanticipated issues.

Hardware component	Installed software
• 2x Mellanox MQM8700 200Gb InfiniBand switches	• Firmware 3.9.2110

Hardware component	Installed software
1x Ansible control node (virtualized): • Processors: Intel® Xeon® Gold 6146 CPU @ 3.20GHz • Memory: 8GB • Local storage: 24GB	• CentOS Linux 8.4.2105 • Kernel 4.18.0-305.3.1.el8.x86_64 Installed Ansible and Python versions match those in the table above.
10x BeeGFS Clients (CPU nodes): • Processor: 1x AMD EPYC 7302 16-Core CPU at 3.0GHz • Memory: 128GB • Network: 2x Mellanox MCX653106A-HDAT (one port connected per adapter).	• Ubuntu 20.04 • Kernel: 5.4.0-100-generic • InfiniBand Drivers: Mellanox OFED 5.4-1.0.3.0
1x BeeGFS Client (GPU node): • Processors: 2x AMD EPYC 7742 64-Core CPUs at 2.25GHz • Memory: 1TB • Network: 2x Mellanox MCX653106A-HDAT (one port connected per adapter). This system is based on NVIDIAs HGX A100 platform and includes four A100 GPUs.	• Ubuntu 20.04 • Kernel: 5.4.0-100-generic • InfiniBand Drivers: Mellanox OFED 5.4-1.0.3.0

Copyright information

Trademark information