Cloud Storage Benchmarking

Data Transfer Options and Available Software

Sara Willis

November 5, 2019

Contents

Part I

Introduction

Chapter 1

Overview

1.1 Testing Structure

The tests detailed in this document were single-file transfers. Various dummy files were generated using the `mkfile <size> <filename>` command and were individually transferred five times. After all transfers were complete, the mean and standard error were found for the estimated transfer speeds of each file size and plotted in R.

The six file sizes used were: 1 MB, 10 MB, 100 MB, 1 GB, 10 GB, 100 GB

An attempt was made to perform four different tests:

1. PC → Google Drive
2. Google Drive → PC
3. HPC filexfer Node → Google Drive
4. Google Drive → HPC filexfer Node

1.2 Profiling Scripts Accessibility

Access to the benchmarking scripts used for these analyses (when applicable), output csv files, and plotting scripts are available from Github.

1.3 Units

It may be worth mentioning that I am using the prefixes kilo, mega, and giga as the standard S.I. prefixes and not binary, so when I say a megabyte, I mean 10^6 bytes and not 1024^3 bytes. This is particularly noted since there are mixing of conventions; the command `mkfile` on Macs creates files with prefixes using the binary convention. The command `truncate` on Linux creates files, when specified with the option `MB` in base 10 metric units. To keep my results uniform, I extracted the exact size of each file in bytes and converted it to metric MB before computing speeds.

1.4 Warnings and Disclaimers

- This file is in flux as I add instructions on installing software and edit. This may mean wordy sentences, repetition, terrible grammar, and other unsightly word messes which I appologize for in advance.
- When downloading files from Google Drive, if you are overwriting an existing file by downloading one with an identical name, the program you are using may not immediately delete the preexisting copy. Instead, it may download the file in chunks which it will concatenate at the end of the download into a single file which will *then* overwrite the existing file. As a result, you may exceed your disk quota which may interrupt file transfers.
- The latest release of Cyberduck (V 7.1.0) has a bug related to the duo authentication and cannot currently connect to the filexfer node using sftp. The previous version (v 7.0.2), however, works. I will update this in the future if this is resolved. This appears to be a known issue.

- The installation of the CLI software is not done using virtualenv. It seems entirely probable that the software could be set up this way. For the time being, I have published notes specifically on how I set up my software and will update with different methodologies if it's decided they are more appropriate.

Chapter 2

Installing Software on HPC

Some of the tests done in this document use CLI programs and so need to be installed on HPC where users don't have root privileges. Some handy blanket instructions are here so they aren't repeated multiple times in the installation guides below.

2.1 Unpacking/Installing .rpm Files

On a Linux machine where you have root privileges, you would typically use the command `rpm` to unpack and install a file with a .rpm extension. On a headless server where you do not have permission to make a system-wide installation, this is not possible. Instead, use the command:

```
rpm2cpio <file-to-install.rpm> | cpio -idv
```

This will unpack the file without making any attempt at installation. Once you have unpacked the file, there should be an executable that you can use to run your program. If you don't want to point to the executable each time you run it, you can permanently add its location to your PATH variable.

2.2 Adding Executables to PATH

To add an executable to your PATH variable so you don't have to specify the path each time you call it:

```
vi ~/.bashrc # Opens a file that lets you permanently change your BASH environment
shift A # To edit
export PATH=$PATH:/path/to/executable # Enter on a blank line to add to PATH
shift : # Turns off editing
wq # Saves and quits
```

Part II

Google Drive

Chapter 1

Google Drive Overview

1.1 Untested Software

There are other options for connecting to Google Drive other than those tested in this section. Other options are listed below:

Subscription-Based Applications

- Globus CLI – **These tests have now been started and will be added to the document as results come in.**
- Odrive – Some features are freely available while others require a subscription. Without the additional features, working with files can be cumbersome and challenging. I also found the interface a bit more challenging than other options. It is possible in the future I will have some results for this application, but I'm focusing on the programs that I have found to be more user-friendly and are free.
- Netdrive
- Expandrive

Free Applications

The following options are freely available. I may have benchmarking results for these at some point.

- Google Drive File Sync
- Web Interface

1.2 Unzipping Files in Google Drive

I don't know all the options for unzipping files once they're in Google Drive. I will update if I become aware of more.

1.2.1 Google Chrome Extension

1.3 Personal vs. Shared Drives

The below information was taken from Google Support where more in-depth/comprehensive information can be found.

	My Drive	**Shared Drives**
Who can add files?	The person who owns My Drive	Any member with **Contributor** access or higher
What types of files can I add?	All file types	All file types (except Google Maps and Data Studio reports)
Who owns files and folders?	The individual who created the file or folder	The team
Can I move files and folders?	Yes, you can move files and folders around in My Drive	• If you have **Contributor** access or higher, you can move files from My Drive to a shared folder • If you have **Content manager** access or higher, you can move files and folders within a shared drive • If you want to move folders from My Drive to a shared drive, contact you GSuite administrator
Can I sync files to my computer?	Yes, using Drive File Stream or Backup and Sync	It depends on which sync solution you use: • Drive File Stream: Yes • Backup and Sync: No
How does sharing work?	Different users might see different files in a folder, depending on their access to individual files	All members of the shared drive see all files.
How long do files I delete stay in Trash?	Files or folders in Trash remain there until the user selects **Delete Forever**.	Each shared drive has its own Trash • Members with **Content manager** access and above can move files to trash • Files and folders in Trash are deleted forever after 30 days • Members with **Manager** access can permanently delete files before 30 days
Can I restore files?	Yes, if you're an owner of the file	Yes, if you have at least **Contributor** access

Chapter 2

Results

2.1 Best Performances

Results below were performed with default settings, no additional preferences or flags were specified to boost performance.

Transfer Type	File Size	Software	Average Transfer Speed	Estimated Transfer Time
Fastest Download Speed: Gdrive → Personal Computer	◦ 1G	Gdrive	194 MB/s	5 seconds
	◦ 10G	Gdrive	232 MB/s	43 seconds
	◦ 100G	Gdrive	237 MB/s	7 minutes
Fastest Download Speed: Gdrive → HPC	◦ 1G	Gdrive Rclone	189 MB/s 187 MB/s	5 seconds
	◦ 10G	Gdrive	242 MB/s	41 seconds
	◦ 100G	Cyberduck CLI Gdrive	243 MB/s 238 MB/s	7 minutes
Fastest Upload Speed: Personal Computer → Gdrive	◦ 1G	Cyberduck GUI	77 MB/s	13 seconds
	◦ 10G	Cyberduck GUI	66 MB/s	3 minutes
	◦ 100G	Cyberduck CLI	49 MB/s	34 minutes
Fastest Upload Speed: HPC → Gdrive	◦ 1G	Rclone	50 MB/s	20 seconds
	◦ 10G	Cyberduck CLI Rclone	61 MB/s 55 MB/s	3 minutes
	◦ 100G	Rclone	55 MB/s	30 minutes

2.2 Software Rankings

Transfer Type	Ranking	1G	10G	100G
Download Speeds Gdrive → Personal Computer	1	Gdrive 194 MB/s	Gdrive 232 MB/s	Gdrive 237 MB/s
	2	Cyberduck GUI 77 MB/s	Cyberduck GUI 77 MB/s	Globus 81 MB/s
	3	Rclone 65 MB/s	Globus 72 MB/s	Cyberduck GUI 72 MB/s
	4	Globus 47 MB/s	Rclone 68 MB/s	Rclone 66 MB/s
	5	Cyberduck CLI 45 MB/s	Cyberduck CLI 52 MB/s	Cyberduck CLI 51 MB/s
Download Speeds: Gdrive → HPC	1	Gdrive 189 MB/s	Gdrive 242 MB/s	Cyberduck CLI 243 MB/s
	2	Rclone 187 MB/s	Rclone 142 MB/s	Gdrive 238 MB/s
	3	Cyberduck GUI 74 MB/s	Cyberduck GUI 126 MB/s	Rclone 138 MB/s
	4	Globus 51 MB/s	Globus 70 MB/s	Cyberduck GUI 88 MB/s
	5	Cyberduck CLI 38 MB/s	Cyberduck CLI 35 MB/s	Globus 72 MB/s
Upload Speeds: Personal Computer → Gdrive	1	Cyberduck GUI 77 MB/s	Cyberduck GUI 66 MB/s	Cyberduck GUI 53 MB/s
	2	Cyberduck CLI 42 MB/s	Cyberduck CLI 52 MB/s	Cyberduck CLI 49 MB/s
	3	Globus 21 MB/s	Globus 26 MB/s	Globus 30 MB/s
	4	Rclone 20 MB/s	Rclone 20 MB/s	Rclone 19 MB/s
	5	Gdrive 19 MB/s	Gdrive 18 MB/s	Gdrive 19 MB/s
Upload Speeds: HPC → Gdrive	1	Rclone 50 MB/s	Cyberduck CLI 61 MB/s	Rclone 55 MB/s
	2	Cyberduck GUI 45 MB/s	Rclone 55 MB/s	Cyberduck GUI 48 MB/s
	3	Cyberduck CLI 37 MB/s	Cyberduck GUI 52 MB/s	Cyberduck CLI 36 MB/s
	4	Globus 22 MB/s	Globus 25 MB/s	Gdrive 18 MB/s
	5	Gdrive 15 MB/s	Gdrive 16 MB/s	Globus 14 MB/s

2.3 Software Pros and Cons

Pros	Cons
Globus	
◦ User-friendly interface ◦ Sends email confirmations when file transfers complete ◦ Versatility with both a CLI and Web version	◦ Slow upload speeds
Cyberduck CLI	
◦ Excellent download speeds for large files to HPC ◦ Reasonable upload speeds	◦ Consistently slow download speeds for anything under 100 GB
Cyberduck GUI	
◦ Convenient drag-and-drop interface ◦ Relatively easy setup ◦ Fastest upload speeds	◦ Some hiccups in user interface which involve navigating errors ◦ As of September 19, 2019, the newest version has a bug that will not allow duo-authentication which means it can't connect to the filexfer node ◦ Not Linux compatible ◦ Does not display progress for files travelling between remote servers
Gdrive	
◦ Easy syntax ◦ Uses a file ID system which allows for storing multiple files with the same filename in Google Drive ◦ Very fast download speeds ◦ Modifiable options to optimize transfer speeds	◦ Slow for file uploads ◦ Cannot download/delete files from/in Google Drive using filename due to file ID system ◦ Not robust to interrupted connections. Long uploads may get "connection reset by peer" errors which will crash the transfer
Rclone	
◦ Very customizable with a large number of user flags	

Chapter 3

Software: Tests, Installation, and Results

3.1 Globus Online Interface

These tests were performed using the temporary Globus Google Drive connector. There is a new permanent endpoint available which can be found using arizonahpc-dtn1. The tests performed with the new connection are located in the Globus CLI section. There doesn't appear to be any difference in transfer speed between Globus CLI and the web interface, so users have the option to set up their transfers how they want. Since this section is tedious because it requires manual job submissions that cannot be automated and there doesn't appear to be a difference between interfaces, I will leave these tests as-is for the time being, keeping the results from the test connection. The set up for the new endpoint is the same, except for the different identifier which should be used in place of sdmz-dtn-3. For all permanent endpoint results, check under Globus CLI..

Each job was submitted independently and its transfer speed was pulled from Globus' Activity summary.

Each transfer was done without any modification to any of the Globus settings. I don't know if it's possible to alter Globus to get faster transfer speeds or if there's any sort of bandwidth throttling for the Google Drive uploads that's making the transfers slow, this could be something to look into. I do know that Globus' connection to Google Drive is fairly new, so this potentially could explain the upload speeds.

In terms of some possible modificiations that may speed up time to transfer completion (note, not transfer speeds themselves): Globus, by default, verifies file integrity post-transfer. Some tests with AWS S3 have shown that this process can be rather time-intensive. Disabling this feature may help with the warning that if some file corruption takes place during the process, Globus will not retry.

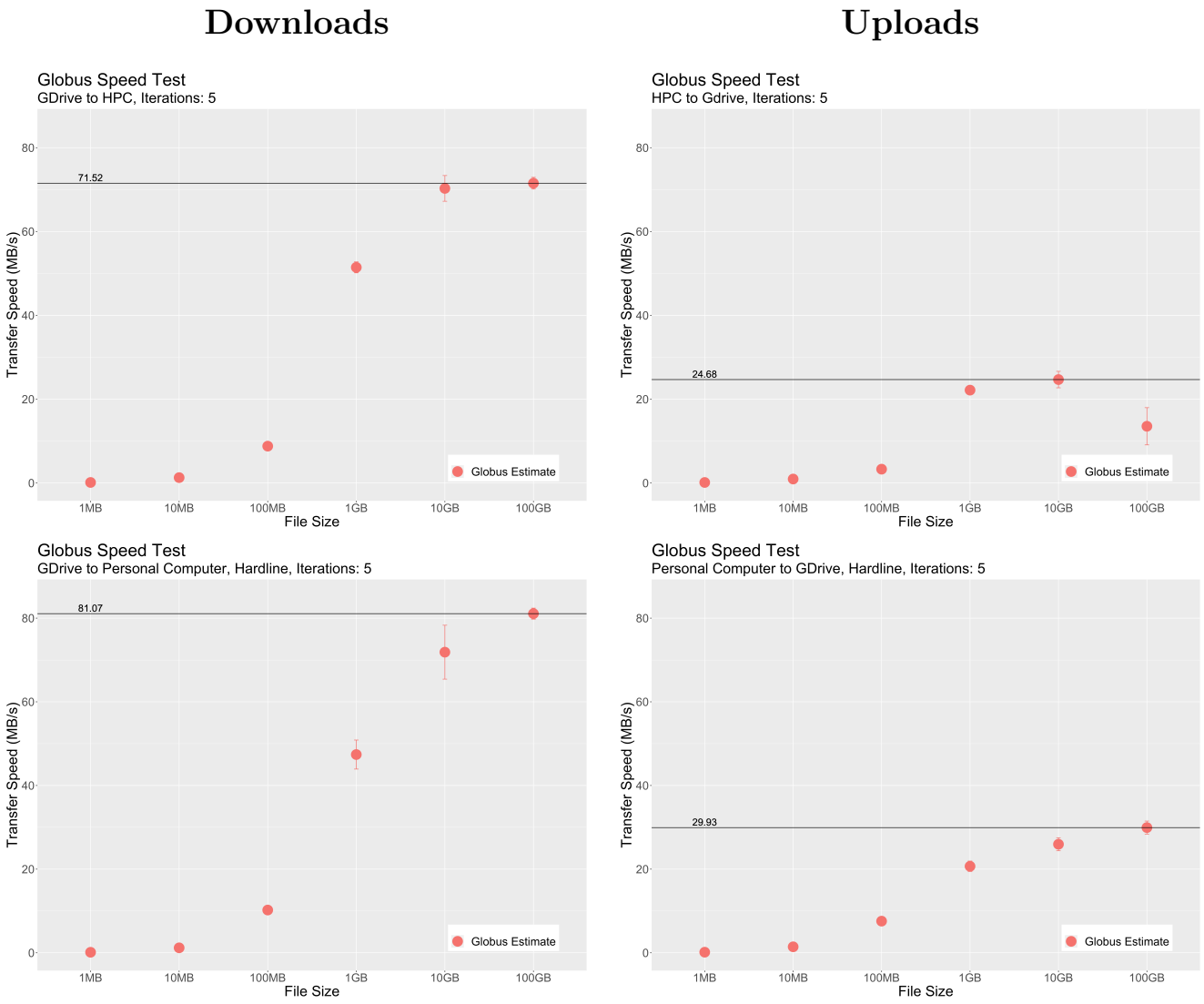
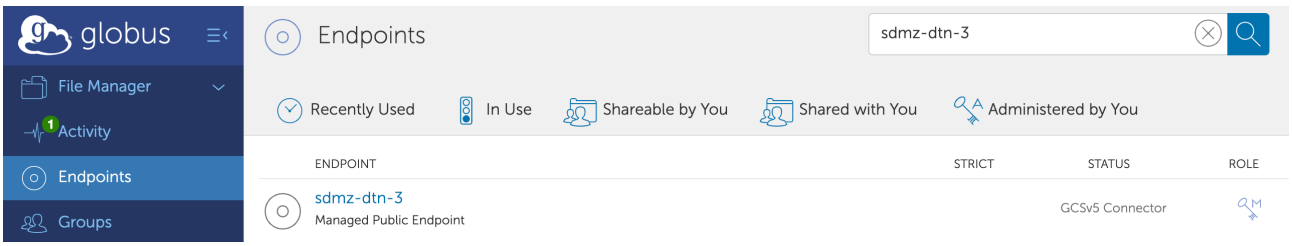


Figure 3.1: Download speeds from Gdrive to both my PC and to HPC were markedly faster than Upload speeds going the other direction. The maximum mean profiled speed is plotted as a horizontal line.

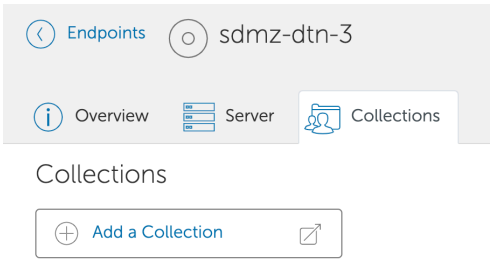
3.1.1 Software Access and Usage

There are instructions on how to set up Globus and link it to endpoints in our Online Documentation. Once you've gotten your endpoint(s) set up (either to the HPC filexfer node, your personal computer, or both), you can add your UofA Google Drive as an endpoint via the following steps:

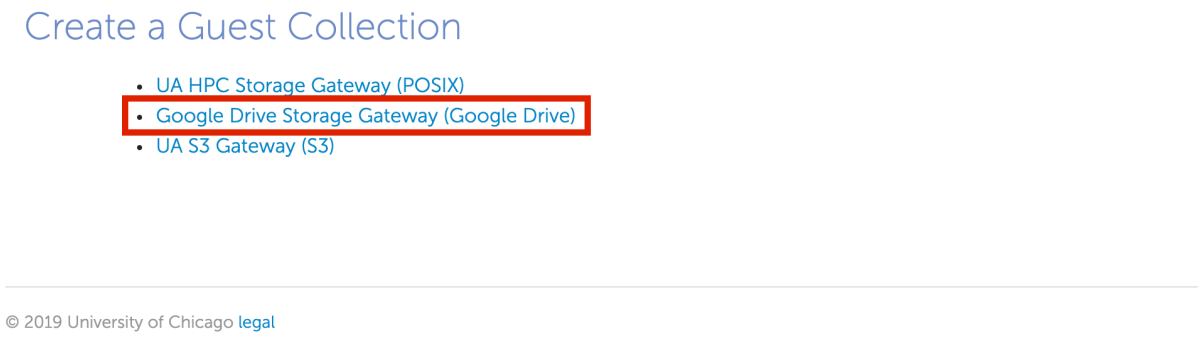
1. Search for the endpoint sdmz-dtn-3 and click the resulting endpoint



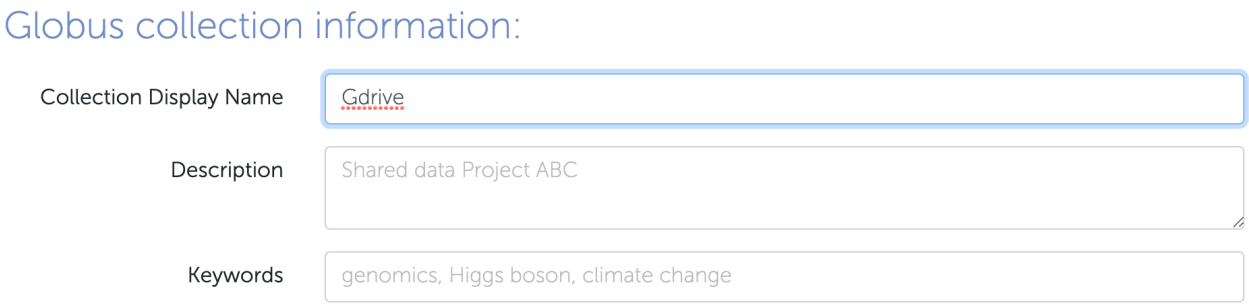
2. Under the **Collections** tab, select **Add a Collection**



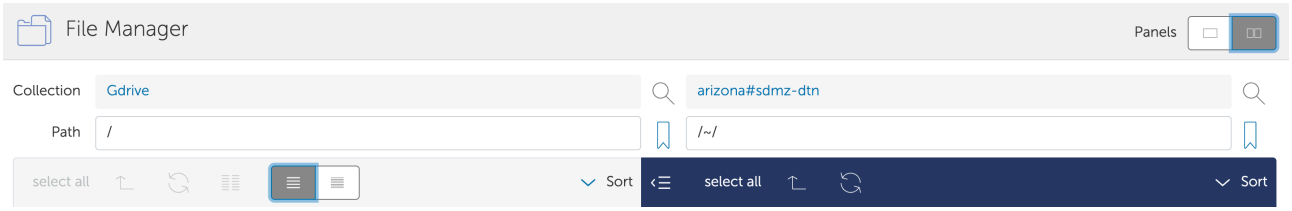
3. Select Google Drive



4. Name your **Collection Display Name** something descriptive and identifying for future use



5. You can now make transfers to/from Google Drive! Set up transfers in the **File Manager**, accessible from the side menu



3.2 Globus CLI - Permanent Endpoint

This section was created after the Globus Google Drive connector was officially purchased and the 100 gigabit endpoint permanently established taking the place of the temporary one. The same script that's used in the AWS tests (see: end of this document) was used to transfer files.

When using Globus CLI, you're submitting jobs as you might using the web interface, so you're able to log into the Globus web console to track your job's status, cancel your job, etc. You will also receive an email each time you transfer a file. There is an option to turn off email notifications:

```
$ profile -n off
```

This will turn off all notifications, not just for successful transfers, so you won't be notified if something goes wrong. This may be nice, however, for users who are transferring a large number of files.

During the transfer of some of the 100GB files from HPC to Google Drive, I receieved the error: "Warning: endpoint too busy." This didn't cancel the file transfer, but rather gave me a small warning icon and the transfer speed slowed down considerably. Eventually, the warning went away.

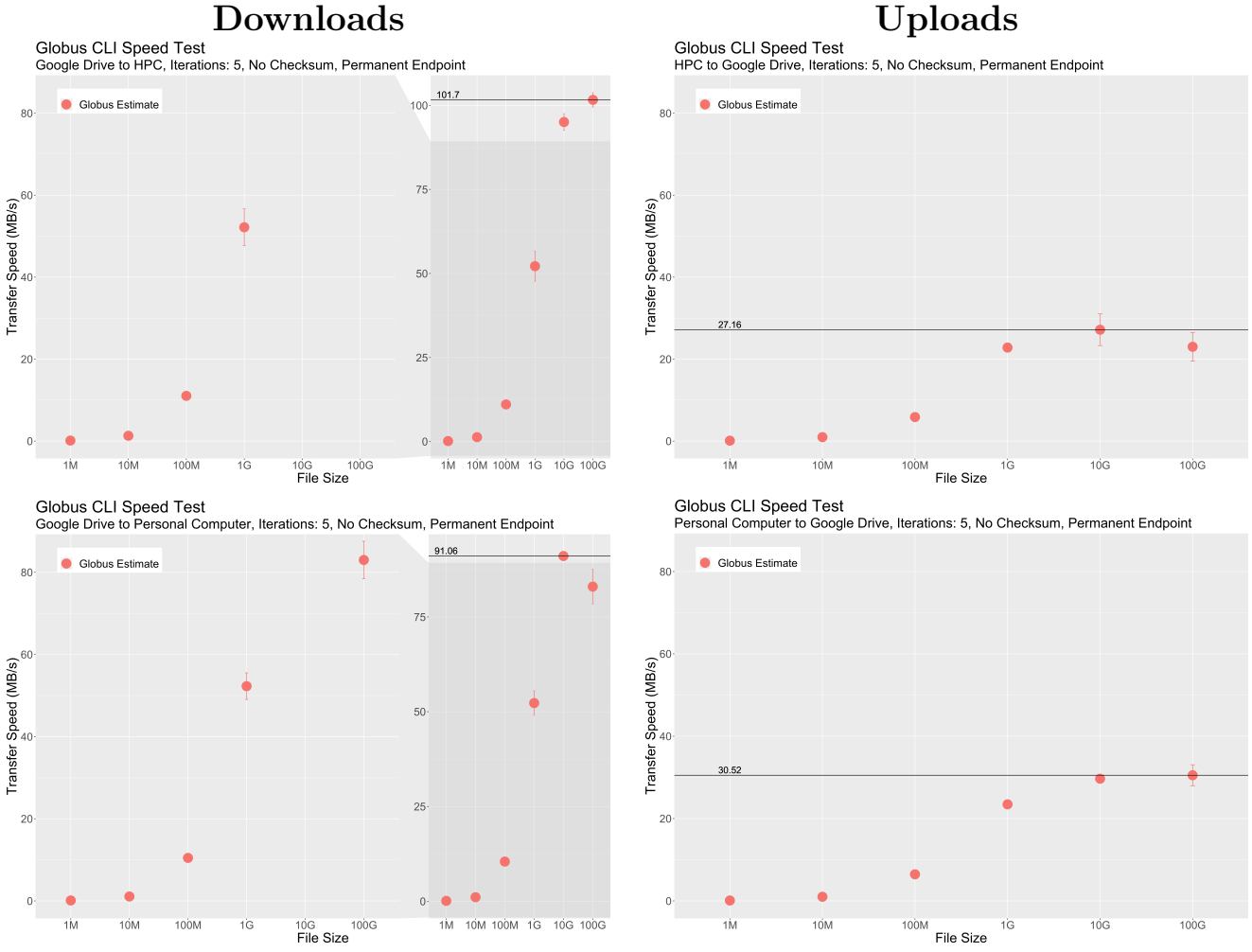


Figure 3.2

3.3 Cyberduck CLI

3.3.1 Testing and Results

To test the data transfer speeds of the Cyberduck CLI, I recorded two separate estimates. Both were extracted from a python script which:

1. Used a profiler, cProfile, to capture the time taken to successfully transfer the file. The exact size of the file was then extracted and the total time was divided by the filesize to get the mean transfer speed.

2. Extracted instantaneous duck estimates which were printed by the application to stdout. The average was then taken over all instantaneous outputs.

Unsurprisingly, there was a reasonably large discrepancy between the predicted profiler speed and the averaged duck output. I trust the profiler speed more given the methodology (which feels dodgy for the duck estimates) and because other tests have shown the profiling speed measuring up really well against estimates given by other applications.

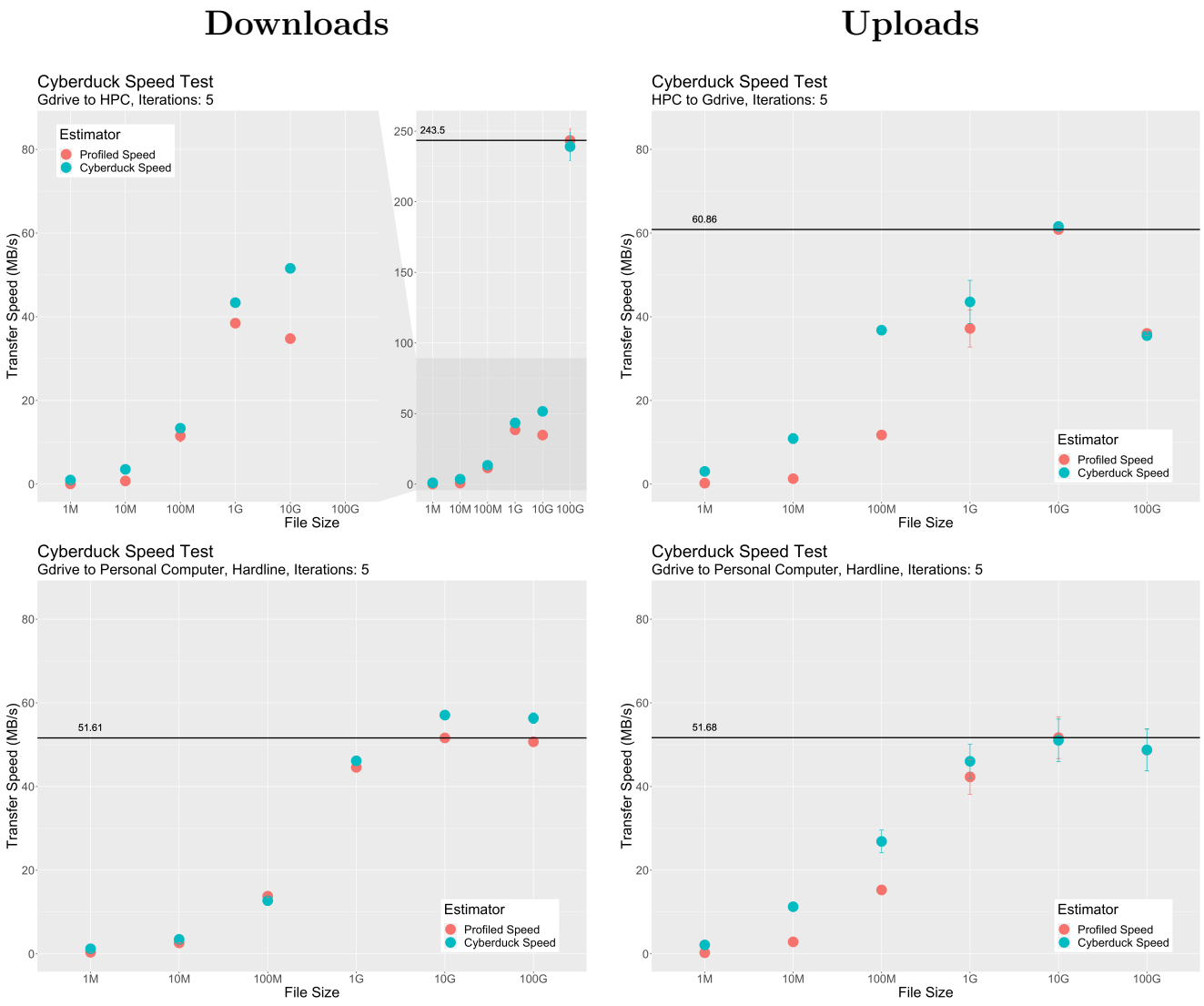


Figure 3.3: For the most part, the results were fairly uniform with download speeds being slightly higher than upload speeds. There was an interesting effect hitting the 100GB range, however, when transferring files off of Google Drive to the filexfer node. Three independent tests confirmed the transfer rate jumped to over 200 MB/s. These results did not change when a new 100GB file was used and were consistent between trials performed on different days.

It's not clear to me whether there are any settings that could boost the speed of file transfers. Duck does have some user options, but fiddling with them I haven't gotten noticeably different results.

3.3.2 Installation

These instructions are for users who do not have root privileges on HPC and who want to set up their own personal copy of duck in their home directory.

1. Go to https://repo.cyberduck.io/stable/x86_64/

2. Download the latest version

3. sftp to filexfer node and put file into bin. In my case, I made a directory called Duck in bin where I stored the zipped file. Unzipping duck generates a lot of files and keeping them partitioned isn't a bad idea.

4. Unpack the rpm file and add the executable to your PATH variable. For details, see Installing Software on HPC

3.3.3 Usage

Once the program has been unpacked and the system knows where to find the program, you can set it up to connect with your personal Google Drive account.

You should only need to connect it to your personal Google Drive account the first time you use it. An example is shown below:

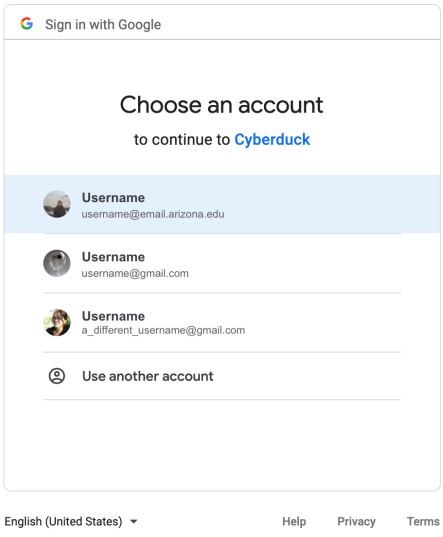
1. In the terminal, specify the direction of your file transfer and any additional options

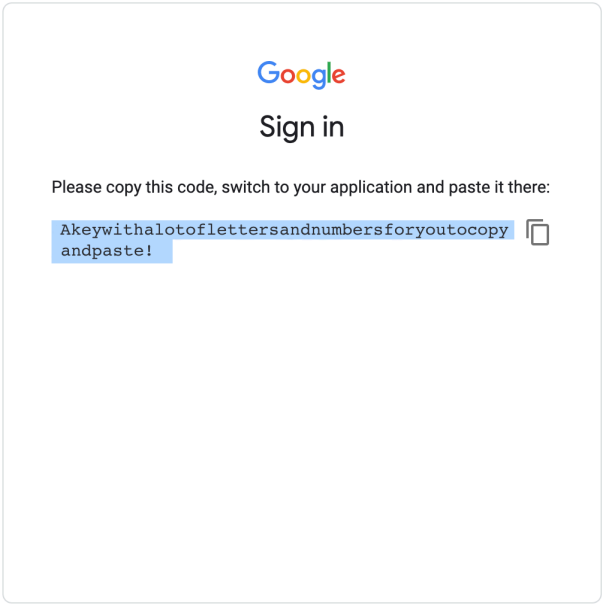
```
duck --username NetID@email.arizona.edu --download "googledrive:My Drive/<remote_filename>" \
<local_filename>
```

2. Copy/paste the url the program gives you into your browser

```
Resolving www.googleapis.com...GLib-GIO-Message: 00:00:00.000: Using the 'memory' GSettings
backend.  Your settings will not be saved or shared with other applications.
Google Drive connection opened...

https://accounts.google.com/o/oauth2/auth?client_id=000000000000.apps.googleusercontent.com&
redirect_uri=urn:ietf:wg:oauth:0.0:oob&response_type=code&scope=https://www.googleapis.com
/auth/drive&state=00000000
```

3. Select the relevant gdrive account and copy the key that appears in the browser





4. Paste the copied key into the terminal. Note: you will not see any characters when you paste the key in. This is normal behavior and everything is working.

```
OAuth2 Authentication. Paste the authentication code from your web browser.
Authentication Code:
```

5. If you don't want to authenticate again in the future, save the password with "y" and always use the option `--username NetID@email.arizona.edu`.

```
WARNING! Passwords are stored in plain text in ~/.duck/credentials.
Save password (y/n): y
```

For more information on Duck usage:

```
duck --help
```

Warning: When downloading files, if you only specify the filename without including a path, it will save to the app folder under opt/duck/ which already has a large number of files in it. Specifying the absolute path with the filename will avoid this problem. You can also specify the relative path, but you will need to remember that the path is relative to the opt/duck directory and not your working directory.

Warning: When downloading, even if you include the `--existing overwrite` option, the file that you're going to overwrite isn't immediately overwritten. Cyberduck creates a folder where it downloads your file in chunks and once all the chunks have been downloaded, they then get concatenated into the file that will overwrite the previously-existing one. This means that if you're low on space where you're saving your file, you may run exceed your storage quota.

3.4 Cyberduck GUI

The Cyberduck GUI is available for Windows and Mac but, unfortunately, is not Linux-compatible.

The speed tests run using the Cyberduck GUI were a bit fuzzier in methodology than the command-line programs which could be automated. As a consequence, there aren't any scripts that are available to run profiling tests. Instead, I manually transferred the files by double-clicking them in the Cyberduck window to initiate the transfers and started a simple timing program written in Python. Since the timing relied on my reflexes, the precision will be worse than previous tests, but for larger transfers shouldn't lead to a tremendous amount of error. Cyberduck is fast so the smaller file transfers (10M) should be taken with a grain of salt. The 1M files were roughly instantaneous, so no tests were performed in this range.

One other drawback of using the GUI to transfer files between remote machines is you will not be given any indication of the transfer's progress. When copying files between a remote machine and your personal computer, Cyberduck has a loading bar that gives you an idea of how far the transfer has progressed. When transferring between HPC and Google Drive, you will only see the message "Copying <source filename>to <dest filename>until the transfer is complete.

One final drawback to downloading from Google Drive to HPC is that (at least for v. 7.0.2 on my specific machine, a Mac) an error will pop up at the end of the transfer that says `Error: Unknown application error.` It appears that the transfers are successful, at least in my tests, if you choose "Cancel," but will fail if you press "Try Again." This is only for downloads and does not occur for uploads.

Despite the drawbacks mentioned above, it turns out it is one of the fastest uploaders with speeds approaching 80 MB/s. This is substantially more than other programs profiled here and makes this program an asset in the challenge of getting large quantities of data onto Google Drive.

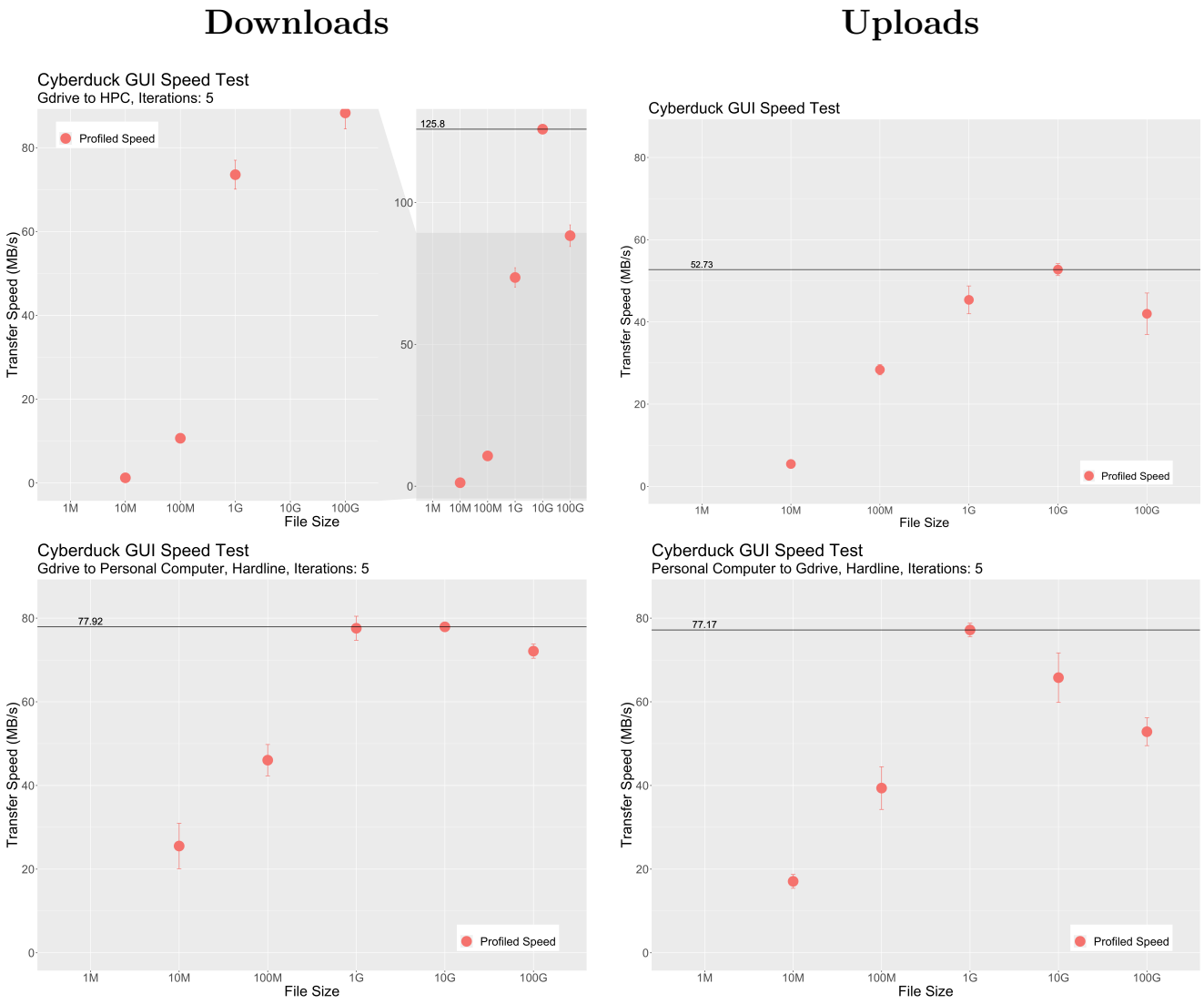


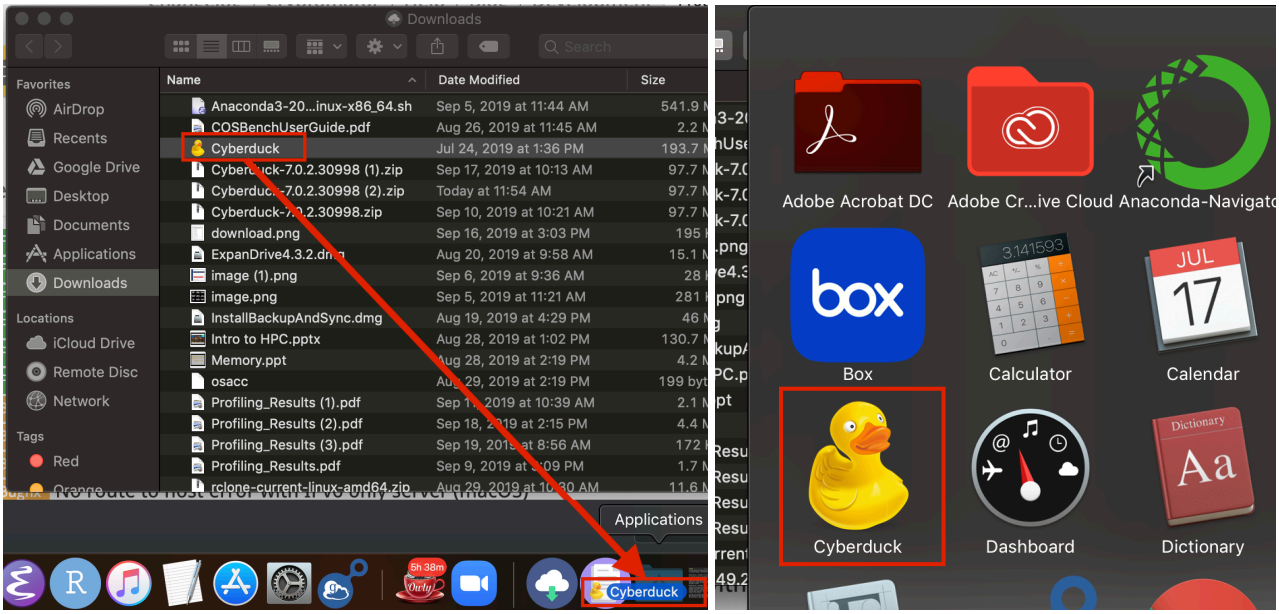
Figure 3.4

3.4.1 Installation and Usage

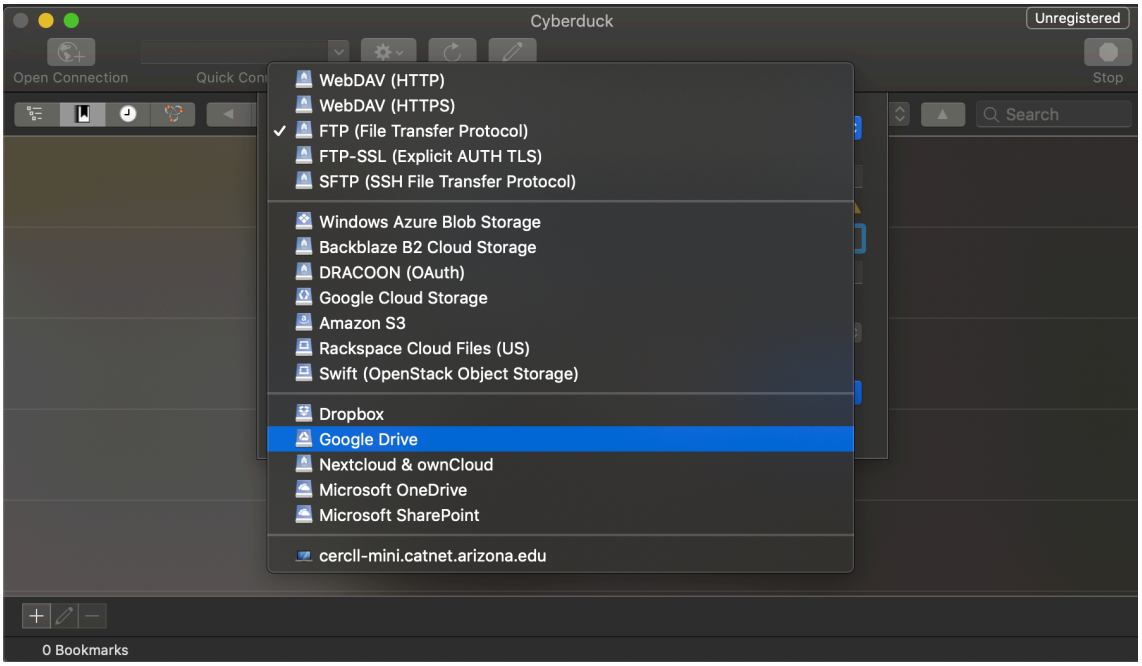
Mac

Installation is fairly simple. Various versions of Cyberduck are available for download. Until the current bug is fixed that successfully allows for Duo Mobile authentication, I recommend using the previous version 7.0.2.

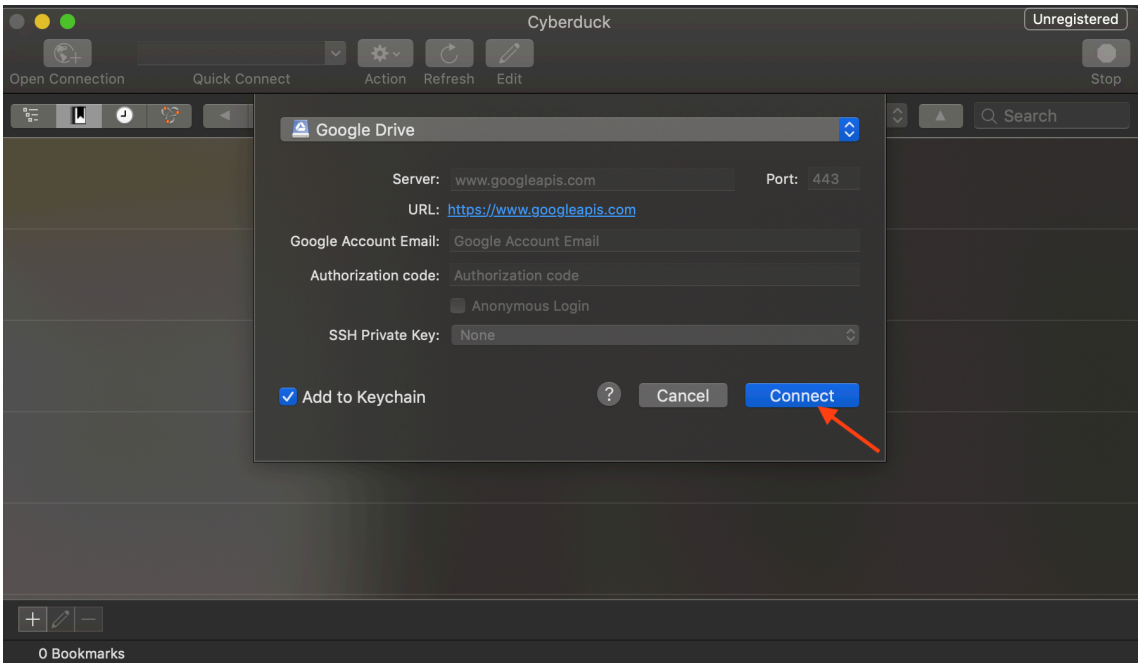
1. Download the zipped file, unpack, drag/drop the program into your applications folder, and double-click the icon to start Cyberduck.



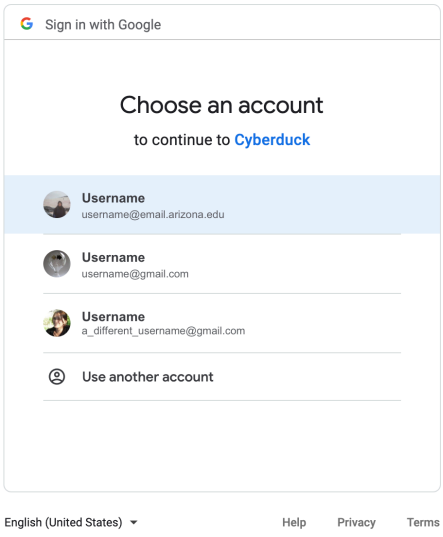
2. To establish a connection with Google Drive, use the drop-down menu by clicking “Open Connection”.

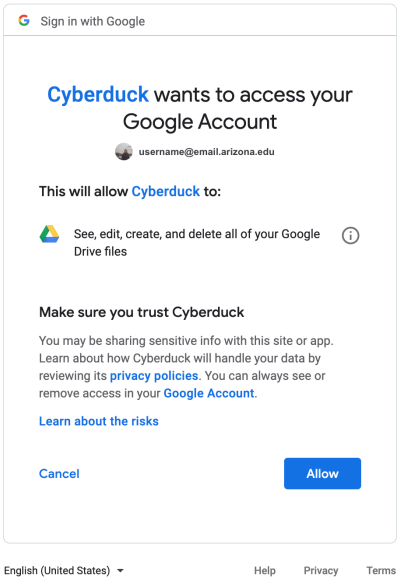


3. The default settings do not need to be edited; click Connect to proceed.

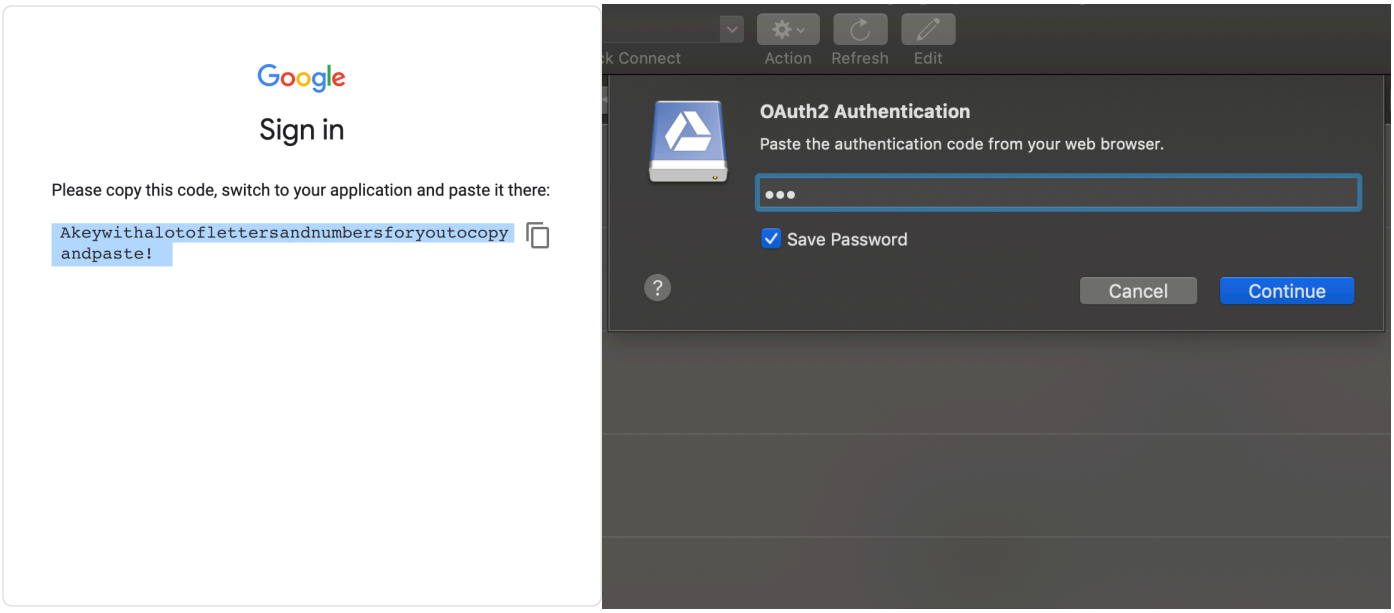


4. Select the account you'd like to connect with and grant Cyberduck access

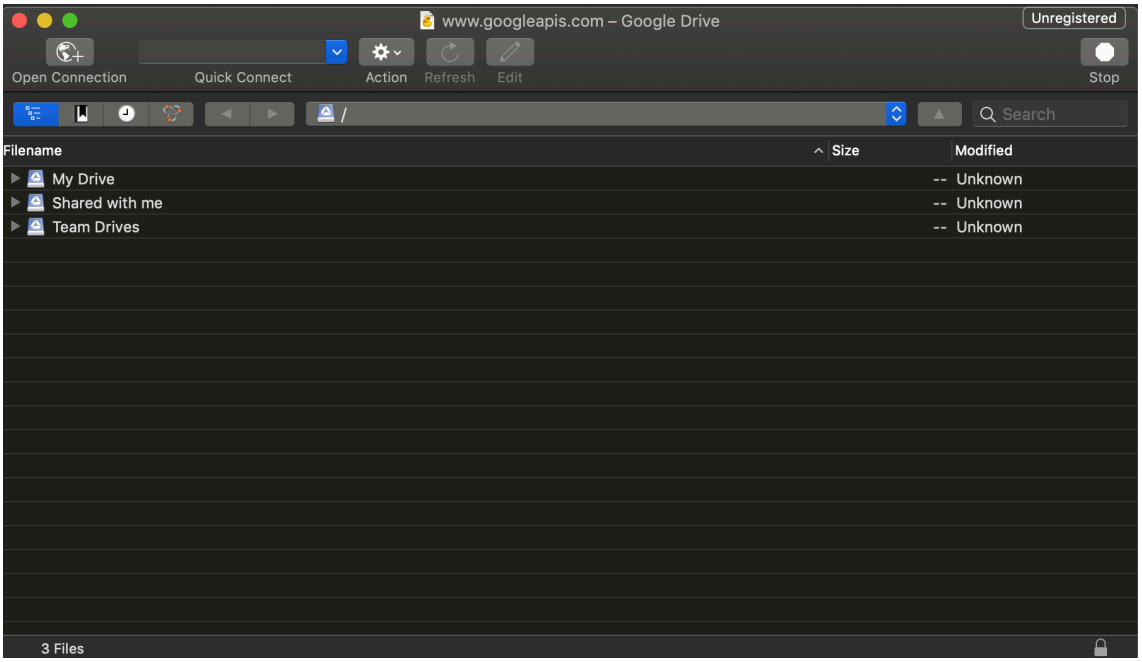




5. Copy and paste the code that appears in your browser



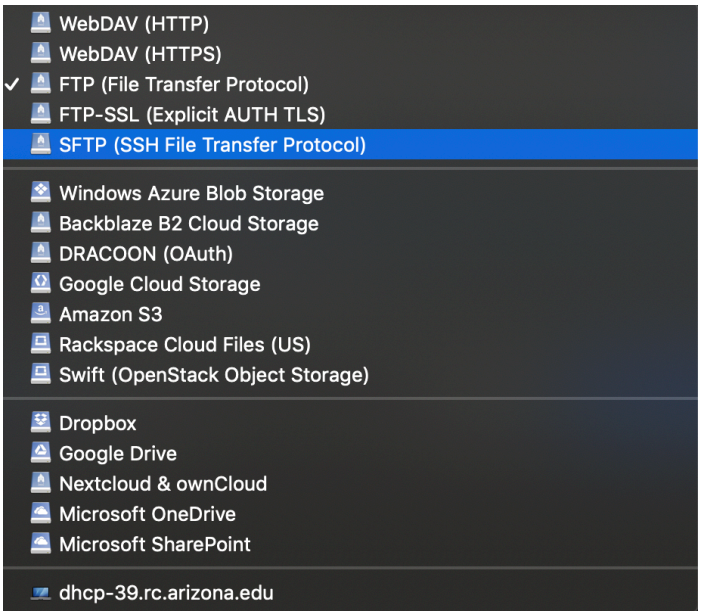
6. You are now connected to Google Drive!



To make transfers between your personal computer and your Google Drive account, you can simply drag-and-drop files. To transfer files between Google Drive and HPC, you will need to open a second window to connect to HPC's filexfer node:

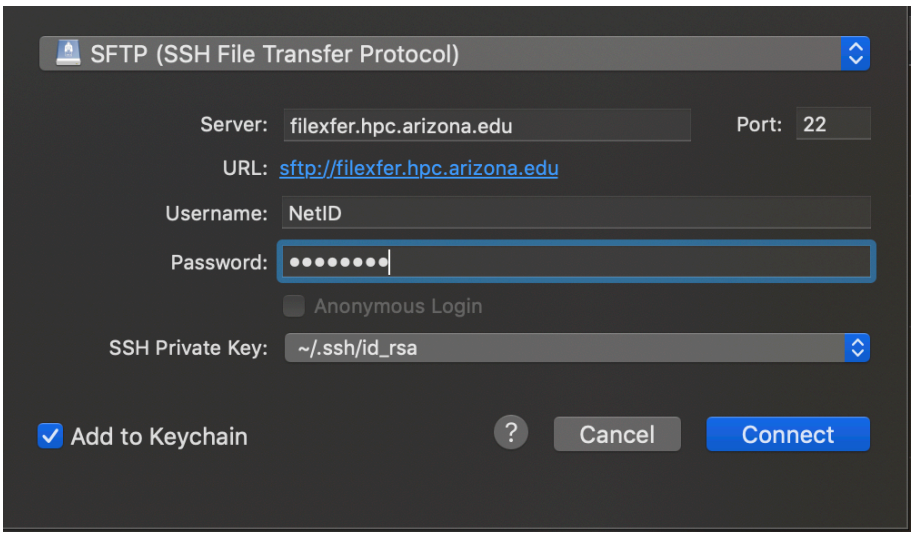
1. Go to file and select **New Browser**

2. Under **Open Connection** select **SFTP**

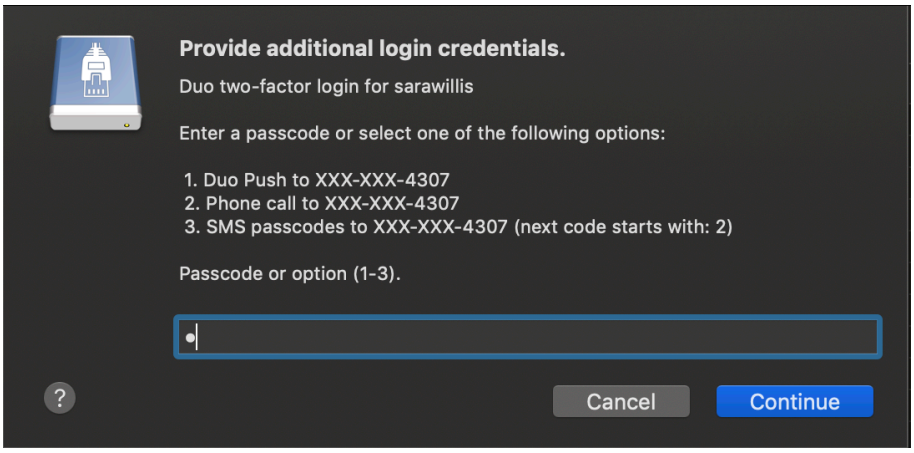


3. Enter your credentials to connect with the filexfer node:

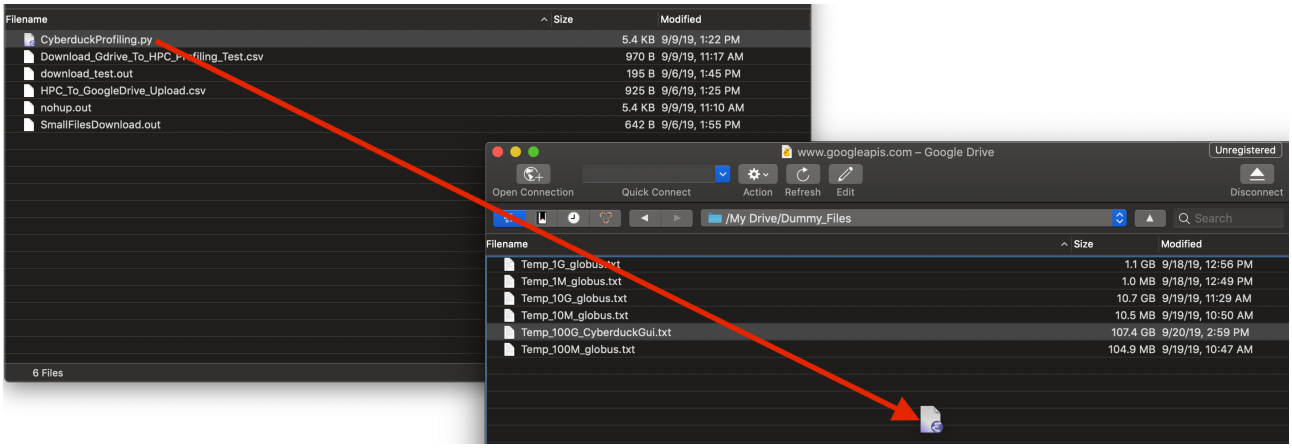
   - Server: filexfer.hpc.arizona.edu
   - Port: 22 (default)
   - Username: <NetID>
   - Password: <Your password>



4. You will be prompted to enter an option for two-factor authentication



5. You are now connected! To initiate a transfer, drag-and-drop files between the Google Drive and filexfer windows



3.5 Gdrive

Gdrive is another command-line interface program built to transfer files to Google Drive. It was built by an engineer who was working at Google at the time and so is somewhat "official."

- Plusses: It's relatively easy to work with and the download speeds are excellent.
- Minuses: Uploads are not particularly fast. There are some options you can tweak, specifically the chunk size, but it's not (at least for me) intuitive how to optimize this. This would require additional benchmarking.

One other factor that differentiates gdrive from other applications is that it utilizes ID's when transferring instead of filenames. When uploading to Google Drive, you can specify a filename and, unlike other applications, this will not overwrite any documents already stored on Google Drive but will add an additional file with the same name. To download or delete a file, however, you cannot specify a name, you must specify the file ID. This can be accessed using the command `gdrive list`. Used in conjunction with `| grep <filename>`, you can easily pull the ID.

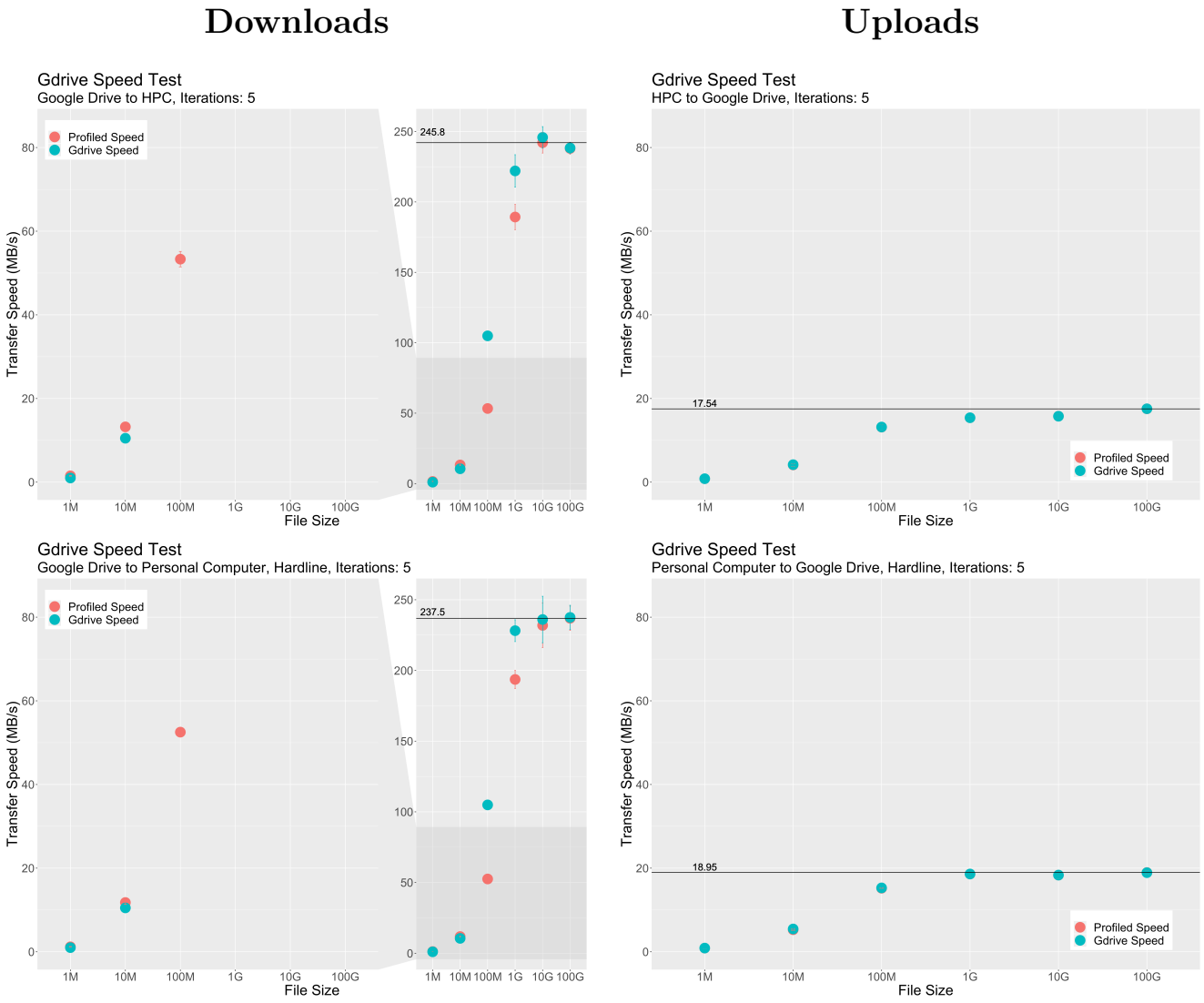


Figure 3.5: Gdrive had some of the best download speeds of any of the programs tested. There was some discrepancy between the profiled speeds and the gdrive-predicted speeds for downloads. This is likely due to how rapidly the files were transferred. Gdrive seems to have a lower bound for estimates of transfer time which I assume to be 1 second, so if a file takes less time than 1 second, it estimates the the transfer speed as file size/second, which is different from the exact measurements of the profiler. The discrepancies are far less pronounced for large file transfers and file uploads which are much slower

3.5.1 Chunk Size Optimization

As I mentioned above, changing the chunk size used in the transfer process is possible with Gdrive. I've written a script that alters chunk size and performs various upload/download tests while maintaining a constant file size. I will upload the results

soon, but will prioritize completing the file size transfer tests for other programs before doing so.

3.5.2 Installation

Gdrive's Github gives a good overview of the installation process for various operating systems here: https://github.com/gdrive-org/gdrive

To run Gdrive on HPC, download the Linux 64-bit file, transfer it to HPC, and add executable permissions:

```
chmod +x gdrive
```

You may also want to change the filename to something like `Gdrive`.

3.5.3 Usage

The command

```
gdrive help
```

gives a list of relevant commands. When used in conjunction with a command, gdrive prints specific instructions for a command, e.g.:

```
(base) cc-ea-lafrese:Desktop sarawillis$ gdrive help download
Download file or directory
gdrive [global] download [options] <fileId>

global:
  -c, --config <configDir>         Application path, default: /Users/sarawillis/.gdrive
  ...
```

3.6 RClone

3.6.1 Testing and Results

There are many flags that can be used with rclone. To see all the flags use the command:

```
rclone help flags
```

I haven't done enough testing to say which flags optimize transfer speeds, but they do have an option that allows you to set chunk size like Gdrive. Unlike Gdrive, you can turn off chunking completely.

One thing to note: if you are attempting to upload files to Google Drive using RClone, it will not overwrite existing files if they are identical. Instead, it will silently quit without throwing any exceptions.

Rclone Transfers without Flag Specifications

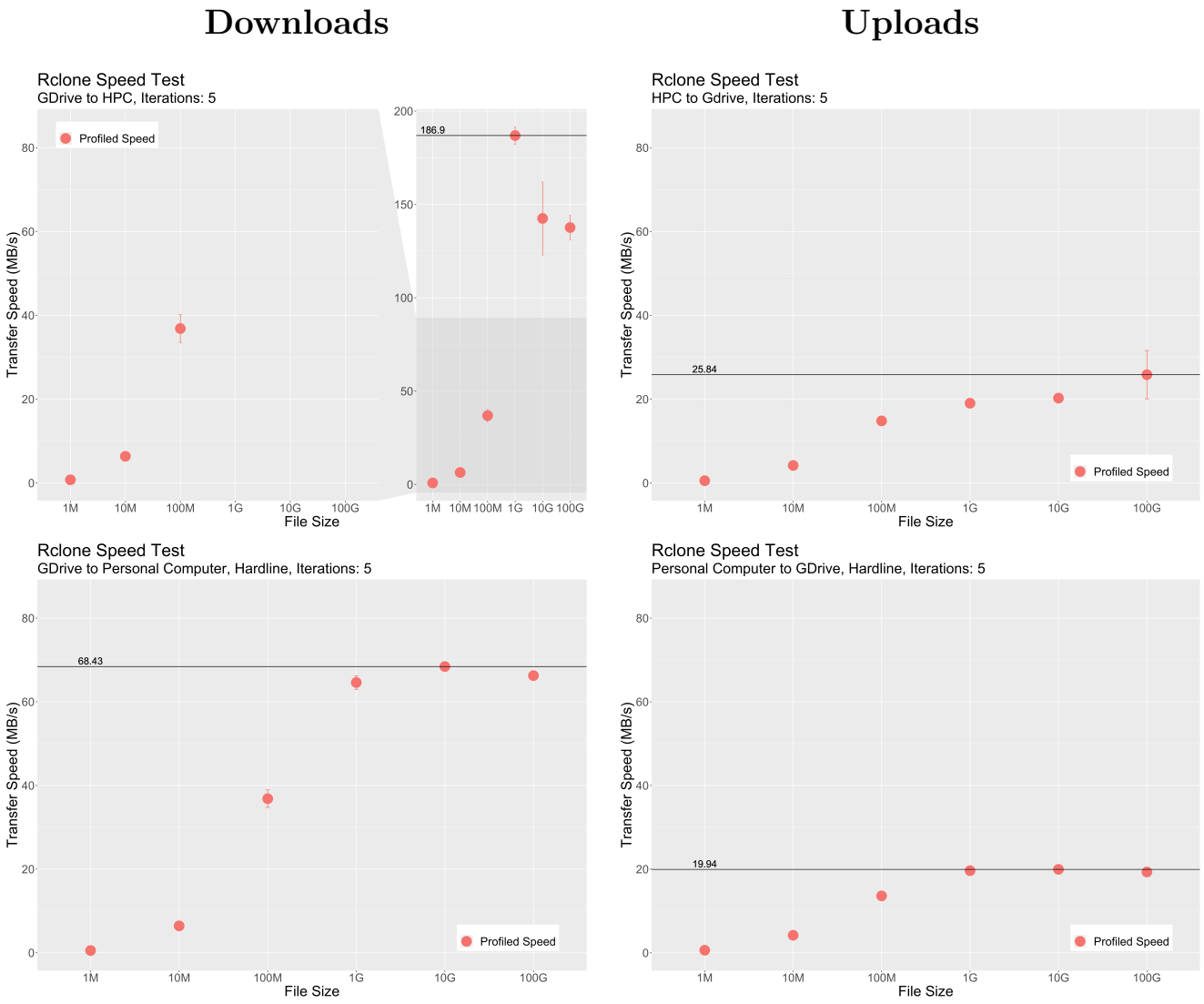


Figure 3.6: Rclone upload speeds are not particularly good when no flags are specified. Later tests that altered the upload-specific chunk size improved speeds. Examples will be provided below. Download speeds from Google Drive to HPC were very good, approaching those that were given by Cyberduck.

Rclone Transfers with Flags – HPC to Google Drive

Uploading to Google Drive from either HPC or your Personal Computer can be a slow enterprise. To speed up the process,

there are flags available to customize your rclone commands to optimize performance. To date, I have not run benchmarking tests to optimize these flags. This may be done at some later date. Until then, I will share my experiences modifying my upload speeds from HPC to Google Drive using rclone commands with the warning that the options that I have selected for myself are for demonstration purposes and are not recommendations for optimal performance.

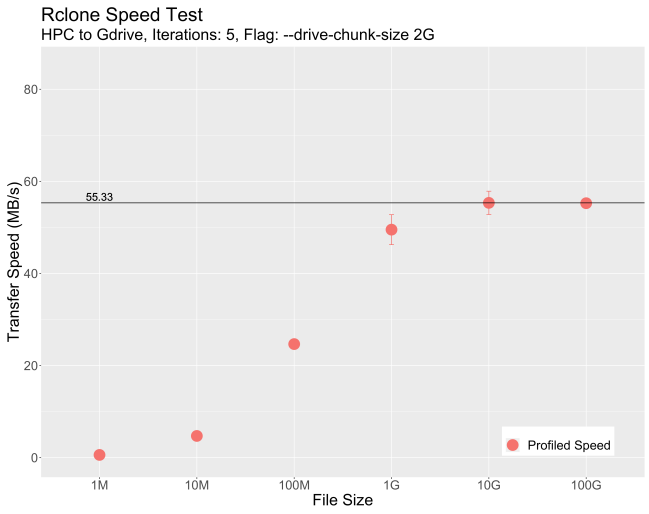


Figure 3.7: Rclone upload speeds improved when flags were specified. Clockwise from upper left:
1. `--drive-chunk-size 2G` : increased the chunk size of the file upload to 2GB/chunk which nearly tripled the upload speed. Further tests should be done to determine whether there is an optimal chunk size.

3.6.2 Installation

Rclone setup requires some additional steps if you want to use your own key. I'll include instructions on that below. Fortunately, Rclone has good documentation.

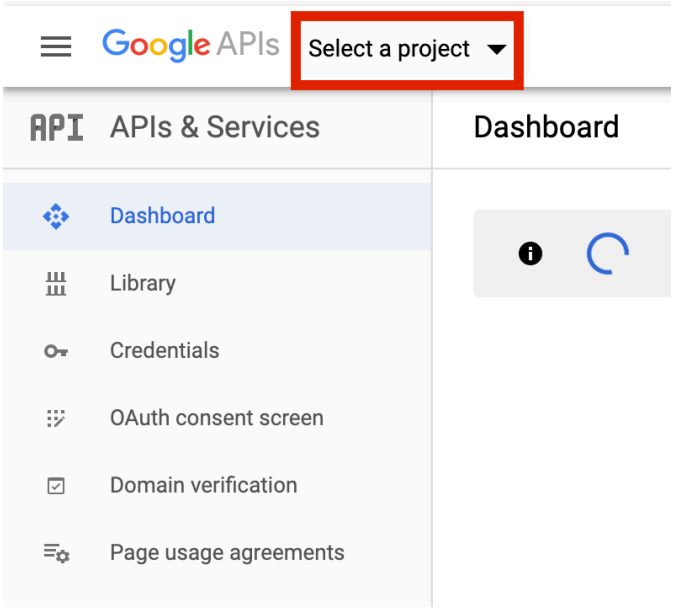
1. Download the `.rpm` file from https://rclone.org/downloads/
2. Use sftp to transfer rpm file to filexfer node
3. `Rpm2cpio <filename>.rpm | cpio -idv` # To unpack the rpm file without installation which requires root privileges
4. `cd usr/bin` # usr is located in the unpacked file
5. `./rclone config` # and follow prompts:

```
No remotes found - make a new one
n) New remote
s) Set configuration password
q) Quit config
n/s/q> n

name> <something descriptive>

Type of storage to configure.
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value
 1 / 1Fichier
   \ "fichier"
 2 / Alias for an existing remote
   \ "alias"
...
Storage> 12

** See help for drive backend at: https://rclone.org/drive/ **
```

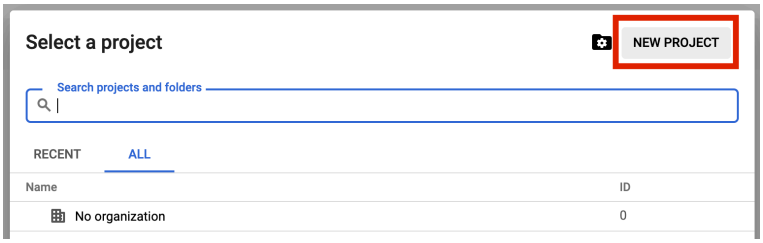
```
Google Application Client Id
Setting your own is recommended.
See https://rclone.org/drive/#making-your-own-client-id for how to create your own.
If you leave this blank, it will use an internal key which is low performance.
Enter a string value. Press Enter for the default ("").
```

As rclone states, you can either create your own client ID or you can use the default. I haven't done any testing to determine the speeds using the public key vs. creating your own. Note: Thus far, I have not been successful in creating a client ID with my UofA profile and have only been successful when using a personal account. To create your own key, follow the instructions on the rclone site which I'll replicate below:

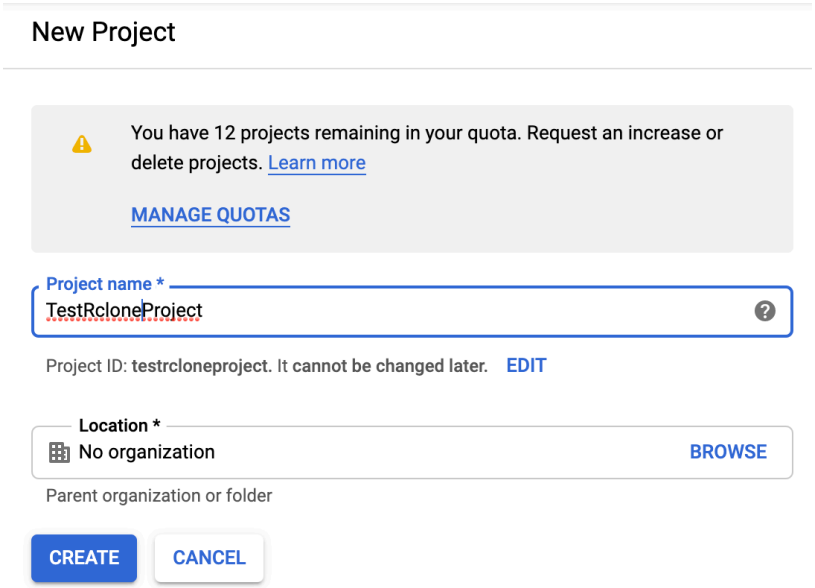
1. Go to Google's API console
2. click **Select a project**



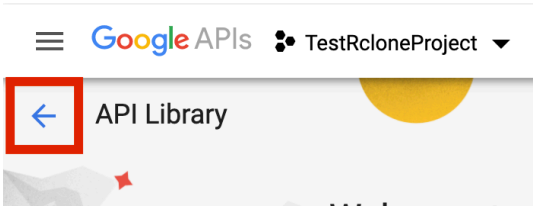
3. Select "New Project"



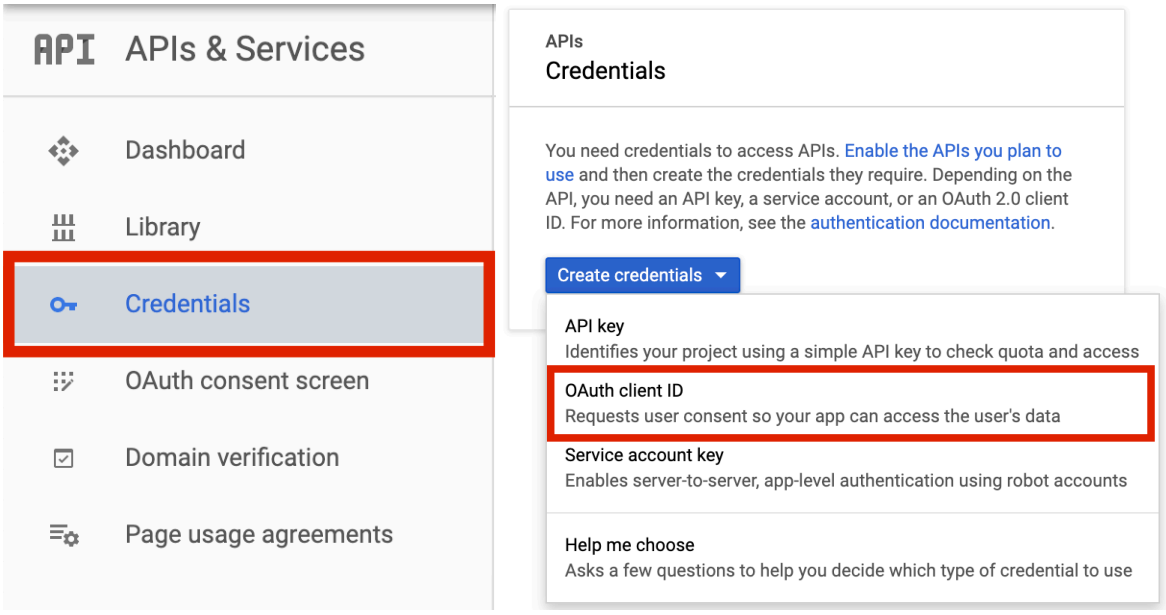
4. Give your project a descriptive name and then click **Create**



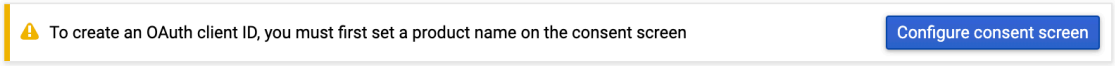
5. This will take you to an API Library page. To continue with this process, click the back button

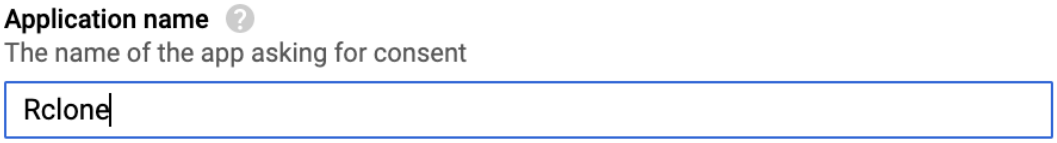


6. Select **Credentials** from the side menu and select OAuth client ID

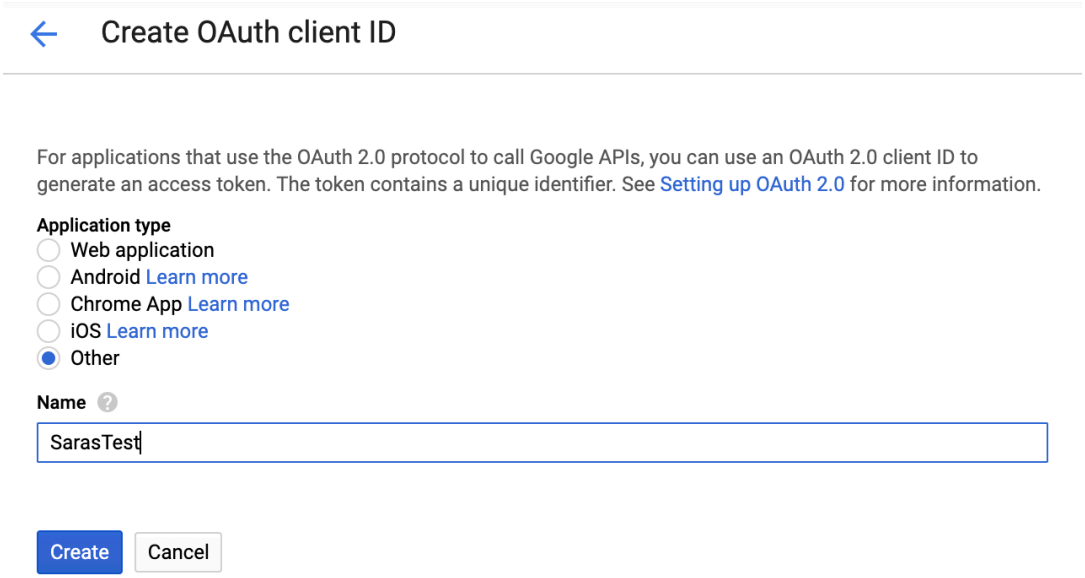


7. Click **Configure consent screen** and name your application something descriptive. The Application name is what will pop up when Google Drive asks for your consent when you try to connect to it using Rclone.





8. Give your OAuth client ID a name and click **Create**. Your Client ID and Secret will appear. You'll need to hold onto these for the next step in the Rclone setup



9. Your Client ID and Secret will appear. You'll need to hold onto these for the next step in the Rclone setup

If you decided to get your own Client ID, once you have it and your Client Secret, enter them to continue. If you've decided to use the public key, leave the entry blank and hit enter.

```
client_id> <Your Client ID>

Google Application Client Secret
Setting your own is recommended.
Enter a string value. Press Enter for the default ("").
client_secret> <Your Client Secret>

Scope that rclone should use when requesting access from drive.
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value
 1 / Full access all files, excluding Application Data Folder.
   \ "drive"
 2 / Read-only access to file metadata and file contents.
   \ "drive.readonly"
   / Access to files created by rclone only.
 3 | These are visible in the drive website.
   | File authorization is revoked when the user deauthorizes the app.
   \ "drive.file"
   / Allows read and write access to the Application Data folder.
 4 | This is not visible in the drive website.
   \ "drive.appfolder"
   / Allows read-only access to file metadata but
 5 | does not allow any access to read or download file content.
   \ "drive.metadata.readonly"
scope> <1-5>

ID of the root folder
Leave blank normally.
Fill in to access "Computers" folders. (see docs).
Enter a string value. Press Enter for the default ("").
root_folder_id>

Service Account Credentials JSON file path
Leave blank normally.
Needed only if you want use SA instead of interactive login.
Enter a string value. Press Enter for the default ("").
service_account_file>

Edit advanced config? (y/n)
y) Yes
n) No
y/n> n

Use auto config?
 * Say Y if not sure
 * Say N if you are working on a remote or headless machine
y) Yes
n) No
y/n> N # Use N if working on HPC, Y is fine for your PC
If your browser doesn't open automatically go to the following link: <longurl>
Log in and authorize rclone for access
Enter verification code> <verification code>

Configure this as a team drive?
y) Yes
n) No
y/n> n

--------------------
```

```
[GoogleDriveTest]
type = drive
client_id = <Your Client ID>
client_secret = <Your Client Secret>
scope = drive
token = {Info}
--------------------
y) Yes this is OK
e) Edit this remote
d) Delete this remote
y/e/d> y

Current remotes:

Name                 Type
====                 ====
GoogleDriveTest      drive
MyGoogleDrive        drive

e) Edit existing remote
n) New remote
d) Delete remote
r) Rename remote
c) Copy remote
s) Set configuration password
q) Quit config
e/n/d/r/c/s/q>
```

Multiple Google Drive connections can be established so you can connect to as many drives as you wish. You just need to go through the installation process for each new connection you create, though it isn't necessary to create multiple client IDs.

Part III

AWS

Chapter 1

Overview

AWS offers various data storage services, two of which are S3 and Glacier. S3 is intended for data that is in use and needs to be retrieved on an ongoing basis. Glacier is intended for archival data that doesn't need to be accessed frequently. Storage costs for S3 are higher than for Glacier, while retrieving data stored in S3 is less expensive than retrieving data stored in Glacier.

Chapter 2

Glacier

2.1 Pricing

Storage in AWS Glacier appears to be relatively straightforward: $0.01 per GB and doesn't appear to scale by the amount of data stored as S3 does. Pricing, however, does vary by geographical region. Retrieval pricing is where the cost gets tricky as retrieval time, retrieval size, the hosting server's location, and the number of requests all contribute to the cost. For a complete breakdown, see: aws.amazon.com/glacier/pricing. Below I've made an attempt to break it down somewhat.

2.1.1 Retrieval Pricing

Retrieval is not the same thing as downloading your archives. Before anything can be downloaded, you must first initiate a retrieval request. You have three options available for retrieving your data, listed below. These options determine how quickly your data are made available to you.

Retrieval Method	Time	Price/GB	Price/1,000 Requests
Expedited	1-5 minutes	$0.033	$11.00
Standard	3-5 hours	$0.011	$0.055
Bulk	5-12 hours	$0.00275	$0.0275

Figure 2.1: Listed pricing is for Northern California region.

Requests are calculated as the number of archives that are being retrieved. The total cost of a retrieval is calculated as:

$$\text{cost} = (\text{size of archives} \times \text{cost per archive}) + (\text{number of archives} \times \text{cost per request})$$

Chapter 3

S3

Currently, we have a Globus temporary subscription and an S3 trial bucket, so only benchmarking data have been determined for this combination. Users who want to use Glacier instead of S3 but have data in S3 can set up a protocol such that their archives are automatically transferred to Glacier after a certain period of time. Go to Configure bucket, select Vault Lifecycle to Glacier, set Lifecycle Rule.

3.1 Globus CLI

When using Globus CLI, you will be able to see all of your activity in the web portal, same as if you were initiating transfers from the website itself. You will also still receive emails for each completed transfer.

3.1.1 Benchmarking

I found a dramatic difference in upload speeds with the option "Checksum" in place as opposed to disabling this feature. In the Globus Web Interface, this can be seen as the option under **Transfer & Sync Options** as **Verify file integrity after transfer**. This is the default option and will always be switched on unless the user manually turns it off for each new transfer initiation. When using Globus CLI, the user will have to use the option

Using this option when transferring files to AWS S3 with Globus CLI dramatically improved transfer times.

When the checksum option is left in place, if you check your activity in the Globus web portal, you may notice that the total number of requested bytes have been transferred but the job is just hanging in the "transferring" state for up to several hours with no other noticeable progress being made. This is likely due to the checksum process.

Checksum in Place I was unable to complete all testing of the Globus transfers with the checksum in place due to the time limitations imposed by our Globus trial and the brutal transfer speeds. I do not recommend using Globus with the checksum in place for transferring large files. The download speeds without the checksum really aren't any better, but the upload speeds are at least more reasonable.

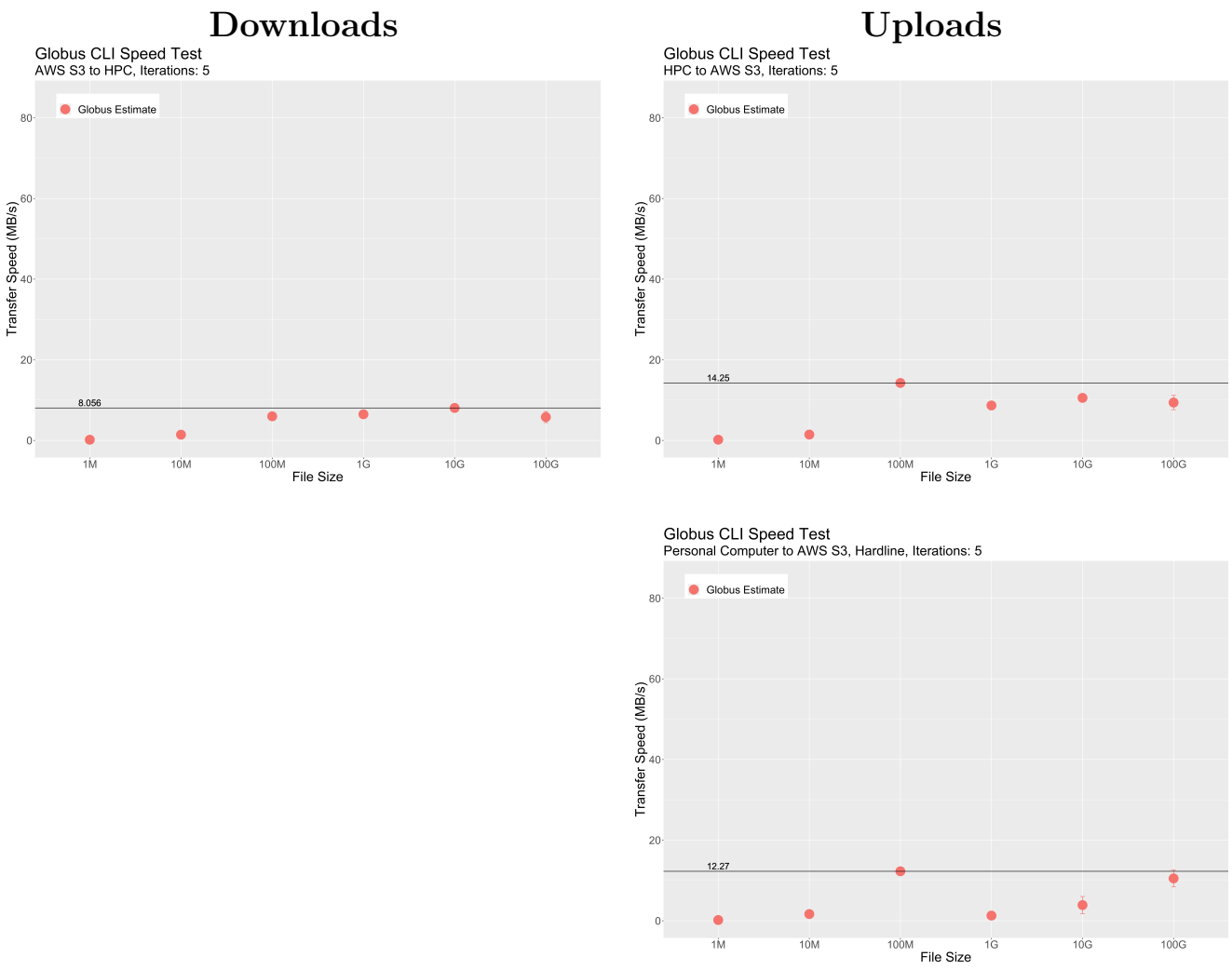


Figure 3.1

Checksum Disabled

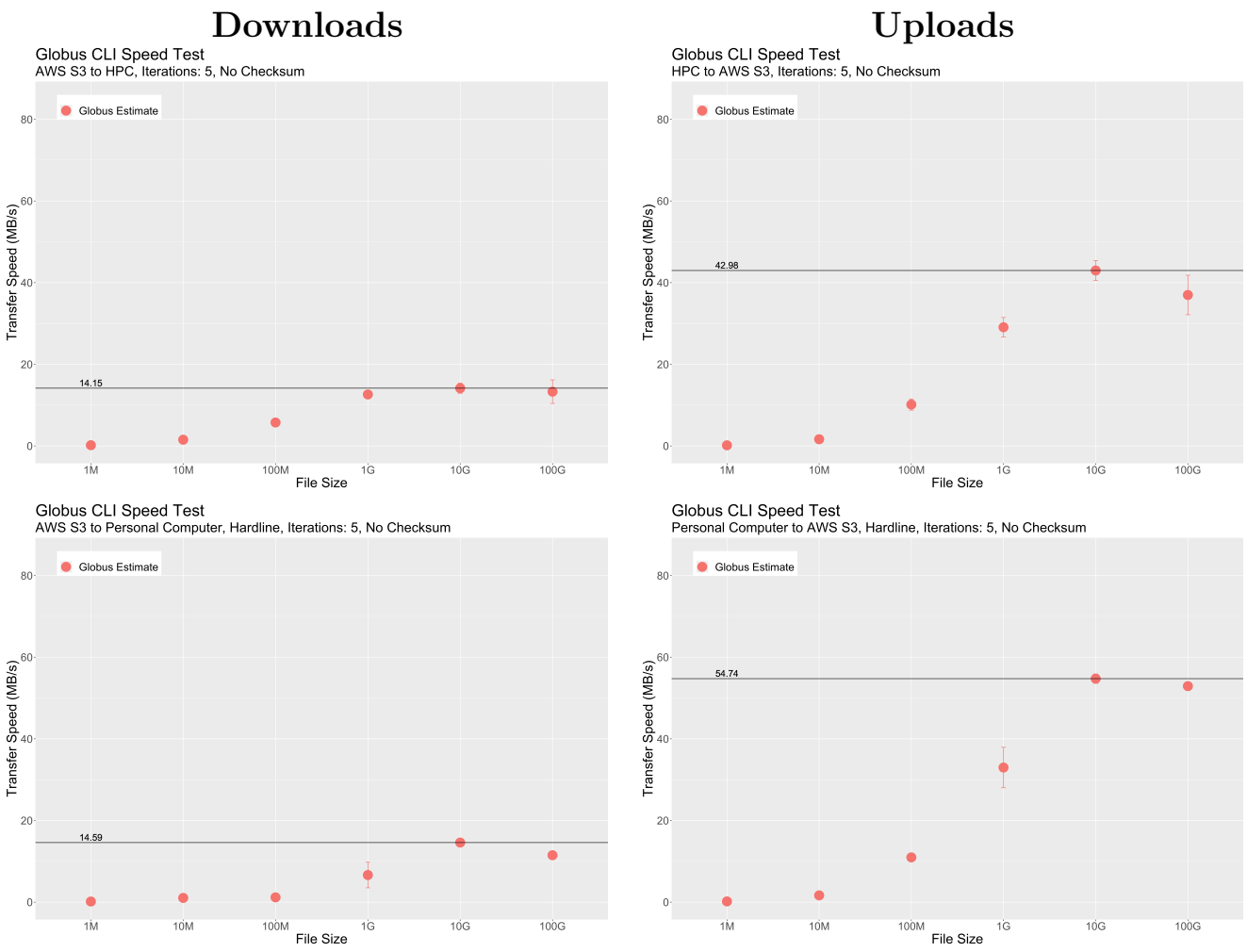


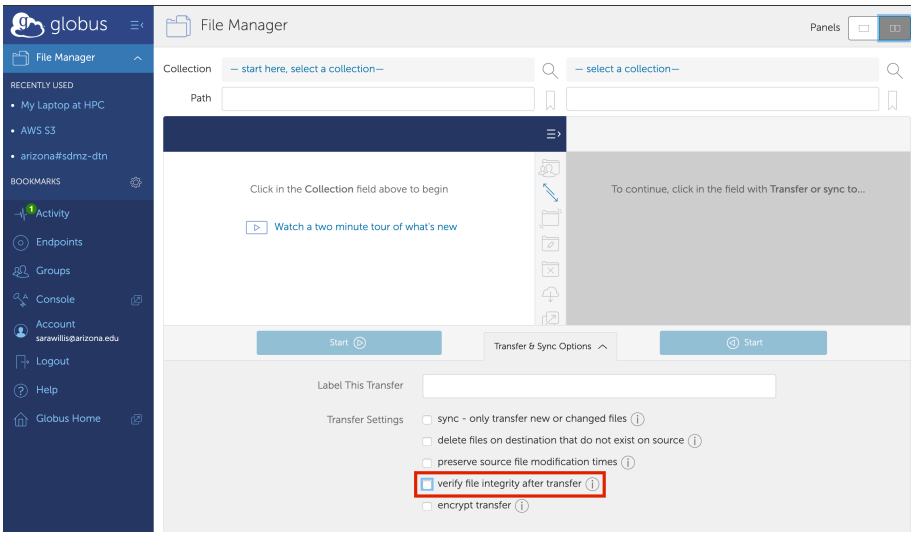
Figure 3.2

3.1.2 Installation, Setup, and Usage

Disabling Checksum

The option to verify file integrity post-transfer can be disabled both using Globus CLI as well as when using the web interface.

When using Globus online, open **File Manager**, select **Transfer & Sync Options** from the bottom of the screen, and deselect **verify file integrity after transfer**.



To disable Checksum using Globus CLI, use the option `--no-verify-checksum`, e.g.:

```
globus transfer --no-verify-checksum <source_path+filename> <destination_path+filename>
```

Installation

```
pip install --upgrade --user globus-cli
globus --help # Check that globus CLI has been installed
globus update # to update your version of the CLI to the latest
pip uninstall globus-cli # to remove CLI
```

```
dhcp-10-132-178-181:~ sarawillis$ globus endpoint search arizona#sdmz-dtn
ID                                   | Owner                                                          | Display
------------------------------------ | -------------------------------------------------------------- | -------
27cf226c-5402-11e6-824b-22000b97daec | tmerritt@arizona.edu                                           | arizona#
8c3f744e-baa8-11e9-9396-02ff96a5aa76 | 0df23eb0-3c61-4dd0-86ed-cec97c0f0aaf@clients.auth.globus.org | sdmz-dtn

dhcp-10-132-178-181:~ sarawillis$UAGlobus=27cf226c-5402-11e6-824b-22000b97daec
(base) dhcp-10-132-178-181:~ sarawillis$ globus endpoint show $UAGlobus
\Display Name:              arizona#sdmz-dtn
ID:                         27cf226c-5402-11e6-824b-22000b97daec
Owner:                      tmerritt@arizona.edu
Activated:                  False
Shareable:                  True
Department:                 None
Keywords:                   None
Endpoint Info Link:         None
Contact E-mail:             uits-hpc-team@list.arizonaedu
Organization:               University of Arizona
Department:                 None
Other Contact Info:         None
Visibility:                 True
Default Directory:          /~/
Force Encryption:           False
Managed Endpoint:           True
Subscription ID:            50762b6c-44e0-11e9-a618-0a54e005f950
```

```
Legacy Name:               u_ynojplx3nmi6llt4wmzmpoaqcq#sdmz-dtn
Local User Info Available: True
(base) dhcp-10-132-178-181:~ sarawillis$ globus ls $UAGlobus:\~/
CyberduckProfilingTest/
GdriveProfilingTest/
GlobusProfilingTest/
Intro_to_HPC/
ProgrammingSandbox/
PythonTests/
R/
RcloneProfilingTest/
Slurm_Hello_World/
ZippedExecutables/
bin/
include/
lib/
lib64/
miniconda3/
mpi_hello_world/
ondemand/
share/
InstallFSL.sh
Transfer.pbs
TransferToGoogleDocs.py
UploadTests_gdrive_2019-08-19 19:31:03.581258.csv
UploadTests_gdrive_2019-08-29 13:41:10.123468.csv
fslinstaller.py
osacc
restats
testInstall.r
```