Dell EMC ECS: GeoDrive Best Practices

Abstract
This paper provides an overview, architecture, performance, and best practices information for Dell EMC™ GeoDrive. GeoDrive is a free application that enables access to Dell EMC ECS™ and Atmos™ storage from Microsoft® Windows® desktops and servers.

June 2019
Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>December 2018</td>
<td>Initial release</td>
</tr>
<tr>
<td>June 2019</td>
<td>Removed references to Dedicated Cloud</td>
</tr>
</tbody>
</table>

Acknowledgements

This paper was produced by the following members of the Technical Marketing Engineering and Solution Architects team in the Unstructured Data Storage division of Dell EMC:

Author: Chris Kraft

Please send your feedback to unstructured.tme.sa@emc.com.

The information in this publication is provided “as is.” Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any software described in this publication requires an applicable software license.

Copyright © 2018-2019 Dell Inc. or its subsidiaries. All Rights Reserved. Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners. [7/1/2019] [Best Practices] [H17494.1]
Table of contents

Revisions ................................................................................................................................. 2
Acknowledgements .................................................................................................................. 2
Table of contents ..................................................................................................................... 3
Executive summary .................................................................................................................. 5
Audience ..................................................................................................................................... 5
Scope ......................................................................................................................................... 5
1 Introduction ........................................................................................................................... 6
2 Value of GeoDrive .................................................................................................................. 7
   2.1 Use cases ......................................................................................................................... 7
   2.2 High-level features .......................................................................................................... 7
   2.3 Overview ......................................................................................................................... 8
   2.4 Cache Path/Virtual Drive FAQ ...................................................................................... 9
3 Architecture ............................................................................................................................ 10
   3.1 Components .................................................................................................................... 10
   3.2 Data upload ..................................................................................................................... 11
      3.2.1 Asynchronous upload (default) .............................................................................. 11
      3.2.2 Synchronous upload .............................................................................................. 12
   3.3 Data restore ..................................................................................................................... 12
   3.4 Multi-site sync ................................................................................................................. 13
4 Performance study ................................................................................................................. 14
   4.1 Testing strategy .............................................................................................................. 14
   4.2 Tests, datasets, and metrics ........................................................................................... 15
   4.3 Upload tests ..................................................................................................................... 15
   4.4 Restore (download) tests ............................................................................................... 16
   4.5 Mixed workload testing ................................................................................................ 16
   4.6 Performance options ...................................................................................................... 17
   4.7 Test environment ............................................................................................................ 17
5 Performance results ............................................................................................................... 19
   5.1 Uploads ............................................................................................................................. 19
   5.2 Restores ............................................................................................................................ 20
   5.3 Mixed workloads ............................................................................................................. 21
      5.3.1 50/50 workload ........................................................................................................ 21
      5.3.2 20/80 workload ........................................................................................................ 22
      5.3.3 80/20 workload ........................................................................................................ 22
Executive summary

High demand for object storage is being driven by the adoption of modernized workloads and the need for multi-petabyte, archival-type storage. Dell EMC™ GeoDrive is a lightweight application which allows upload and download of files between Microsoft® Windows® systems and Dell EMC ECS™ and Atmos™ cloud storage platforms. GeoDrive creates a Windows virtual drive to object store and transfers data from Windows using the S3 REST API. It is designed as an easy access to data in the cloud by allowing Windows applications to interface with an object storage server.

ECS is a software-defined, cloud-scale, object storage platform that combines the cost advantages of commodity infrastructure with the reliability, availability, and serviceability of traditional arrays. ECS storage combined with GeoDrive provides applications and users efficient access to content in the cloud from a Windows platform.

Audience

This paper is intended for Dell EMC field personnel and customers who are interested in understanding the value and architecture of GeoDrive. It also provides information relating to the performance, tuning options and best practices for using the application with ECS.

Scope

While GeoDrive works with Atmos, this document focuses primarily on using GeoDrive to access ECS and provides an overview as well as information on the architecture, performance, and best practices.

This paper does not cover installation, administration, and upgrade procedures. The performance results described in this document do not cover scenarios related to geo-replication and enabling other features such as Access During Outage and Data at Rest Encryption. The performance results in this paper are from testing done using a previous version of the product, CIFS-ECS v1.2, and ECS v3.1, as described below in Table 2. Similar performance can be expected using the most recent versions of GeoDrive and ECS.

Updates to this document are done periodically and may coincide with major releases or new features and functionality.
1 Introduction

Beginning with GeoDrive 2.0, ECS and Atmos can be accessed using a single application instance of GeoDrive. Prior to GeoDrive 2.0 separate applications existed, one for each Dell EMC object storage product type. CIFS-ECS was available for use with ECS, and GeoDrive (pre-version 2.0) was available for use with Atmos. The two applications were practically identical, and this latest version simplifies deployment for all users by combining the functionality in to one branch of code.

The current features of both products are supported in the combined GeoDrive application. Both CIFS-ECS and GeoDrive can be seamlessly upgraded to GeoDrive 2.0. Supporting connections between both Atmos and ECS, and Windows, GeoDrive 2.0 simplifies migration for organizations currently using GeoDrive with Atmos that want to migrate to ECS. All application configuration is preserved during the upgrade process. Configuration files can now be merged in to the existing configuration.

This paper provides an overview of GeoDrive and a high-level architecture. It also provides information on performance studies conducted, tuning options available, and best practices for the application.
2 Value of GeoDrive

GeoDrive provides fast and transparent access to ECS. It does not require complicated shares, mount points, or API development. Since GeoDrive presents an ECS bucket as a local drive in Windows, users and applications can store content such as photos, music, videos, documents and email attachments in the cloud using existing applications written for Windows, with no additional IT or infrastructure cost. The software is provided with no charge for all licensed ECS and Atmos customers. It is simple to install, configure and use. Files are maintained in Windows in a local disk cache and pushed to the cloud either asynchronously or synchronously by GeoDrive.

2.1 Use cases

GeoDrive was designed as an easy on-ramp to the cloud. A few of the primary use cases are:

- Enterprises that would extend efficiencies of ECS for native SMB access. In this scenario, GeoDrive is installed in a centralized server and IT would be responsible for mapping the virtual drive as a network share to clients and users.
- Application vendors (i.e. life science, healthcare industries, etc.) who would like to use GeoDrive as the target for their Windows SMB-based application, eliminating the need for additional development, and have ECS serve as the long-term end destination.
- Global Content Repository/Archive and Long-Term Retention.

GeoDrive is not a good fit for the following use cases:

- High performance workloads/Dynamic file sharing.
- Sync and Share.
- Desktop solutions.

2.2 High-level features

Key high-level features of GeoDrive include:

- User friendly, customer installable, software application with simple GUI for management, monitoring and install wizard. Remote CLI installation is also available.
- Cloud object storage available to a local drive in Window Operating Systems.
- Supports Windows Server Failover Clustering.
- Built-in load balancer with round-robin algorithm.
- Includes options for synchronous upload and multi-site sync.
- Includes advanced options for data throttling.
- Built-in disaster recovery - data is accessible from another system if the local system fails.
- Integration with Windows Performance Monitor, Event Viewer and Explorer.
- Event Reporting and Messaging.
- Uses ECS S3 REST API (supports LAN and WAN).
- Maintains Windows file properties (attributes and security descriptor).
- Allows for custom user metadata and file exclusion rules.
- S3 ACL Support - translates Windows ACL to S3 ACL on uploads and permission changes.
- S3 Versioning Support - allows users to recover files deleted or overwritten using Windows Explorer.
2.3 Overview

GeoDrive enables Windows administrators to create a relationship between Windows and cloud storage systems. GeoDrive associates Windows local NTFS or ReFS storage with ECS storage. GeoDrive does not create a new file system or volume in Windows. It piggybacks on an existing NTFS or ReFS volume and uses a portion of it as a cache area for data sent to, or retrieved from, ECS. Multiple associations between Windows and ECS can exist.

The basic construct of GeoDrive is a cloud drive. A cloud drive is the term GeoDrive uses to define a virtual drive created by the application. Cloud drives are the link between storage on Windows and ECS. In Windows, a single volume can be common to multiple cloud drives, or a cloud drive can use a volume independently of other cloud drives. In ECS, a bucket is configured for each cloud drive. More than one cloud drive can use the same underlying Windows storage volume. Similarly, more than one cloud drive can reference the same ECS bucket, however, if multiple cloud drives refer to the same bucket there is a real possibility that the data won’t always be in sync, or a possibility that one user’s data can overwrite another user’s data. For example, if two users create a file with the same path at about the same time in two cloud drives with a common bucket, one of the files will be lost since one will be uploaded first, and then the other will be uploaded and overwrite the first. It is important when sharing buckets to have strong conventions as to which files can be modified by which server.

A cloud drive is generally created in GeoDrive by first selecting a Windows drive letter not currently in use. Having a drive letter associated with the data is optional however. Using drive letters for configuration is available as a convenience for users to be able to find the data easier. A cloud drive associates a directory with a bucket. If no drive letter is configured during cloud drive creation, files are accessed through the configured local Windows cache directory. All file data existing locally in Windows is considered cached by GeoDrive.

A common term used by GeoDrive is ‘stub.’ A stub file is a shortcut file which contains only the information required by GeoDrive to determine where to retrieve file data from the cloud. A file can be stubbed only after it has been uploaded to ECS. If, after being uploaded, the file is not accessed for a period (the “Access Wait Time”), the file will be stubbed. A stub file can be identified by the “X” overlay on the file’s icon and because the OFFLINE attribute is set.

Stubbing increases the amount of free storage available on the Windows local disk while maintaining user access to the data. Upon a data access request to a stubbed file, GeoDrive restores the data from ECS. After retrieval from ECS, the data is stubbed again per the configured time interval.

For each cloud drive a local storage option is selected which determines how files are stored in the cloud drive. The two options are:

- **Mirror mode** - Mirror all content to the cloud. Maintains a local and cloud copy of all data.
  - Uses more local disk space than the stubbing option. Requires enough local disk space to support the entire set of files to be stored in the cloud.
  - Provides consistent speeds during data access.
  - Data is always accessible

- **Push-to-cloud mode** - Replace recently un-accessed files with a stub file.
  - Uses less local disk space than the mirror option.
  - Access to files stubbed to the cloud may be delayed.
  - Network connectivity to the cloud storage system is required.
Optional two additional options for cloud drive configuration are available. They are:

- **Metadata** - The metadata configuration defines metadata that is created for an object when it gets uploaded to the cloud. Saving metadata tags with the objects can allow the data to be searchable based on the tags.
- **Excludes** - Excludes allows administrators to specify any files that GeoDrive should not store in the cloud. This is done by specifying individual files, file patterns, folders, folder patterns, or as a file path.

**Note:** Windows administrators can share GeoDrive data with users just like they do with any other data. It is important to understand that if a drive letter is assigned to a cloud drive, administrators should not use the drive letter configured with the cloud drive when creating the Windows share. A Windows share of GeoDrive data should be created using its filesystem path, not the configured GeoDrive letter.

### 2.4 Cache Path/Virtual Drive FAQ

A common misconception is that the virtual drive created from a cloud drive is a new filesystem that reflects the contents of the ECS bucket. In other words, if a drive I: is created, and the properties of the drive I: are viewed, they will show the amount of space used in ECS by the bucket, and the amount of free space on ECS. It does not, and there is no relationship between the size of a windows volume and the amount of storage at ECS.

The following points clarity this relationship:

- GeoDrive does not create a new file system or volume. It piggybacks on an existing NTFS or ReFS volume and uses a portion of it as a cache area for data pulled down from ECS.
- Data local on the windows system that has not been referenced for a period is stubbed. This will **increase** the amount of free storage available on the local disk while keeping the amount of data on ECS constant.
- Files are pulled down to the local disk as they are referenced. All files won’t be local or stubbed unless all folders have been opened or referenced, or if you run **Recall All Directories** from the root folder in Explorer.
- Having a drive letter associated with the data is optional. It is there as a convenience for users to be able to find the data easier. It is not really a new drive. It is a junction point into where the cache directory resides. If the properties on the I: drive are viewed, they will be the same as the C: drive, or whatever drive that is used for the cache.
- When creating a cloud drive, the cache directory will default to under the data directory. For applications with a lot of data, it is highly recommended that you don’t accept that default and put it on its own volume. Click on Advanced when creating the cloud drive. The cache path can be modified only when initially creating the cloud drive.
- Be sure to read the appendix in the user guide on sizing the cache directory. It is possible for the cache volume to run out of disk space while there is plenty of space on ECS.
3 Architecture

GeoDrive is installable on Windows Operating systems. Installation requires a minimum 1 GB of disk space for the binaries and logs; and 8GB is the recommended minimum for memory. It uses local storage available to the Windows system as a temporary cache to store and restore data from ECS. The local drive cache resides in volumes formatted as NTFS or ReFS (Windows 2012 R2 or later) and acts as GeoDrive virtual drive. GeoDrive uses S3 REST API to write and read data on ECS. Applications can create, modify and read files normally using GeoDrive cloud drive, and files are automatically uploaded asynchronously or synchronously to ECS. Files not accessed locally in Windows for a specified amount of time are replaced with a stub file in order not to consume local disk space. Stubbed files can be cached again locally to disk on Windows when accessed or sent straight to the requesting client without persisting the data again to disk.

3.1 Components

The components of the GeoDrive application as shown in Figure 1 below include the following:

- **Console GUI/CLI** – provides application and cloud drive configuration, the status of connections to cloud storage, the activity of data (upload/download rate, etc.), and error logging. A command-line interface (CLI) is also available.
- **GeoDrive Filter Driver** – monitors disk I/O in the Windows kernel. The driver is tightly linked with GeoDrive service to perform functions like monitor file opens and closes, stubbed file restores and file deletes, renames, modification of attributes, directory and file syncs, and handling of “file not found” scenarios.
- **GeoDrive Service** – main service responsible for handling file upload to ECS, performing stub operations when a file is not accessed for a specified amount of time, and managing the internal thread pools. The service also works with the driver to suspend applications while stubbed files are restored, downloads directories from ECS, updates files if there are more recent versions found on ECS and downloads a file if an application is opening a file that does not exist on the local disk cache.

![Figure 1](image_url)  
**Figure 1** GeoDrive core components are the console, driver, and service
3.2 Data upload

When a file is created or updated in a cloud drive it is written in the local disk cache and uploaded asynchronously to ECS by default. Files are asynchronously uploaded to ECS based on a user-defined period after a file is last modified. For large files, a multi-part upload occurs such that the file is divided into parts and then is sent to ECS in parallel threads. This has the following benefits:

- Better utilization of the available bandwidth
- Faster uploads
- If a network error occurs and a retry is necessary, it will not retry the entire file but can retry only the affected part

GeoDrive also offers synchronous uploads to ECS.

3.2.1 Asynchronous upload (default)

By default, after a file has been created or modified, the file is created or updated on local disk cache, and then uploaded by GeoDrive to ECS asynchronously. The client application will get an acknowledgment of the completed file operation request as soon as the data has been written to the local disk cache. Once a file in local cache has not been modified for a specified period, it is added to the queue to be uploaded as illustrated in Figure 2 below. The data is uploaded to ECS using the S3 API. The upload queue is processed as a background process.

![Client file operation acknowledgment during asynchronous upload](image)

Figure 2 Client file operation acknowledgment during asynchronous upload
3.2.2 Synchronous upload

Enabling synchronous upload eliminates the normal delay in uploading files to ECS. A file is immediately uploaded during the last close of the file. The benefit of using synchronous uploads is a reduction of the risk of data loss due to an unrecoverable Windows server or local disk cache failure. The downside is the application will be held up during the last close of the file for the duration of the file upload. GeoDrive maintains a counter on the number of times a file is opened. When all applications accessing a file have done a close of the file, the GeoDrive service synchronously uploads the file to ECS. With synchronous uploads, clients will get file operation acknowledgment after the upload to ECS has completed. This is shown in Figure 3 below.

![Client file operation acknowledgment during synchronous upload](image)

3.3 Data restore

When a stubbed file is opened for read, it is restored from ECS storage as pictured in Figure 4. To reduce space on local disk cache, files are stubbed when a file is not accessed for a certain user-defined period, or when a certain space threshold on local disk cache has been reached. For large files, restores are done in parts and in parallel and then restored for performance.

![Restoring a stubbed file on read](image)
3.4 Multi-site sync

Multi-site sync allows multiple GeoDrive servers to reference the same ECS data. Using this feature minimizes the amount of time that peer GeoDrive servers spend out-of-sync. By enabling multi-site sync, peer GeoDrive servers are notified when a file create or modify operation is completed by one of its peer servers. This is shown in Figure 5 below. When a file is created on one GeoDrive server and after it is uploaded to ECS, peer GeoDrive servers are notified and a directory sync operation occurs on peer servers to recognize the file creation. In this scenario, the file will appear as a stub on peer servers and file will be restored to local disk cache when the file is opened for access. When an existing file is modified or updated, the GeoDrive server that initiated the modification notifies its peers after the upload to ECS has completed, and the peer servers initiate a restore of the file in to its local disk cache. The file upload to ECS will be asynchronous unless the synchronous upload option is enabled.

Figure 5  Multi-site sync peer server communication
4  Performance study

Studies were previously conducted to understand the performance of GeoDrive, uncover the limitations, and measure its full capabilities. The investigation in to the advanced options such as number of upload and restore threads, and multipart threads to determine if these parameters can be tuned to improve the overall performance of large file transfers was also done. Analysis which are not in the scope of this study include: geo-replication setup, since replication is done within ECS and performance of GeoDrive is affected in scenario where one site goes down and data would need to be retrieved from a remote site; and enabling other ECS features such as Access During Outage and Data at Rest Encryption.

**Note:** Windows systems which have been patched for the Meltdown and Spectre vulnerabilities may result in decreased performance compared to non-patched systems.

4.1  Testing strategy

The test strategy consisted of a series of different tests to exercise GeoDrive performance when uploading and downloading files from an ECS appliance. The challenge in using existing performance testing tools such as iozone, FSCT, or SPECSsfs benchmarks to test GeoDrive is that data is uploaded to ECS asynchronously or at a delayed specified time (minimum 1 second) after the file is closed and written to local disk. Using these tools will only measure the performance of reads and write to local disk and not measure the performance of GeoDrive. Thus, test scripts were written to create the datasets used and normal windows applications such as drag and drop items to folder and features within GeoDrive tool to upload and restore files from ECS. Filesystem logging was enabled to track all GeoDrive operations as shown in Figure 6. These logs were utilized to determine the time it took to fully upload and restore all files.

![Logging enabled for GeoDrive operations to determine time to upload and restore files](image-url)
4.2 Tests, datasets, and metrics

A python script was written to generate the dataset, “n” number of files of the following sizes: 10KB, 100KB, 1MB, 10MB, 100MB, 200MB and 1GB (for large transfer testing) to observe the performance of GeoDrive for certain file sizes. For the files to not be easily compressed, the contents of the files were randomized by using the python random function. Workloads tested include 100% uploads, 100% restores and mixed workload of 50% restores and 50% uploads, 80% restores and 20% uploads and 20% restores and 80% uploads. Each of the tests was repeated a minimum of three times and an average was calculated. Results are represented in MB/Seconds for throughput and Files/Seconds for transaction rate. Table 1 below provides the number of files for each of the sizes utilized for the testing. For the 50%-50% workload testing, the total number of files utilized was doubled (i.e. for the 10 KB data size, 30,000 files were uploaded, and 30,000 files were restored, at approximately the same time.)

<table>
<thead>
<tr>
<th>Data size</th>
<th>Number of files</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 KB</td>
<td>30,000</td>
</tr>
<tr>
<td>100 KB</td>
<td>20,000</td>
</tr>
<tr>
<td>1 MB</td>
<td>10,000</td>
</tr>
<tr>
<td>10 MB</td>
<td>5,000</td>
</tr>
<tr>
<td>100 MB</td>
<td>1,000</td>
</tr>
<tr>
<td>200 MB</td>
<td>1,000</td>
</tr>
<tr>
<td>1 GB</td>
<td>1,000</td>
</tr>
</tbody>
</table>

4.3 Upload tests

In the upload scenario, the GeoDrive UI was used to manually pause, un-pause and force upload using the buttons shown in Figure 7. These functions where used to drain the queue as well as fill the queue with files to upload. Windows drag and drop functionality copied files to the GeoDrive local disk cache. Once the files were completely copied to local disk cache, then GeoDrive was manually un-paused and a force upload was conducted. The start, end and duration times captured from the logs were used to calculate the throughput and transaction rates.

![GeoDrive UI Force and Pause Upload buttons](Image)
4.4 Restore (download) tests

In the restore scenario, the GeoDrive right-click context menu on the cache directory path in Windows Explorer was used to Force Stub and Bulk File Restore as shown in Figure 8 below. For the restores, all files in the local disk cache were forcibly stubbed. As soon as all files were marked as stubbed in the directory, a “Bulk File Restore” was initiated in the directory within the local disk cache. The logs captured actual restore times to GeoDrive local disk cache. The start, end, and duration times in the logs allowed for the calculation of the overall time to complete restore.

![Right-click context menu available to a cloud drive cache directory in Windows Explorer](image)

Figure 8  Right-click context menu available to a cloud drive cache directory in Windows Explorer

4.5 Mixed workload testing

Two directories were created where one directory will contain a percentage of files written and the other directory will contain a percentage of files restored for the mix-workload testing. The same steps defined above for upload and restore processed the mixed workloads. As mentioned, GeoDrive was paused for the upload part, and files were forcibly stubbed for the restore part, before the tests were initiated. Through the GeoDrive command line interface (CLI) the virtual drive was un-paused and the upload was forced. An example of the command line to initiate the tests:

```
cloud_cli resume n::cifs-ecs_cli force n:
```

Once the transfer status as shown in Figure 9 indicated that the upload has started, a “Bulk File Restore” was initiated on directory to restore the stubbed files. To calculate the throughput and transaction rates, the logs were again employed to capture the completion times.

![Transfer status window grabbed during testing](image)

Figure 9  Transfer status window grabbed during testing
4.6 Performance options

The performance parameters were set to defaults except for the upload and restore threads which were set to 64. The thread settings were determined to provide the best peak performance for the workloads conducted in this study.

Other advanced options such as the number of upload and restores threads and multipart threads are available. These options were modified one at a time to validate the effect they have on overall performance for uploads and downloads. For upload and restore threads testing, 30,000 files of 10KB data size was utilized and for the multipart threads a single 1GB file size was used as the datasets. The effects of these performance parameters are discussed in the Tuning section of this paper.

4.7 Test environment

Table 2 below describes the specification of the hardware and software utilized for this performance study and Figure 10 below illustrates the hardware and network connectivity setup.

Table 1 Hardware and software used for testing

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Specifications</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>CPU: 2x Intel® Xeon® E5-2650 v3 Processor (2.3 GHz, 10 Cores, 10x 256KB L2 and 25MB L3 Cache), Memory: 256 GB Disks: 12 600GB 10K RPM SAS drives 4 disks: Virtual single drive: RAID 0, NTFS for GeoDrive local cache drive 1 disk: RAID 0 where all datasets were stored</td>
<td>Windows 2012 Server R2 Standard CIFS-ECS version 1.2 (x64) Python 3.5.1</td>
</tr>
<tr>
<td>ECS Appliance</td>
<td>U2000 Gen 2 Appliance 8 nodes each with dual, six-core 2.4 GHz Intel Xeon® (E5-2620) processors Memory: 64 GB Disks: Each node has direct access to 30 8TB 7200RPM SAS drives</td>
<td>ECS v3.1 Single Site VDC with Single Bucket Pre-populated with a minimum of 1.72 Million objects</td>
</tr>
<tr>
<td>Network</td>
<td>10GbE</td>
<td></td>
</tr>
</tbody>
</table>
The ECS Appliance was pre-populated with a minimum of about 1.72 million objects. One GeoDrive drive was mapped to a single site, a single namespace and a single bucket. The GeoDrive drive used as a local disk cache consisted of a virtual drive of RAID 0 with four 600GB SAS drives (10K RPM) and formatted with NTFS. The IP addresses of each of the ECS nodes were specified as the hosts and GeoDrive was responsible for doing the load balancing across the nodes. GeoDrive was set up to connect to ECS via HTTP. One 10GBe network port uplink to the 10GBe switch was used on the GeoDrive server. ECS had two 10GBe uplinks to the switch one per ECS top of rack 10GBe switch.

Figure 10  10GbE test network topology
5 Performance results

The performance results and analysis discussed in this section is based on a single application instance using a Gen2 ECS U2000 appliance. Results for each of the workloads run are described in detail below.

5.1 Uploads

For uploads, a certain number of files of certain sizes were read from the local disk cache and sent to ECS appliance. Figure 11 shows the results for 100% upload in Files/second and MB/second. Peak throughput of 586.50 MB/seconds was observed for files of 10M in size. The peak transaction rate performance achieved was 346.29 Files/second for files of size 100K.

![Figure 11 100% uploads](chart.png)
5.2 Restores

For the restores of files, it required a read request from ECS Appliance and a write to mapped local disk for GeoDrive. As previously mentioned in the test environment description, a virtual disk drive of RAID 0, four 600GB 10K RPM SAS disks were utilized for this study. Using different types of drives such as SSD may further improve the performance of GeoDrive restores. Figure 12 shows the performance obtained for restores. The peak transaction rate performance was 697.34 Files/second for files of size 10K and for files of size 100M the peak throughput was 1120.26 MB/second.

![Performance Graph](image)

**Figure 12** 100% restores
5.3 Mixed workloads

In most use cases, workloads are usually mixed where there is a combination of uploads and restores occurring. Mixed workload studies of the following percentages: 50-50, 80-20 and 20-80 were conducted and the results presented in this section provide a reference of GeoDrive performance for these types of workloads.

5.3.1 50/50 workload

The performance for 50/50 workload is shown in Figure 13. The maximum transaction rate achieved was 584.99 Files/second for file sizes of 100K and the maximum throughput was 558.27 MB/second for files of 10M in size.

![Figure 13 50% uploads, 50% restores](image-url)
5.3.2 20/80 workload

Figure 14 illustrates the performance of 20% restores and 80% uploads. As can be seen from the graph, the peak Files/second achieved was 378.95 for file sizes of 10K and peak throughput was 430.33 MB/second for 100M sized files.

5.3.3 80/20 workload

For the mixed workload of 80% restores and 20% upload, the peak transaction rate attained was 733.66 Files/second for 10K sized files and peak throughput was 1301.75 MB/second for files 100M in size. The results were affected by the large percentage of restores in this workload compared to the percentage of uploads done. Figure 15 shows the results from this test.
5.4 Observed load on the server
GeoDrive was designed to be a lightweight application and thus the load on the server was generally low. The overall CPU usage ranged from 4%-37% depending on workload, memory utilization ranged from 2%-23% and network utilization ranged from 0% to 92%. Highest network utilization was seen for restores of large files and highest CPU usage was for restores. Depending on the specifications of your server these numbers can vary. There are options in GeoDrive to minimize the server resource utilization such as limiting the number of threads for uploading and downloading of files from ECS and or data throttling to allow for the server to be employed for other purposes.

5.5 Observed load on the ECS appliance
A single instance of GeoDrive had minimal impact on the load of the ECS Appliance. ECS can handle a higher transaction rate and throughput than a single GeoDrive instance can provide. This means more instances of GeoDrive can be deployed on other servers to further utilize and maximize the performance of ECS.

5.6 Analysis summary
GeoDrive single instance performance can be summarized as follows:

- Peak throughput was observed predominately for file sizes of 100M (large files) and peak transaction rates were observed for files sizes of 10K (small files).
- Performances of mixed workloads are lower than the 100% workload scenarios.
- Performance tends to plateau for files greater than 100M and less than 100K.
- Workloads having a high percentage of restores affect overall performance when compared to workloads where uploads are predominant.
- A single instance of GeoDrive had low to normal impact on overall load of server used for this testing.
- A single instance of GeoDrive has minimal impact on ECS Appliance such that more instances of GeoDrive can be deployed on other servers to utilize the full potential of ECS Appliance.

The graphs in Figures 16 and 17 show the transaction rates and throughput for each of the workloads respectively. This provides a better picture of how each of the workloads compare with each other and re-iterates some of the key points summarized above.
Performance results

Figure 16  Files/second per workload

Figure 17  MB/second per workload
6 Tuning options

GeoDrive provides several performance parameters that are configurable using the GUI. The parameters tested to observe the effect on overall performance of GeoDrive include number of upload and restore threads and number of multipart upload and multipart download of threads.

6.1 Threads

The number of threads GeoDrive spawns to handle requests is a user configurable option via the GUI. However, the maximum thread values available depends on the available resources (i.e. CPU, memory, etc.) of the system GeoDrive is installed in. Uploads and restores tests of 30,000 files of 10K size were conducted to evaluate the performance when the upload and restore threads were increased.

6.1.1 Upload threads

An improvement of 78% was seen when increasing the upload threads from 8 to 16 as illustrated in Figure 18. As mentioned, the tests to achieve this included uploading 30,000 files of 10K in size. If there are available resources on the server, increasing the upload threads to 64 can further improve performance as reported in the Performance Results section.

![Figure 18: Gain from increasing upload thread count](image-url)
6.1.2 Restore threads

Increasing the restore threads from 8 to 16 threads had an 11% improvement as pictured in Figure 19. For restores the write performance of the local disk cache can affect overall performance. Thus, increasing the threads for restores may cause high disk activity on local disk cache.

![Figure 19: Gain from increasing restore thread count](image)

6.2 Multipart threads

S3 is the protocol GeoDrive uses to send and receive data from the ECS storage platform. The S3 REST APIs provides multipart uploads which involve breaking up large files in to smaller parts and sending the parts to ECS in parallel. GeoDrive utilizes this attribute to improve GeoDrive upload performance. However, for restore, the S3 REST API does not offer this capability. GeoDrive simulates this behavior by initiating multiple read commands using the offset and length fields. Once all the parts of the file are received, they are combined in to the whole file and copied over to its destination. Large files greater than the multipart minimum threshold size benefits the most from this feature. Tests were run to determine the impact increasing the multipart threads for uploads and restores. For these tests, the minimum threshold and maximum part size were set to default values of 128MB. These values represent the minimum and maximum sizes the parts should be broken up in to. For large files, throughput (MB/s) performance was where the most improvement can be seen. In the following sections, results from the tests are discussed.
6.2.1 Multipart upload threads

For the multipart uploads, a single file of 1 GB size was uploaded to ECS. As can be seen from Figure 20, there was an 88% improvement when the value of multipart-upload was increased from 2 to 4 MPU threads.

![Figure 20](image)

Figure 20  Gain from increasing multipart upload threads (1 GB single file upload)

6.2.2 Multi-part restore threads

For the multipart restore threads, a single 1 GB file restore with increasing values of multipart restore threads was performed. As pictured in Figure 21 below, there is a 58% improvement when increasing the multipart restore threads from 2 to 4 MPU threads.

![Figure 21](image)

Figure 21  Gain from increasing multipart restore threads (1 GB single file restores)
7  Best practices
This section describes best practices for GeoDrive.

7.1 Windows backup
Drive letters can be associated with GeoDrive cloud drives to represent the local disk cache. When adding a GeoDrive drive, the registry system parameters will automatically exclude the drives mapped to an ECS bucket. If your backup application does not read the parameter specified in the registry (HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\BackupRestore\FilesNotToBackup) then it is advised to exclude the folders used by GeoDrive within your backup application.

Dell EMC recommends excluding files managed by GeoDrive from your Windows backups. The reasons are:

- The files are already backed up in the cloud.
- Backing up a stub file does not provide any value because:
  - It is a shortcut representation of an inactive file stored in the cloud.
  - In the event of a catastrophic failure, a new GeoDrive instance can pull stub info from ECS.
- The Windows backup operation may cause all stubbed files to be restored. Restoring stub files takes time depending on the number of stub files and their size which may delay the Windows backup operation.
- The interaction between the Windows backup operation, the restored files, and the cloud causes performance thrashing between your system and the cloud. This may also exhaust the system disk space and the network connection bandwidth.

7.2 One ECS site per host
GeoDrive has a built-in load balancer and when specifying the ECS hosts, it is recommended the ECS IP addresses or hostnames provided per host are all from one site. Specifying hostnames or IP addresses of more than one ECS site is not recommended.

7.3 Running executables
Running executables (.exe) inside the GeoDrive drive is not recommended. The executable or files being used by the executable may be stubbed out and may not function as expected. Pointing a database/SQL application to a GeoDrive drive is not supported either for the same reason.

7.4 Tuning
Understanding the workload utilizing the GeoDrive is key in determining the performance parameters to modify for tuning the application. The performance results presented in this paper can provide references on how best to set these parameters. It is best practice to start with defaults and adjust performance parameters one at a time only if performance degradation is observed. Caution is advised when increasing these values beyond the capabilities of the server hosting the GeoDrive. Performance degradation may occur in this situation.
7.5 Compression and encryption of GeoDrive files
Files compressed or encrypted in GeoDrive cannot be read by other cloud applications (S3Browser or Cloud Explorer). Do not enable GeoDrive compression or encryption if you intend to access files in the cloud through applications other than GeoDrive.

7.6 Avoid multiple concurrent writes to files
GeoDrive was not designed to be a sync and share application. With GeoDrive files are first cached locally in Windows before being sent to the cloud storage platform. Files are uploaded to ECS once the file has not been accessed for a period or only after a close of file. As a best practice, manage the GeoDrive environment such that multiple users are not accessing the same files at the same time from two or more different GeoDrive instances. This reduces the risk of users overwriting another user’s changes.

7.7 Isolate local disk cache for high-performance workloads
Windows local disk cache can act as a cache for several ECS buckets. If a workload requires higher performance, it is recommended to isolate cache for that workload to its own local disk volume, instead of sharing the space with other buckets. In addition, Dell EMC recommends using SSDs or faster disks for workloads requiring more performance. Always activate the journal feature (USN Journal) for each drive to aid in recovery in case of a crash, power loss or blue screen.

7.8 Sharing GeoDrive folders
When sharing out GeoDrive folders, always share out a folder from the cache path, not a virtual drive letter. This is because stopping GeoDrive may cause shares on cloud drives to be lost.

7.9 USN change journal
The USN Change Journal feature of NTFS should always be active. This feature aides GeoDrive by identifying files that have changed since the last successful checkpoint. Having USN Journal enabled could prevent a full recovery in the case of a power failure, blue screen or other failure. This is especially important for clustered systems. The USN journal can greatly shorten a GeoDrive recovery. In the case of a recovery, the Windows outage must be short enough to be covered by the events recorded in the journal file.

7.10 Do not manually disable the GeoDrive service
The GeoDrive service works together with the GeoDrive driver. The driver monitors file operations in the Windows kernel and sends file operations GeoDrive needs to handle to its service for processing. If the service is unavailable to the driver, the driver sends file operations to a spool file. If the service is stopped for a long period, the spool file can become excessively large and may fill up the local file system. The spool file exists in the GeoDrive Data directory.
### 7.11 Avoid a large logging messages queue size

Avoid a high log message queue size. A high number of log messages can slow down the user interface considerably. This is because the GeoDrive user interface loads messages from its logs at the start and during other events where it determines its message log needs updating.

As a best practice, the following are suggested:

- In the GeoDrive Options/Log tab, change the maximum size to no more than 10000.
- Send SEVERE and WARNING log messages to the Event Viewer. Do not send INFO log messages to the Event Viewer. Do send INFO log messages to the Notification Area in the UI if desired.
- Uncheck "Log Uploads" and "Log Restores." This is important because there are typically so many of this message type that they push out more important WARNING and other INFO messages.

If logging upload, restore and stub events is desired, the recommended configuration method is to double-click on the drive entry, click on the Logging tab, and enable logging. Verify that "GeoDrive Operations" to log uploads, restores and stubs is enabled. This allows for more efficient high-volume logging and allows flexibility in determining how much disk space to devote to logging.

Using the settings above provides administrators with the latest 10,000 SEVERE/WARNING/INFO messages in the UI. At times it is desired to look back further, the Event Viewer will contain only SEVERE and WARNING messages. These recommendations are shown in Figure 22 below.

![GeoDrive - Options](image)

Figure 22  GeoDrive global logging options

### 7.12 Sizing the cache directory

Be sure to read the appendix in the user guide on sizing the cache directory. It is possible for the cache volume to run out of disk space while there is plenty of space on ECS.
### 7.13 Avoid mixed-use buckets

Avoid allowing access to ECS buckets used by GeoDrive by non-GeoDrive applications or users. Updates to data stored in GeoDrive buckets by non-GeoDrive applications will lead to issues. GeoDrive tries to keep relevant Windows operations in sync with ECS by checking the corresponding buckets for changes that may have happened without its knowledge.

It does this with the following process:

- If a process opens a folder, GeoDrive checks the last time it synced the folder. If it has been longer than a configurable timeout, it syncs it. The requested open operation by the process is stalled while the folder is synced.
- If a process opens a file, GeoDrive does the same thing with the file: checking if the file on ECS is later than what is on the local filesystem.
- If a process opens a file that does not exist in the local filesystem, it checks ECS to see if it exists there. If it does, it is pulled down to the local filesystem and the file open request will then succeed.

The three functions above are gated by the “Update Wait Time” parameter. This means that if a process opens a non-existent file, it will check ECS the first time, but subsequent times it will just immediately fail the open.

Often GeoDrive performance issues can be due to operations as described above. If a folder contains 50,000 items, for example, a sync may take a minute or more to complete. Only allowing GeoDrive access to dedicated GeoDrive-only buckets, eliminates issues with Windows and ECS being out-of-sync.

### 7.14 Always use lowercase naming

GeoDrive administrators should always use lower-case characters when creating purpose-built-for-GeoDrive paths and naming schemes. This is important because ECS (and S3) are case dependent. Case dependence means that a file named `file.pdf` is a different file than one named `File.pdf`. The Windows filesystem is case independent. This means that Windows cannot distinguish between files that are identical in name except where case is concerned.

If a Windows user creates a file named `File.txt`, GeoDrive cannot upload it as is. This is because that file may already exist in ECS, but in a different case, such as `file.txt`. If GeoDrive didn't take character case into consideration, two versions of the file would exist in ECS, and one will be chosen at random, to be brought down to Windows when requested by the user.

Before GeoDrive uploads any file to ECS it first retrieves a listing of the folder in ECS to see if the file is there but in a different case. GeoDrive must also consider character case in folder names. GeoDrive works its way down the path until it finds a folder that exists, and then does folder listings on each folder to find the actual path. Significant optimization is built in to GeoDrive to minimize the time spent required to handle differences in character case, however, the impact can still be significant.

Generally, if the users of GeoDrive are Windows users, they may never know about the lower-case option, which is fine. GeoDrive does write files to ECS in lower case and stores the original name case in the metadata. When a file needs to be recreated on Windows, files will be restored using the original filename.
7.15 **Excluding functions for specific processes**

There is a facility in GeoDrive that allows administrators to select a process and either completely exclude it from all GeoDrive processing, or selectively exclude certain functions. The Processes tab in the GeoDrive options dialog is shown in Figure 23 below.

![Processes options dialog in GeoDrive](image)

**Figure 23** Processes global options in GeoDrive

Administrators can select a process, and check “Enable” at the bottom, and then check:

- Synchronize Folder
- Synchronize File
- File Not Found

From then on, whenever that process opens a folder or file, it will not do the sync/not found function.

You may ask, what if the process opening the file is on another box? In that case `<SYSTEM>` is selected, because that file operation comes over via the network redirector in the kernel.

If there is any question about whether the local filesystem is completely up-to-date, you can fix it by:

- If the process exclude is for `<SYSTEM>`, you can open that folder in question with Explorer. If there is no process exclude for Explorer, it will do the syncs and bring the folder up-to-date.
- You can always right-click on a folder and choose: GeoDrive/Recall All Directories. This will do basically the same thing as looking at folders using Explorer, except it will spawn many threads and do all subfolders very quickly.
8 Conclusion

GeoDrive was architected to be a simple and easy to use Windows interface for storing and accessing files in Dell EMC ECS and Atmos cloud storage platforms. GeoDrive presents buckets as Windows local drives so that existing Windows applications can be used without a re-write or development of an application to access the objects or files. An added benefit is that the use of GeoDrive is free of charge and can be easily downloaded from the Dell EMC support site. GeoDrive has numerous features relating to data throttling, a built-in load balancer, metadata rules, integration with Windows performance monitor, event viewer, explorer, and server failover clustering, and S3 support for versioning and ACL. Performance testing conducted have shown that GeoDrive is truly lightweight and offers tuning options to further enhance GeoDrive end-user experience.
A Technical support and resources

Dell.com/support is focused on meeting customer needs with proven services and support. Storage technical documents and videos provide expertise that helps to ensure customer success on Dell EMC storage platforms.

A.1 Related resources

- GeoDrive Product Support Page
  - https://support.emc.com/products/17166

- GeoDrive 2.0 Release Notes
  - https://support.emc.com/docu92033_Dell EMC GeoDrive 2.0 Release Notes.pdf

- GeoDrive 2.0 User Guide
  - https://support.emc.com/docu92034_Dell EMC GeoDrive 2.0 User Guide.pdf

- ECS Product Support Page
  - https://support.emc.com/products/37254

- ECS Overview and Architecture

- ECS Test Drive
  - https://portal.ecstestdrive.com/