Technical Report

StorageGRID Caching with Varnish

Florian Feldhaus and Steve Pruchniewski, NetApp
August 2019 | TR-4796

Abstract

Varnish Cache provides a high-performance cache layer that can be used to accelerate the read performance of NetApp® StorageGRID® storage software.

NetApp®

TABLE OF CONTENTS

LIST OF TABLES

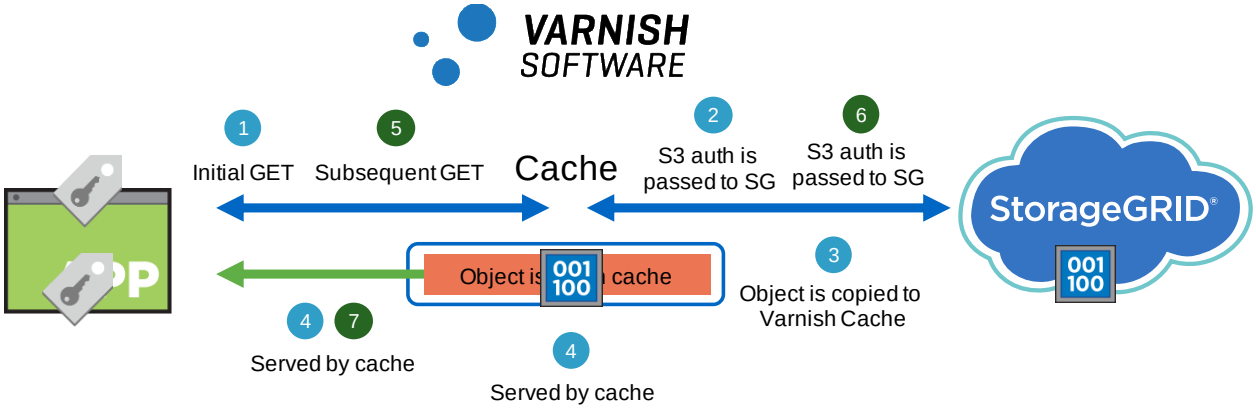
LIST OF FIGURES

1 Introduction

Varnish Cache is a web application accelerator. Placing Varnish Cache in front of an application such as NetApp® StorageGRID® dramatically improves the performance of objects that are repeatedly fetched.

The initial fetch of an object from StorageGRID includes a small amount of latency that is inherent to object store architecture. With Varnish Cache in front of StorageGRID, all subsequent fetches of an object are served from Varnish Cache with low latency and high throughput, as shown in Figure 1.

Figure 1) Varnish and StorageGRID architecture.

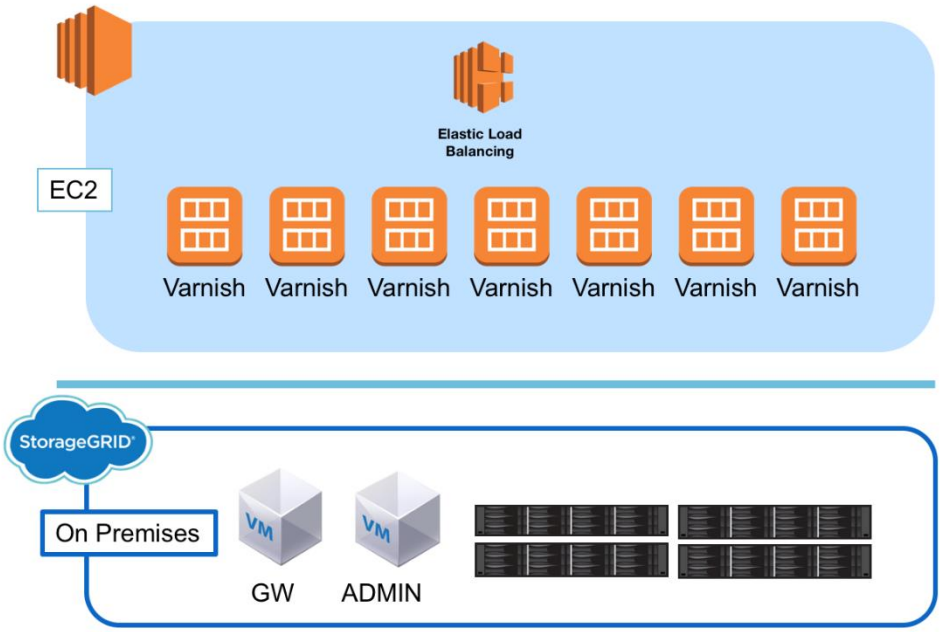


Varnish Cache can also be deployed as part of a content delivery network. By deploying Varnish Cache instances in multiple cloud providers, you can rapidly deploy and decommission as many service endpoints as needed.

By moving the GET performance workload to Varnish Cache, the StorageGRID implementation is simplified and sized to support capacity and ingest. Varnish instances can be rapidly deployed to meet bursty workloads and can then be decommissioned when they are no longer needed. Figure 2 shows Varnish deployed at a cloud provider while StorageGRID is deployed on the premises.

Varnish instances can be stood up in multiple data centers or cloud service providers to deliver massive scale and high throughput.

Figure 2) Varnish and StorageGRID deployment example.



2 Varnish Lab Configuration

For a proof of concept, we deployed a single Varnish Enterprise instance (commercial distribution) on a bare-metal server:

- OS: Centos 7.6
- 2x 8-Core Intel Xeon CPU E5-2640 v2 @ 2.00GHz
- 192GB memory
- 4TB disk
- Network: 1 Gigabit Ethernet (1GbE)
- Varnish Enterprise 6.0.4r1 (commercial distribution)

The StorageGRID configuration included:

- StorageGRID 11.2.0.2
- 6 virtual machine storage nodes, 4 StorageGRID appliances SG5612, and 4 bare-metal storage nodes
- Load balancer: Gateway node
- Load generator and testing: S3tester 2.1.0

2.1 Varnish Installation

This solution was built and tested with Varnish Enterprise, which is optimized for production usage and offers additional features that are relevant for this solution, like the Varnish Massive Storage Engine (MSE) and SSL. MSE is optimized for caching by using disk and memory.

Note: The Varnish Control Language (VCL) requires at least Varnish Enterprise 6.0.4r1, because some required bug fixes and features are available only in that release and later.

Follow the installation steps provided by Varnish to install the software.

2.2 Varnish Configuration

After you have installed Varnish, overwrite the default configuration file.

Provide StorageGRID Custom VCL

To provide the StorageGRID custom VCL, complete the following steps:

1. Replace `/etc/varnish/default.vcl` with the StorageGRID VCL file. Use the `default.vcl` file on GitHub as a basis.
2. Customize the `default.vcl` file to point to your grid:
   a. Enter the DNS name of your load balancer or Gateway node.
   b. Enter the port number. 8082 is the default port for API Gateway Node HTTPS.

Configure Massive Storage Engine

For best performance, NetApp recommends that you use memory and SSD or NVMe drives for caching. To configure the MSE, complete the following steps:

1. Follow the steps provided by Varnish to configure MSE. An example `mse.conf` file is provided on GitHub.
2. Copy the `mse.conf` file to `/var/lib/mse/mse.conf` and customize it for your environment.

   Note: Configure the `database_size` parameter to account for 100 bytes per object cached. Store the database directory on low-latency storage; otherwise, it can negatively impact small object throughput.
3. Initialize your MSE configuration by using `mkfs.mse -f -c /var/lib/mse/mse.conf`.

Configure Hitch to Use Your SSL Certificate

To configure Hitch to use your SSL certificate, complete the following steps:

1. Follow the steps provided by Varnish for setting up Client SSL/TLS termination.
2. Enter the SSL certificate that matches the DNS name of your StorageGRID S3 endpoint.
3. Copy your SSL certificate in PEM format to `/etc/hitch`.
4. Modify `/etc/hitch/hitch.conf`.
5. Set the front end to the desired port.

```
frontend = {
    host = "*"
    port = "443"
```

6. Hitch requires a single file that contains the concatenation of private key, certificate, and CA certificates. From the certificate and private key used in StorageGRID for the Object Storage endpoint, you can concatenate them with the following command:

```
cat example.key example.crt ca.crt > example.pem
```

7. Either copy the file to the default location, which is `/etc/hitch/testcert.pem`, or change the default location in `/etc/hitch/hitch.conf` to match the location of the PEM file.

```
pem-file = "/etc/hitch/your_ssl_cert.pem"
```

Restart Services and Test

To restart services and test, complete the following steps:

1. Configure Varnish to listen on the port of your choice. Hitch requires Varnish to listen on port 8443 for the PROXY protocol, which can be achieved with the parameter `-a :8443,PROXY`. If HTTP access should be allowed, also use the parameter `-a :80`.

   Make sure that the following settings are included:

```
vsl_mask = +Hash
http_gzip_support = off.
```

   The first setting adds the hash to the logging to simplify debugging. The second disables gzip, which breaks S3 authentication if enabled.

   Example of Varnish parameters:

```
/usr/sbin/varnishd -F -a :80 -a :8443,PROXY -s mse,/var/lib/mse/mse.conf -p vsl_mask=+Hash -p
http_gzip_support=off -f /etc/varnish/default.vcl
```

2. Restart the Varnish service.

After you restart Varnish, you can fetch an object from StorageGRID through Varnish by using the AWS CLI or a similar client.

3 Performance Testing and Results

Functional tests were successfully performed with CloudBerry Drive (3.0.1.5), S3 Browser (8-1-5), and s3fs (V1.85) for listing buckets and changing versioning, as well as uploading, downloading, and deleting objects.

Using s3tester as our load generator and testing tool, we ran various tests of Varnish. At a lower thread count, the performance benefits were more dramatic. Tests were conducted at 32, 64, 128, and 256 concurrent threads. Fine tuning yielded better results from both native StorageGRID and Varnish.

Note: Varnish provides extremely high performance. For maximum throughput, make sure that you have a very high request concurrency.

In each test, 100% of requests were served out of cache. To serve objects without additional fetches from StorageGRID, you should carefully consider Time to Live and cache size. The results shown in Table 1 and Table 2 compare native and cached access for full response latency and throughput.

Table 1) Latency.

Object Size	Concurrency	Native	Cached	Reduced Latency
4K	256	213.6ms	26.4ms	88%
32KB	256	286.3ms	26.9ms	91%
1MB	128	1,726.7ms	24.5ms	99%
10MB	64	11,143.0ms	64.9ms	99%
100MB	64	55,756.4ms	752.9ms	99%
1GB	32	286,042.8ms	3,493.5ms	99%

Table 2) Throughput.

Object Size	Concurrency	Native	Cached	Throughput Increase
4KB	256	4.6MBps 1175.1Obj/s	35.3MBps 9024.8Obj/s	767%

Object Size	Concurrency	Native	Cached	Throughput Increase
32KB	256	37.2 MBps 1189.3 Obj/s	269.5MBps 8624.9Obj/s	724%
1MB	128	111.2MBps 111.2Obj/s	4878.8MBps 4878.8Obj/s	4,387%
10MB	64	111.6MBps 11.2Obj/s	6053.5MBps 605.4Obj/s	5,424%
100MB	64	111.7MBps 1.1Obj/s	7887.1MBps 78.9Obj/s	7,061%
1GB	32	111.7MBps 0.1Obj/s	7819.1MBps 7.8Obj/s	7,000%

3.1 Security Considerations

Varnish allows customization by using Varnish Control Language. StorageGRID engineering provides a VCL file to allow S3 authorization. This customization is required to serve objects that require authentication by using StorageGRID. It enables VCLs transparent access to the S3 endpoint. Every request is authenticated and authorized as if the client would directly connect to the S3 endpoint.

4 Application Design Considerations

The provided StorageGRID VCL customization is intended to enable read caching of objects. Other operations, such as PUT, HEAD, and so on, are not optimized.

If you are considering using Varnish Cache for StorageGRID, plan for a development and test effort.

5 Conclusion

Varnish Cache combined with NetApp StorageGRID storage software can result in significant performance and lower latency for S3 read workloads.

Where to Find Additional Information

To learn more about the information that is described in this document, review the following documents and/or websites:

- StorageGRID 11.2 Documentation Center
  http://docs.netapp.com/sgws-112/index.jsp
- StorageGRID Documentation Resources page
  https://www.netapp.com/us/documentation/storagegrid.aspx
- Getting Started with Varnish Cache: The Varnish Book
  https://info.varnish-software.com/the-varnish-book

Version History

Version	Date	Document Version History
Version 1.0	March 2018	Initial release

Version	Date	Document Version History
Version 2.0	August 2019	Updated VCL to support subsequent requests with authentication

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

Copyright Information

Trademark Information

TR-4796-0919