

Tiered Storage Benchmarking

Data Transfer Options and Available Software

Sara Willis

November 27, 2019

Contents

I	Introduction	2
1	Overview	3
1.1	Testing Structure	3
1.2	Profiling Scripts Accessibility	3
1.3	Units	3
1.4	Warnings and Disclaimers	3
2	Installing Software on HPC	4
II	Google Drive	5
1	Google Drive Overview	6
1.1	Untested Software	6
2	Results	7
2.1	Best Performances	7
2.2	Software Rankings	7
2.3	Software Pros and Cons	9
3	Software: Tests, Installation, and Results	10
3.1	Globus Online Interface	10
3.2	Globus CLI - Permanent Endpoint	11
3.3	Cyberduck CLI	12
3.3.1	Testing and Results	12
3.4	Cyberduck GUI	14
3.5	Gdrive	16
3.6	RClone	17
3.6.1	Testing and Results	17
III	AWS	19
1	Overview	20
2	S3	21
2.1	Globus CLI	21
2.1.1	Benchmarking	21
IV	CyVerse	24
1	CyVerse Overview	25
1.1	iRods	25

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

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 apologize for in advance.
- When downloading files 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 merge 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.

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. Instructions on installing the requisite software will be available in a companion document.

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

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

This is being actively updated

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	Globus (No Checksum) 91 MB/s	Globus (With Checksum) 86 MB/s
	3	Rclone 65 MB/s	Globus (With Checksum) 81 MB/s	Globus (No Checksum) 83 MB/s
	4	Globus (No Checksum) 52 MB/s	Cyberduck GUI 77 MB/s	Cyberduck GUI 72 MB/s
	5	Globus (With Checksum) 52 MB/s	Rclone 68 MB/s	Rclone 66 MB/s
	6	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 (No Checksum) 52 MB/s	Globus (No Checksum) 95 MB/s	Globus (No Checksum) 101 MB/s
	5	Cyberduck CLI 38 MB/s	Cyberduck CLI 35 MB/s	Cyberduck GUI 88 MB/s
	6	Globus (With Checksum) 35 MB/s	Globus (With Checksum) 33 MB/s	Globus (With Checksum) 40 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
	6			
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
	6			

2.3 Software Pros and Cons

Pros	Cons
Globus	
<ul style="list-style-type: none"> ◦ User-friendly interface ◦ Sends email confirmations when file transfers complete ◦ Versatility with both a CLI and Web version 	<ul style="list-style-type: none"> ◦ Slow upload speeds
Cyberduck CLI	
<ul style="list-style-type: none"> ◦ Excellent download speeds for large files to HPC ◦ Reasonable upload speeds 	<ul style="list-style-type: none"> ◦ Consistently slow download speeds for anything under 100 GB
Cyberduck GUI	
<ul style="list-style-type: none"> ◦ Convenient drag-and-drop interface ◦ Relatively easy setup ◦ Fastest upload speeds 	<ul style="list-style-type: none"> ◦ 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	
<ul style="list-style-type: none"> ◦ 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 	<ul style="list-style-type: none"> ◦ 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	
<ul style="list-style-type: none"> ◦ 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 modifications 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 speed things up, however, if some file corruption takes place during the process Globus will not retry the transfer.

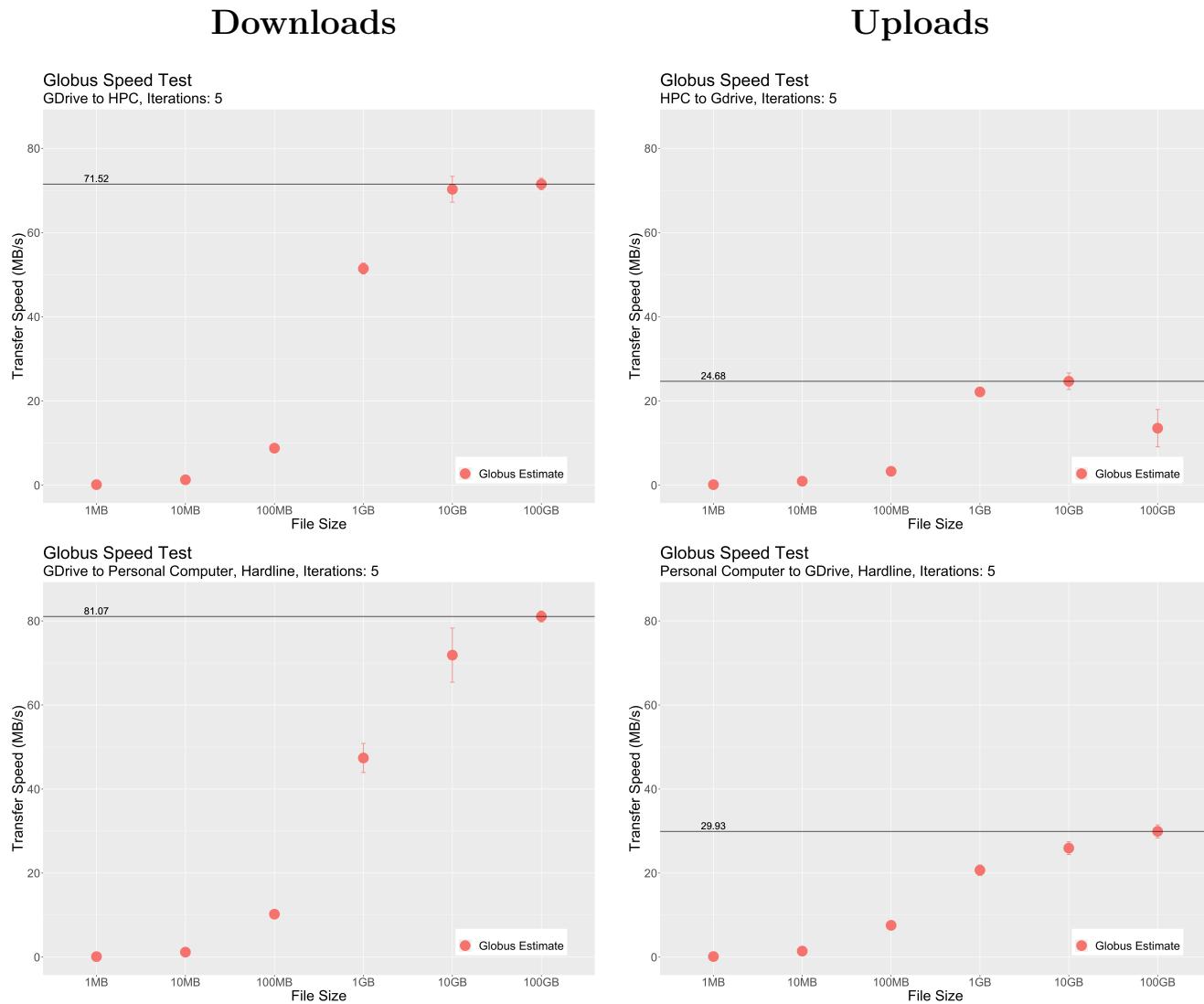


Figure 3.1: Download speeds from Gdrive to both my PC and to HPC were markedly faster than Upload speeds going the other direction. The maximum mean profiled speed is plotted as a horizontal line.

3.2 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.

While transferring some of the 100GB files from HPC to Google Drive, I received 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 and the transfer continued as normal.

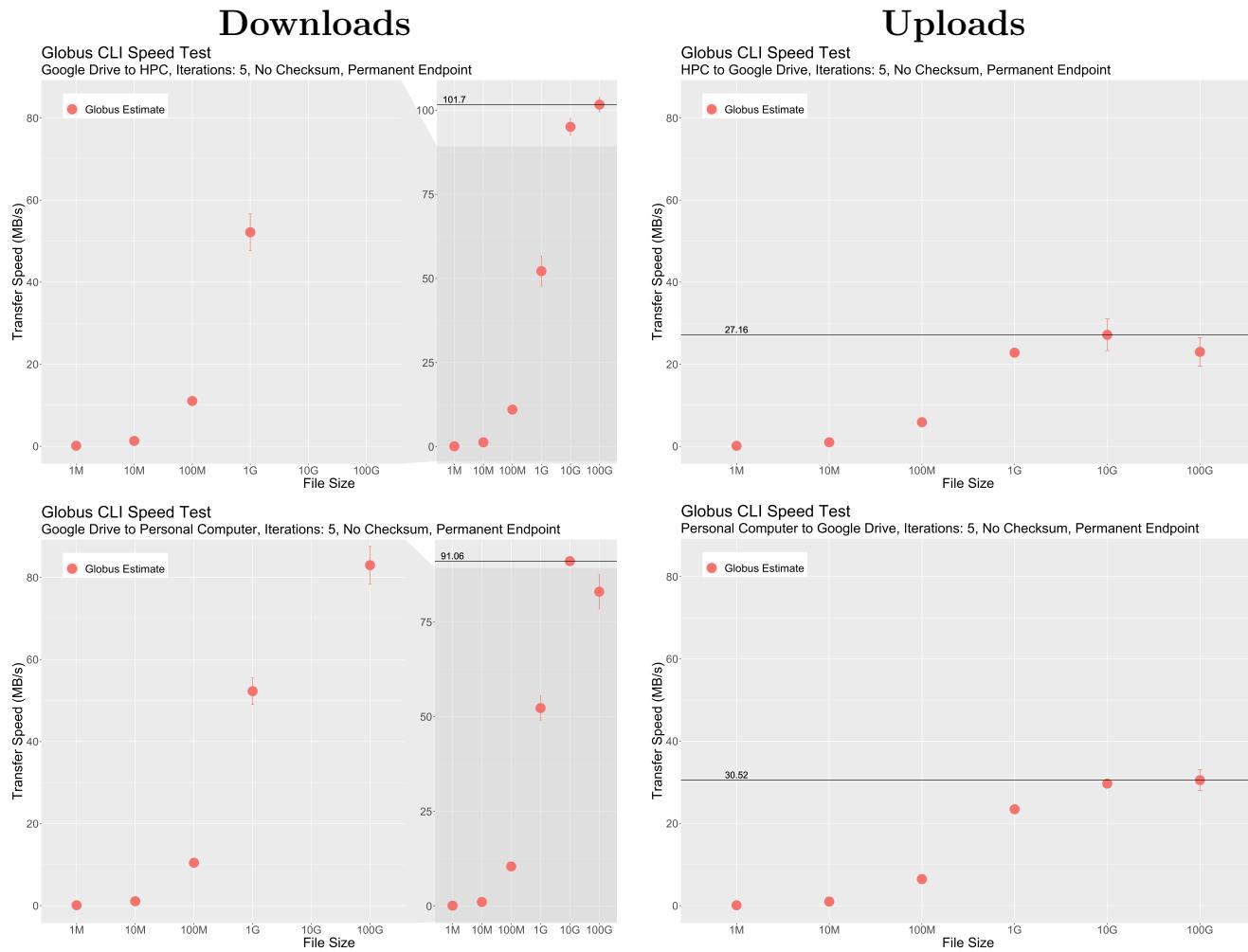


Figure 3.2: Upload speeds continued to be so-so following the implementation of the permanent endpoint, but the download speeds did pick up.

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 file size 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.

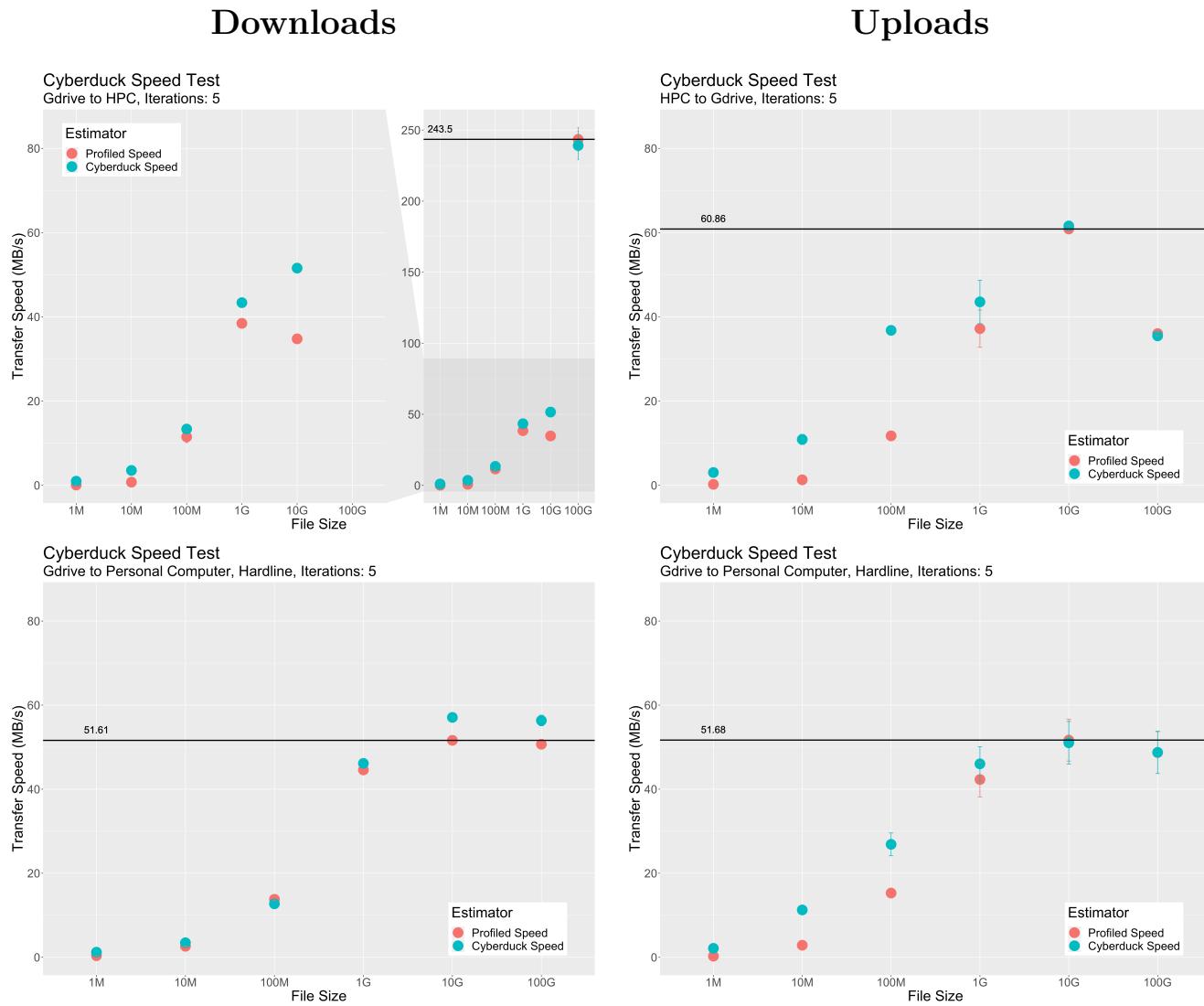


Figure 3.3: For the most part, the results were fairly uniform with download speeds being slightly higher than upload speeds. There was an interesting effect hitting the 100GB range, however, when transferring files off of Google Drive to the filexfer node. Three independent tests confirmed the transfer 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.

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 replace isn't immediately overwritten. Cyberduck creates a folder where it downloads your file in chunks and once all the chunks have been downloaded, they get concatenated into the final product. This means that if you're low on space where you're saving your file, you may run exceed your storage quota and the transfer will fail.

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 available to run profiling tests. Instead, I manually initiated each transfer by double-clicking them in the Cyberduck window 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 programs for Google Drive Uploads with speeds approaching 80 MB/s. This is substantially more than other programs profiled here and makes it an asset in moving large quantities of data.

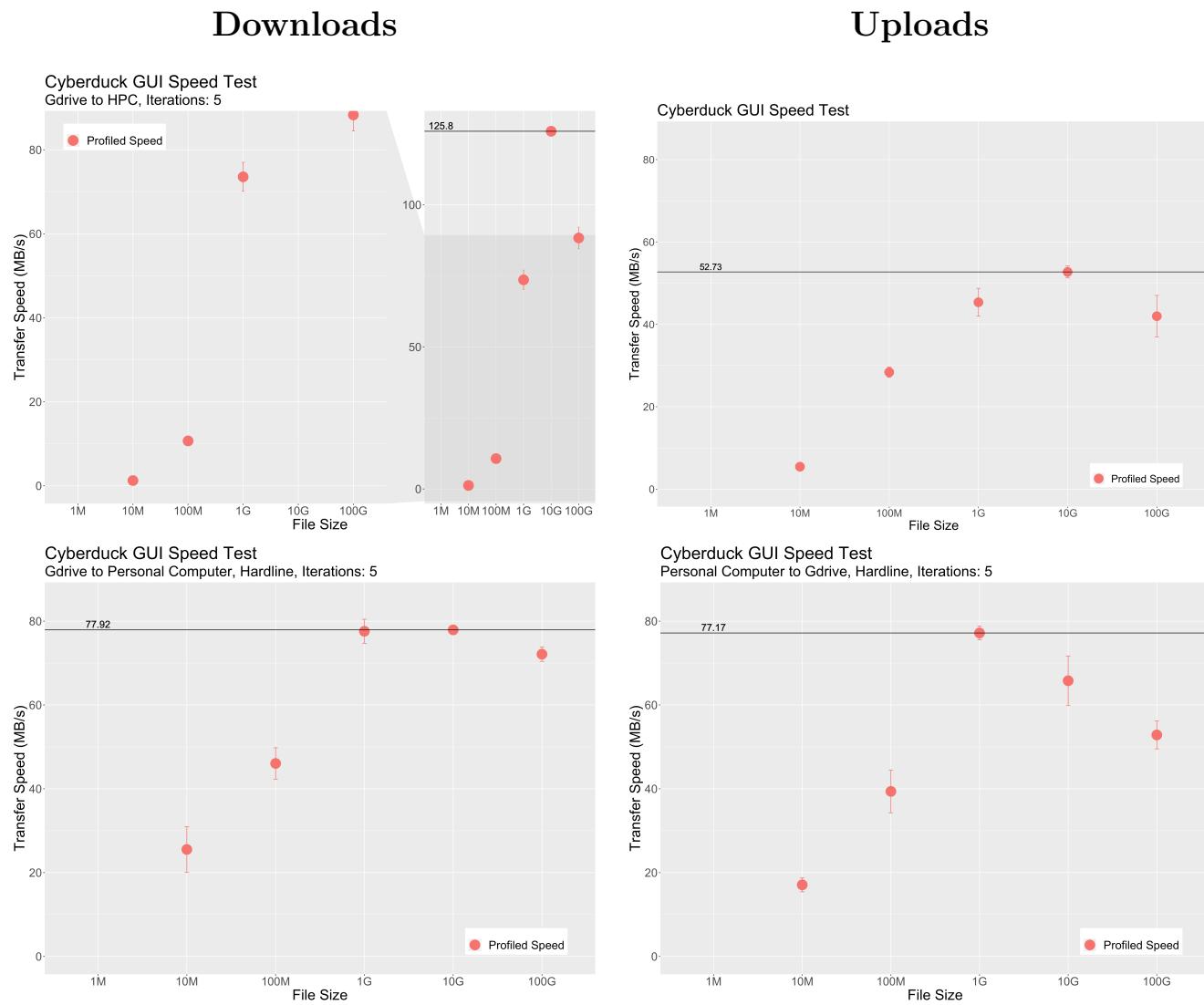


Figure 3.4

3.5 Gdrive

Gdrive is another command-line interface program built to transfer files to/from 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 existing documents but will add an additional file with the same name. To download or delete a file, however, you cannot specify a name, you must use the file ID. This can be accessed using the command `gdrive list`. Used in conjunction with `| grep <filename>`, you can easily pull the ID.

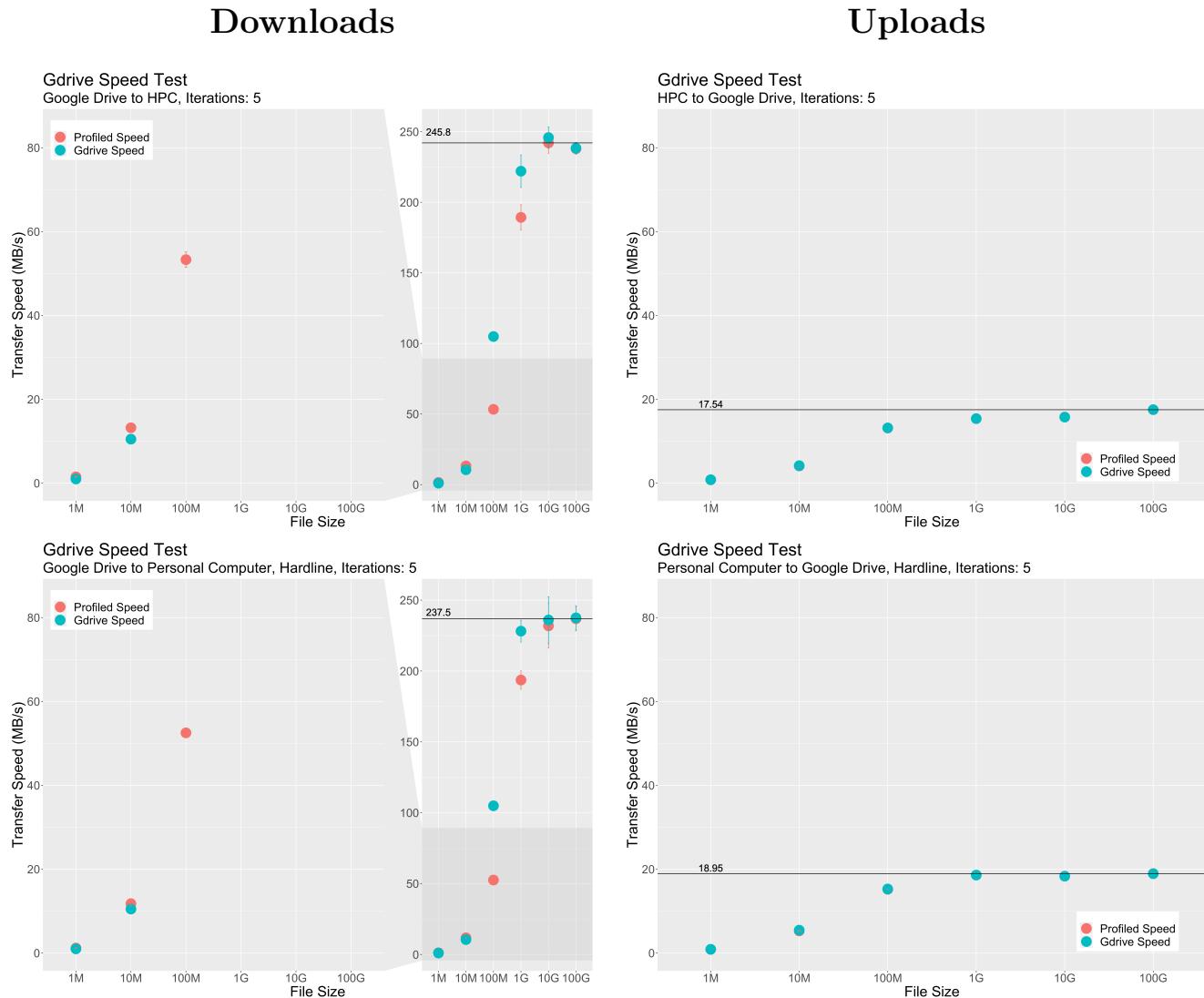


Figure 3.5: Gdrive had some of the best download speeds of any of the programs tested. There was some discrepancy between the profiled speeds and the gdrive-predicted speeds for downloads. This is likely due to how rapidly the files were transferred. Gdrive seems to have a lower bound for estimates of transfer time which I assume to be 1 second, so if a file takes less time than that, it estimates 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

3.6 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. This option should be used with caution since giant files transferred without chunking may beat up your machine's memory.

Important 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

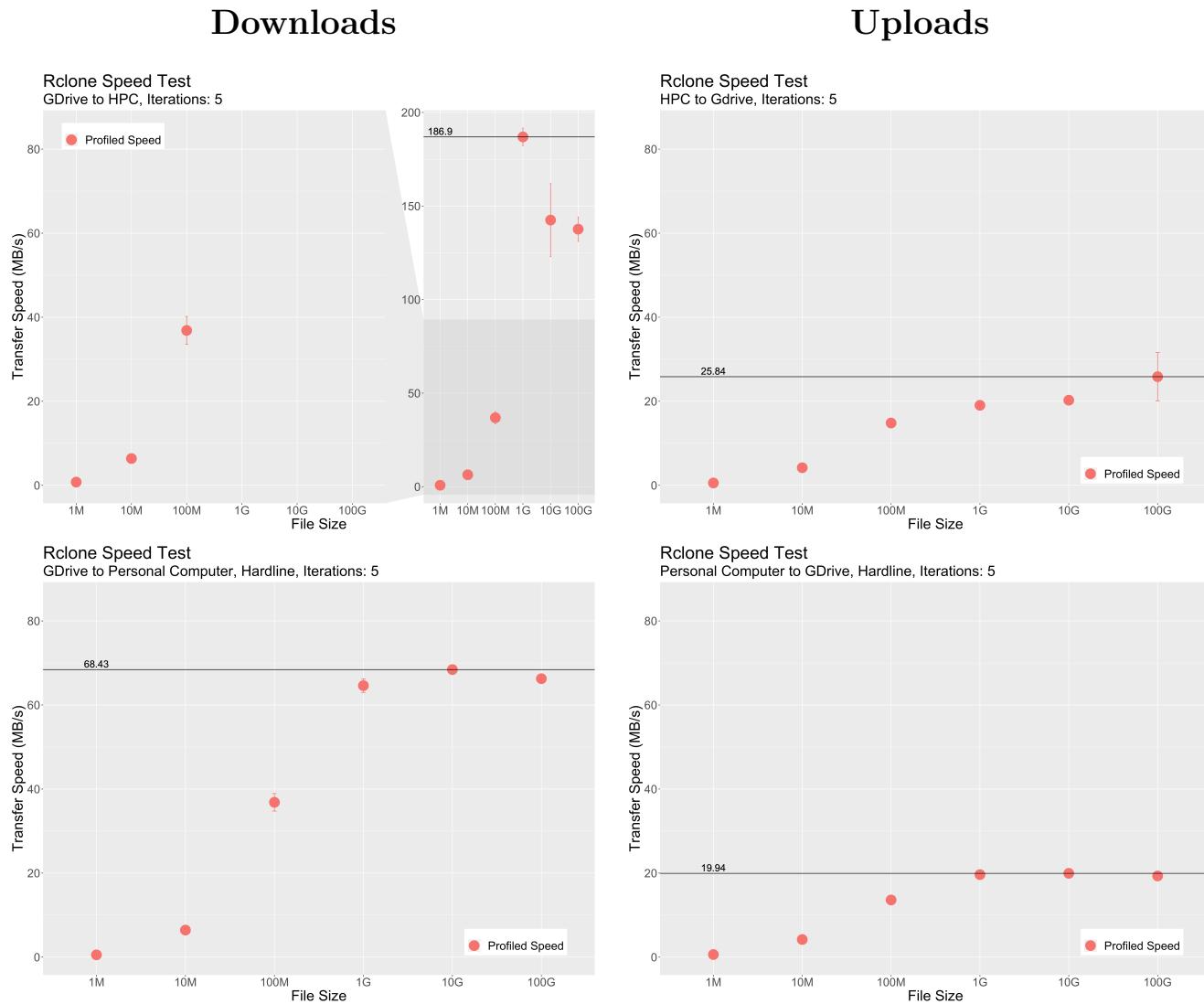


Figure 3.6: Rclone upload speeds are not particularly good when no flags are specified. Later tests that altered the upload-specific chunk size improved speeds. Examples will be provided below. Download speeds from Google Drive to HPC were very good, approaching those that were given by Cyberduck.

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.

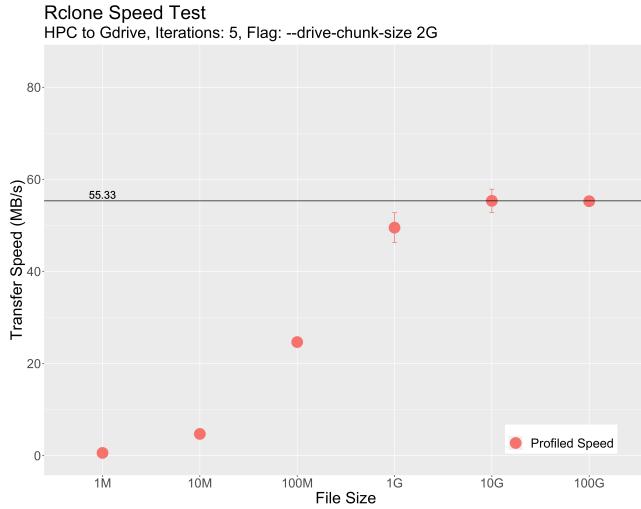


Figure 3.7: Rclone upload speeds improved when flags were specified. Clockwise from upper left:
1. `--drive-chunk-size 2G` : increased the chunk size of the file upload to 2GB/chunk which roughly doubled the upload speed. Further tests should be done to determine whether there is an optimal chunk size.

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.

Currently, the only transfer program that we have tested with AWS is Globus which does not have a connection to AWS Glacier available. To use Glacier, users will need to upload to an S3 bucket and then migrate it.

Chapter 2

S3

Currently, we have a Globus temporary subscription and an S3 trial bucket, so test were performed while this trial was active. Due to the limited trial, not all tests were completed in the timeframe. The complete tests available are listed in the section below.

Users who want to use Glacier 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, and set a Lifecycle Rule.

2.1 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.

2.1.1 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 `--no-verify-checksum`.

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 due to the checksum process (confirmed by Globus at SC19). According to Globus representatives, selecting the checksum option can double the transfer time.

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 least more reasonable.

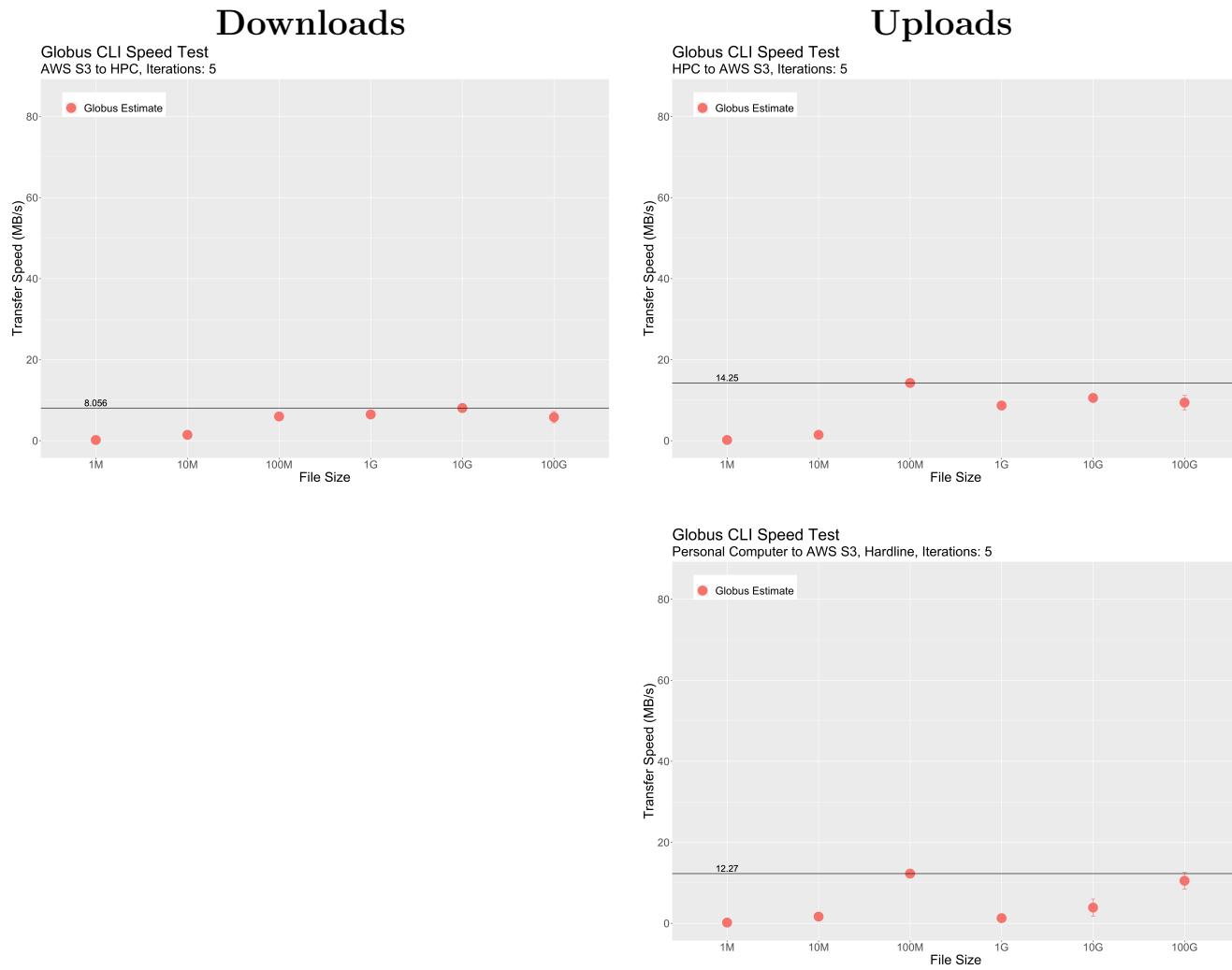


Figure 2.1

Checksum Disabled

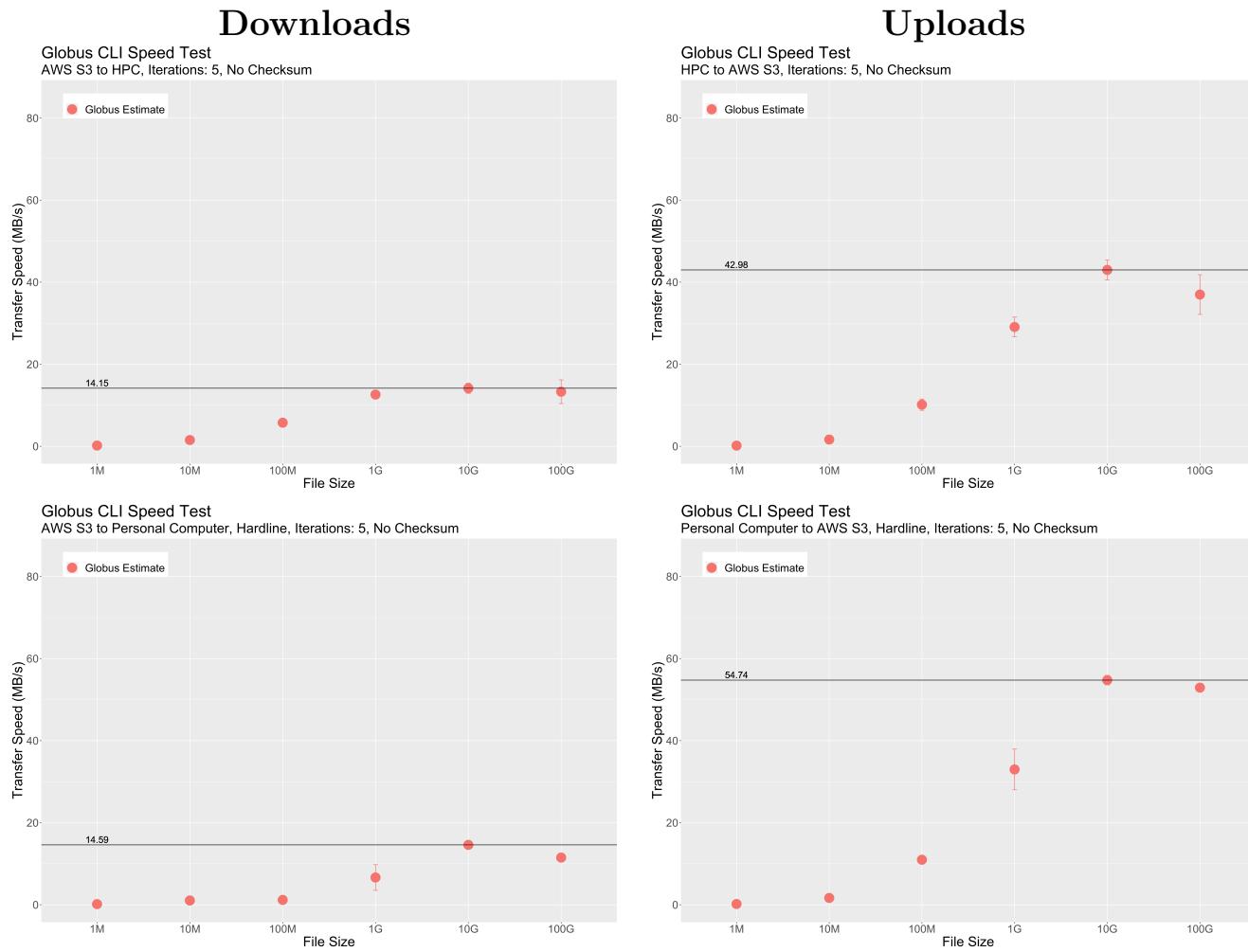


Figure 2.2

Part IV

CyVerse

Chapter 1

CyVerse Overview

1.1 iRods

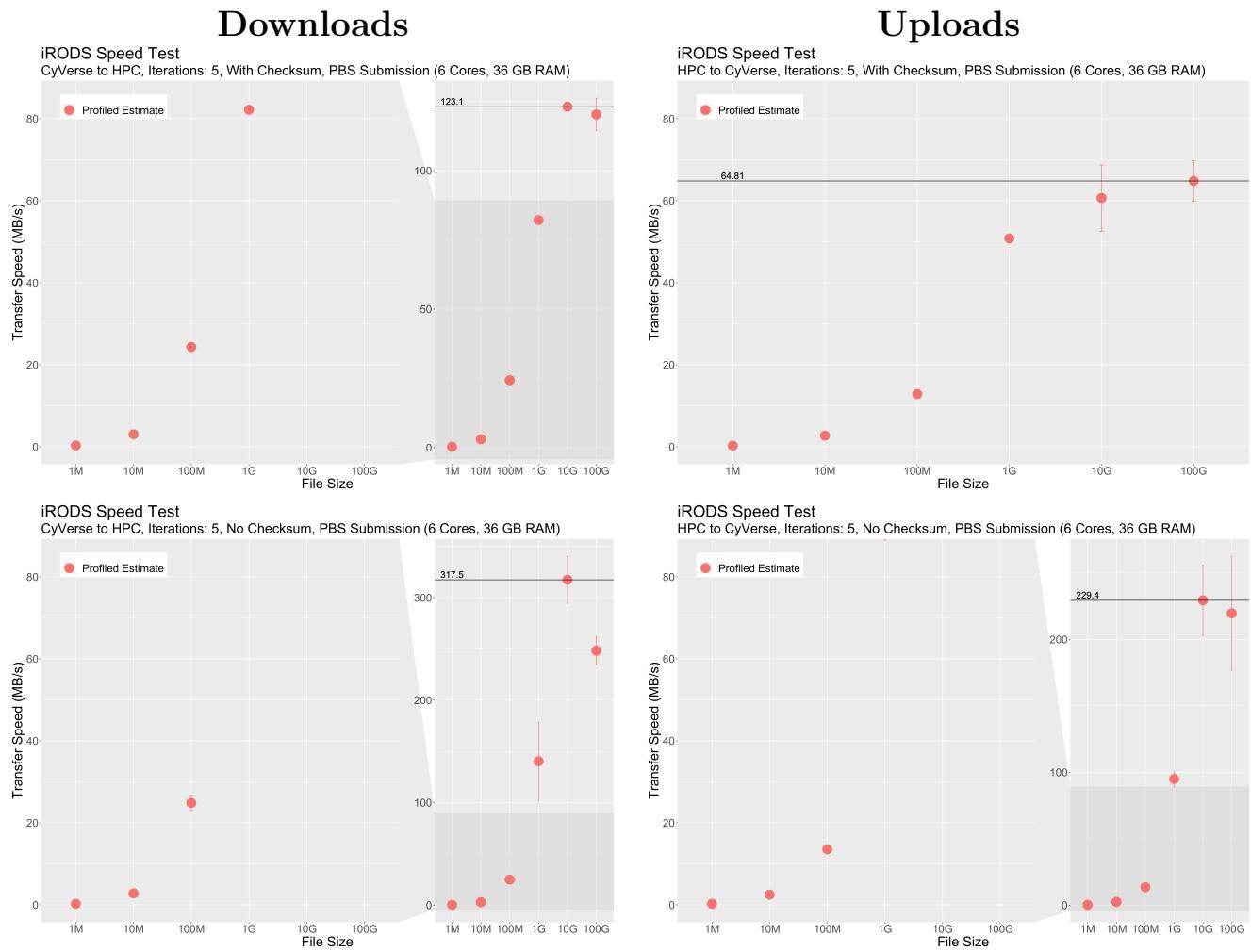


Figure 1.1