universität innsbruck

DPS Research Seminar
Winter Term 2022/23

File Transfer Performance of Cloud Providers

Manuel Meitinger, BSc

Abstract

Scheduling in the cloud focuses primarily on costs and compute resources. It is often assumed that within the same region, file transfers perform equally well between intra-provider endpoints. Inter-region transfers are usually modelled in a content-agnostic way and assuming symmetric bandwidths. This seminar paper will analyze multiple cloud providers, regions and services to uncover differences in file transfer performance, that can aid future scheduling decisions.

Contents

1 Introduction

The possible benefits of multi-cloud approaches have long been realised,[1] and over the last decade, public cloud providers like Amazon Web Services, Microsoft Azure and Google Cloud Platform continuously improved various service models.[2] A more recent and fast growing model, serverless computing, alleviates the burden of managing VMs and software stacks, and instead allows to focus on scheduling.[3] However, research regarding scheduling predominantly gravitates towards compute-intense tasks and often neglects storage transfer performance.[4;5]
In single-cloud environments, scheduling also tends to either focus on intra-region transfers between VM hosts and storage,[6;7] or assumes symmetric inter-region bandwidth.[8]
The contribution of this work is to provide (i) insights in multi-cloud, multi-region transfer performance and capabilities that can aid scheduling, (ii) a test environment that allows to reproduce the results as well as rapid development and execution of additional tests, and (iii) 162k publicly available data points measuring the performance across 77 regions and three providers.

1.1 Related Work

While existing measurement data is hard to find, Imran et al. addressed scheduling with explicit focus on transfer performance in *OneDataShare*.[9] Their testbed and approach supports multi-cloud in theory, yet they only conducted tests on Amazon Web Services. Also, unlike more cloud-oriented services like Function as a Service (FaaS), where scheduling decides which function in which region to execute,[10] the scheduling services considered in *OneDataShare* operate on transfer parameters, like compression, buffer size or transmission protocol. In our work we look at compression as well, but otherwise assume HTTPS for all transfers.
Serhiienko and Spillner looked at the performance of multi-cloud management platforms,[11] where they also measured different download and upload times of files ranging from bytes to megabytes in size, using a recomputable method. In contrast to our work, they performed no inter-cloud or inter-region transfers, as they compared multi-cloud-targeting libraries within a single cloud provider and region.

2 File Transfer and Storage Capabilities

Before we evaluate the performance of each cloud provider, we need to understand what kind of transfers we are actually able to measure, where endpoints reside, and what storage capabilities are available. Also, a file transfer endpoint isn't just a bucket or container (or more precise: the storage service), but other file-based services as well. Performing optical character recognition (OCR) in the cloud, for example, requires a file to be transferred to that service.[12;13;14]
Nonetheless, let's start by covering the storage services first.

2.1 Folders and Files

Each of the three cloud providers have their own storage solution for files. Files are also called objects on Amazon Web Services[15] and Google Cloud Platform[16] or blobs on Microsoft Azure[17]. These files are stored within buckets (Amazon Web Services, Google Cloud Platform) or containers (Microsoft Azure), at a specific geographical region (Amazon Web Services, Google Cloud Platform) or location (Microsoft Azure). All three storage services support role-based and policy-based access control on bucket (or container) level.[18;19;20]. It should be noted that objects in buckets (or blobs in containers) follow a flat structure, but all providers support a logical folder hierarchy.[21;22] Different storage classes (also called access tiers) are available, influencing availability by redundancy and retrieval time.[23;24;25]
For our tests we opted to use a per-bucket (per-container) public access policy and a hot (or standard) storage class. Using a public access policy allows all files to be downloaded using

a publicly accessible URL, which has the advantage of being able to directly measure transfer times, without having any authorization or storage API overhead. Using hot storage keeps the retrieval latency at a minimum, thus also favoring pure transfer times.
In addition, no multi-region redundancy has been used, and - where available - also no multi-zone redundancy, in order to precisely limit the origin of the transfer.

2.2 Endpoints and Regions

Up to now, we've only talked about publicly accessible URLs for files, but not how they work on the different providers, i.e. what the storage service's endpoint looks like.

Amazon Web Services: `https://<bucket>.<region>.amazonaws.com/<object>`[26]
The region is part of the endpoint's host name, which also resolves to an IP address in that particular region.

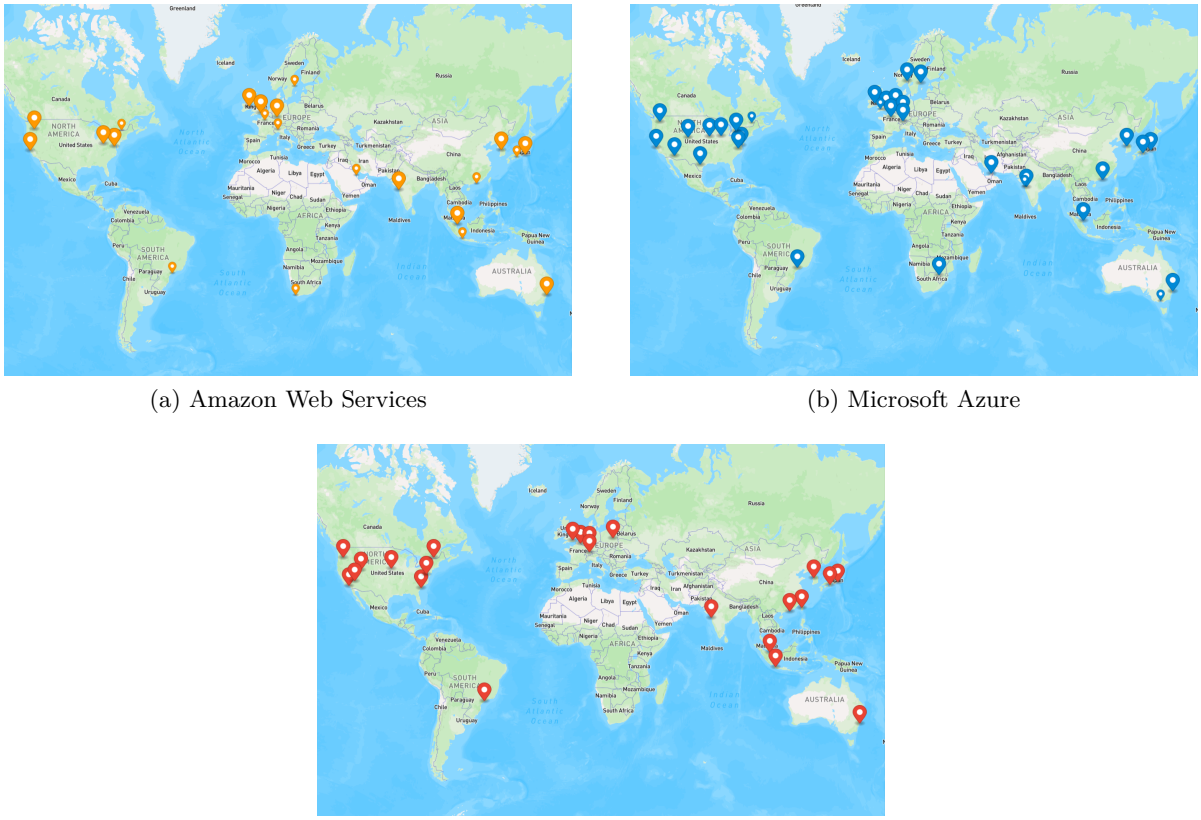
Microsoft Azure: `https://<storageaccount>.blob.core.windows.net/<container>/<blob>`[27]
The last subdomain is the name of the container's *storage account*, which is a collection of containers (and other storage solutions), and is also bound to a specific region.[28] When a client resolves the storage account endpoint's host name, it gets a `CNAME` entry to another host that resides in that region and has an IP address in that particular region as well.

Google Cloud Platform: `https://storage.googleapis.com/<bucket>/<object>`[29]
Unlike Amazon Web Services and Microsoft Azure, Google Cloud Platform uses *anycast* IP addresses.[30] So from outside of Google's network, it is not possible to deduce the region of the bucket by tools like IP geolocation, as the returned IP address will be routed by BGP to the nearest data center, and then internally by Google.
However, when `storage.googleapis.com` is resolved within Google Cloud Platform, say from a Cloud Function, a `CNAME` to an internal storage handling server is returned. So for example in region *europe-west3* (which is located in Frankfurt) the host name `fra24s05-in-x10.1e100.net` is returned, which has a region-bound IP address instead of an anycast one. As these host names can be accessed from anywhere within the Google Cloud Platform, it is possible to 'enforce' a specific endpoint region. Also, IP geolocation is again possible.

Figure 1 was composed using geolocation tools and publicly available information.[31;32;33] It depicts all considered regions and locations.

2.3 APIs and SDKs

In section 2.1 we alluded to an overhead that occurs when using the storage API for file transfer. This also holds for other services, like OCR. Unlike the storage service's API, invoking the OCR service's API cannot be circumvented by publicly available links. As a matter of fact, we cannot even measure transfer times directly at all, rather we regard the service's execution time of the service. For this to work, we need to craft special images (see section 3.3 for more detail) and also call the *REST* API directly.
These REST APIs are the 'closest' one can get to the provider, but they require more arguments than needed for performing the service's task itself. In Microsoft Azure, OCR for example requires a subscription key,[13] Google Cloud Platform needs an authorization token,[14] and Amazon Web Services requires intricate signing altogether.[12] That's why all three providers offer software developer kits (SDKs) for various programming languages, that handle authorization, request signing, target endpoint selection, and so on, before calling the actual REST API.[34;35;36]
Behind the scene, these SDKs therefore invoke additional APIs, which add to the total execution time, and especially to latency, rendering SDKs unsuitable for our needs. Moreover, we want to abstract over all programming languages and SDKs, and not compare SDK performance.



(a) Amazon Web Services

(b) Microsoft Azure

(c) Google Cloud Platform

Figure 1: All regions providing FaaS, small pins indicate no OCR support.
(Map data copyrighted OpenStreetMap contributors.)

The solution was to implement the functionality otherwise provided by the SDKs ourselves, but *outside* of the (timed) REST API invocation.

2.4 Services and OCR

So far we've always talked about OCR as an example for a service that represents a file transfer endpoint, but it's actually *the* service we picked for measuring performance (alongside each provider's actual file storage service). The reason being that it's the only service where all of the following conditions hold:

Available on all three providers. [12;13;14] Each provider has its own suite of file related services, from analytics,[37] synchronization,[38] migration and even transfer.[39] None of them, however, are comparable across all providers.

Possibility of restricting the service to a certain region or location. Some of the aforementioned services operate 'anywhere' in the cloud, or at least outside of the invoker's control. With OCR, the region or location can be restricted using specific endpoints:

- `rekognition.<region>.amazonaws.com` on Amazon Web Services,[40]
- `<location>.api.cognitive.microsoft.com` on Microsoft Azure[41] and
- `<continent>-vision.googleapis.com` on Google Cloud Platform.[42]

Admittedly, on Google Cloud Platform it's a very coarse restriction, but as we've seen in section 2.2, Google Cloud Platform is known for doing routing internally and not offering region-specific endpoints. In fact, when `vision.googleapis.com` is called from a

Google Cloud Function, it resolves to a host within that region, in the same manner as `storage.googleapis.com` does (see section 2.2 for details).

Allowing non-stream input. Translation services would fulfil all previously mentioned requirements, but their input cannot be a reference to a file. Instead, a string of characters has to be provided,[43;44;45] which means we would measure transfer speeds between the invoker and the service, not any file transfer. All three OCR services on the other hand, offer either a bucket and object reference,[12] or an arbitrary URL as input.[13;14]

Possibility of crafting suitable input files. Another time we don't want to measure is the execution time of the service itself, so we don't want OCR to actually be performed. The service therefore has to reject the input file, but only after it has been *fully* read. With a specially crafted image (see section 3.3), that's possible using the OCR service.

Synchronous invocation Speech recognition is another service contender that would comply with our list of requirements so far.* While Microsoft Azure supports synchronous service invocation for short audio files,[46] Amazon Web Services unfortunately only support asynchronous (job-based) invocations.[47] These require multiple API calls and depend on internal scheduling, controlled entirely by the provider, thus not suitable for measuring file transfer times.

Initially, measuring transfer times *across* regions to the OCR service was also considered. But since Amazon Web Services doesn't allow this at all (the region of the referenced bucket must match the region of the service)[48] and Google Cloud Platform only supports continent-level endpoints,[42] this plan was scraped, as only Microsoft Azure supports fine-grained location restriction and arbitrary input URLs.[13;41]

2.5 File Operation Capabilities

To conclude the capabilities sections, we have a look at 'traditional' file operations that providers support. We start with *delete* operations (table 1). All providers support retention policies that

Provider	Retention	Restore Bucket/Container	Restore Object/Blob
Amazon Web Services	yes[49]	yes[50] (Amazon Backup)	yes[50;51] (Amazon Backup or if versioned)
Microsoft Azure	yes[52]	yes (from snapshot[53])	yes[54] (from snapshot)
Google Cloud Platform	yes[55]	not documented	yes[56] (if versioned)

Table 1: Delete operation capabilities.

prevent accidental deletion,[49;52;55] as well as restoring versioned objects or blobs.[51;54;56] Amazon Web Services has a separate backup solution that can also restore buckets,[50] Microsoft Azure uses the same mechanism as it does with blobs.[53]
Next we look at *copying* existing cloud objects or blobs (table 2). Every provider offers APIs to copy objects or blobs across regions,[57;58;59]. Some of these APIs have a synchronous and asynchronous variant, where some restrictions apply to the former.[57;60;61] In both cases, only entire objects and blobs can be copied, not parts.[57;58;59]
Finally, objects and blobs can be up uploaded in parts or be *constructed* from existing objects.

*ID3v2 tags could be used to generate invalid MP3 files with valid prefixed payload.

Provider	Across Regions	Synchronous	Partial File
Amazon Web Services	yes[57] (with restrictions)	yes[57]	no[57]
Microsoft Azure	yes[58]	yes[60] (256MB limit)	no[58;60]
Google Cloud Platform	yes[61]	yes[59;61] (same region or small objects)	no[61;59]

Table 2: Copy operation capabilities.

Amazon Web Services supports uploading multiple parts of a new object in parallel, and even using (partial) content of existing objects (if those objects are stored in a S3 bucket).[62;63] Microsoft Azure goes one step beyond and also supports (partial) external content, where Microsoft Azure will download the content from an publicly accessible URL.[64;65]
The only provider that cannot use existing content in multipart uploads is Google Cloud Platform,[66] but - using resumeable uploads - it supports smaller chunks than Amazon Web Services at the cost of parallel upload.[67] Google Cloud Platform can, however, combine entire existing objects into a new one.[68]
For a more concise summary see table 3.

Provider	Concept	From Existing Objects/ Blocks	From External URL	Partial Content	Parallel Part Upload
Amazon Web Services	Multipart Uploads[62]	yes[63]	no[69;63]	yes[63] (min. 5MB chunks)	yes[62]
Microsoft Azure	Block or Page Blob[70]	yes[64;65]	yes[64;65]	yes (blocks[64] or pages[65])	yes[71]
Google Cloud Platform	Multipart Uploads[66]	no[72]	no[72]	yes (min. 5MB chunks[73])	yes[66]
	Resumable Uploads[74]	no[67]	no[67]	yes (multiple of 256KB[67])	no[67]
	Combine Objects[68]	yes[68]	no[68]	no[68]	yes[68] (objects as parts)

Table 3: Construct (from parts) operation capabilities.

3 Test Environment

Equipped with the knowledge about transfer capabilities and the challenges of measuring transfer times, we now explore the practical solutions to these problems in our *Cloud Workbench*.[75] We will also provide information on how to setup the environment, and how to extend it.

3.1 Design Goals

In general, the design goals behind the test environment where flexibility, extendibility and economics. Let's start with the last one, as it explains the architecture from the bottom up.

3.1.1 Economics

Since we want to measure transfer times across the globe and between all three providers, we need to able to execute code in every region of every provider. Traditionally, this would require almost 100 virtual machines,[31;32;33] which entails the time and cost of deploying, running and managing VMs, as well as implementing some sort of job dispatching. That's why we opted for a serverless solution using functions as a service (FaaS).[10] Not only does it allow us to directly invoke test code, each provider also offers millions of free invocation each month.[76;77;78]

3.1.2 Flexibility

While using FaaS takes care of the costs and managing VMs, deploying the functions is still a time consuming process. Redeploying functions is usually a bit faster, but the initial deployment can take several minutes.[79] This doesn't allow for much flexibility as far as code changes are concerned, especially when you have to deploy to hundreds of regions on different providers.
Our solution to this problem is using a single JavaScript (NodeJS) function, that is able to dynamically execute submitted code.[80;81;82] Not only is redeployment time reduced to zero, it also allows for shared code between providers. As figure 2 shows, most of the function itself is common across all three providers as well. What differs is how errors are signalled to the

```
try {
  if (typeof request !== 'object')
    return badRequest('Body must be a JSON object.')
  if (typeof request.time !== 'number')
    return badRequest('Time must be a number.')
  if (Math.abs(Date.now() - request.time) > 60000)
    return badRequest('Time difference is too big.')
  if (typeof request.command !== 'string')
    return badRequest('Command must be a string.')
  if (typeof request.hash !== 'string')
    return badRequest('Hash must be a string.')
  if (typeof key !== 'string' || key.length < 16)
    return internalServerError('Invalid key.')
  const { createHmac } = await import('node:crypto')
  const hmac = createHmac('sha256', key)
  hmac.update(`<${request.time}|${request.command}>`)
  if (hmac.digest('hex') !== request.hash)
    return forbidden('Hash mismatch.')
  return success(await AsyncFunction(request.command).call(null))
} catch (e) {
  return internalServerError(String(e))
}
```

Figure 2: Common code of FaaS functions.

executing runtime (`badRequest`, `forbidden`, etc.) and the way the result is returned upon `success`.
To ensure that commands cannot be executed by anyone, rather than relying on a simple app

key that gets appended to each query,[83] we employ HMAC using a secret key, that never gets transmitted. Both command and current time get hashed, so even if a request would somehow get intercepted, it can only be repeated in a very short time frame. An adversary can, however, never alter the command without invaliding the HMAC.[84]

3.1.3 Extendibility

The last goal was to ensure that the test environment can be extended without having to deal with any of the aforementioned internals. Also, it should be easy to use it in a scripting language, both of which is achieved by a PowerShell file that can be *dot-included* by running `. ./invoke.ps1`.[85] After doing so, getting for example the NodeJS version for all regions in all providers is as simple as invoking

```
$All | Invoke-CloudFunction -Command "return process.version" -Raw
```

where `$All` and `Invoke-CloudFunction` are provided by the script. Instead of `$All` for all providers, `$Azure`, `$AWS` and `$GCP` can be used as well. Or, to speed things up, execute `Invoke-CloudFunction` in parallel:

```
$All | % -ThrottleLimit 20 -Parallel {
  . ./invoke.ps1
  $_ | Invoke-CloudFunction -Command "return process.version" -Raw
}
```

Since the command is just JavaScript, it can be dynamically generated by using string variables. The script for example exports `$GCP_StorageBucket`, which creates a Google Cloud Platform storage object and selects the bucket in the region of the function. Checking if an object named *waldo.txt* exists in region *us-west1* can therefore be written as:

```
Invoke-CloudFunction -Provider GCP -Region us-west1 -Command ‘
  "return await $GCP_StorageBucket.file('waldo.txt').exists()"
```

So extending the test environment is as easy as defining and using string variables.

3.2 Deployment

To get the test environment up and running, we provide three `prepare.sh` scripts:

Amazon Web Services[86] This script asks the user for a prefix and creates Lambda functions[76] and S3 buckets in configured[87] Amazon Web Services regions, using the provided prefix and the region to generate the bucket name.

Microsoft Azure[88] Executing the script queries for the name of an Microsoft Azure resource group,[89] its primary location and the container name to be used. It then creates the resource group, as well as storage accounts, containers and function apps[77] for all configured[90] locations. The length of the resource group's name added to the length of the longest location name must not exceed 24 characters, as this is the maximum storage account name length,[91] and also how the name of storage accounts are generated.

Google Cloud Platform[92] As with the two previous scripts, functions[78] and buckets are created for configured[93] regions. As prefix the name of the Google Cloud Platform project[94] is queried and used.

All three scripts will install the provider's CLI if it's not present. In addition, they create the necessary permissions, roles and policies for accessing the buckets/containers publicly and being able to write to them using the provider's FaaS solution.
In addition there are also three `update.sh` scripts, which will redeploy all functions and settings if needed.[95;96;97]

3.3 Files and Images

After deployment, the buckets and containers are filled using the `Initialize-CloudFiles` function exported by the invocation script.[85] Figure 3 shows the lines that define what files are being created. As you can see, the files being generated are 512B, 10KiB, 100KiB and 10MiB in size,

```
const sizes = [512, 10*1024, 1024*1024, 10*1024*1024]
const files = [
  ['zero.bin', () => Buffer.alloc(0)],
  ['image_ping.png', () => png(() => null, 0)],
  ['image_full.png', () => png(() => randomBytes(10*1024), 408)],
  ...sizes.flatMap(size => [
    [`zero_${size}.bin`, () => Buffer.alloc(size)],
    [`random_${size}.bin`, () => randomBytes(size)]
  ])
]
```

Figure 3: File specifications in `Initialize-CloudFiles`.[85]

one set with random bytes, one with all zeros, as well as one empty file and two images.
The images are created using the `png` function, which takes as argument a function `chunkData` a number `count`, and returns an invalid PNG image. A PNG image is made of chunks of variable size, and supports public as well as private chunks, "to carry data that is not of interest to other applications".[98] Precisely such a chunk, named *fiLl*, is created using data returned from the `chunkData` function, and placed in the image `count` times. The capitalization in *fiLl* indicates that these chunks are ancillary, private and safe to copy,[98] so a parser will read and ignore them. Valid PNG files start with a specific signature, followed by an *IHDR* chunk describing the image, contain at least one *IDAT* chunk carrying the image data, and end with a *IEND* chunk.[98] The files returned by `png` also start with the signature and an *IHDR* chunk (indicating a 1000x1000 pixel RGB image), followed by `count` *fiLl* chunks and the final *IEND* chunk. There is, however, no *IDAT* chunk, which the cloud provider's image parser only knows after reading all the *fiLl* chunks, when it encounters *IEND*. The reason why multiple *fiFl* chunks are used is to ensure that the parser cannot just seek over one big *fiLl* chunk. Ideally, the size of one *fiLl* chunk matches the parser's read buffer size.

Once all tests are run (and to prevent unnecessary storage cover charges), `Clear-CloudFiles` can be used to delete all files from buckets/containers. `Copy-CloudFile` provides the ability to upload data from a local machine or download a file from some other location into a bucket or container, and `Test-CloudFile` checks if a file exists. All commands take as argument the name of a provider and region, and - just like `Invoke-CloudFunction` - can be piped in one of the helper variables `$All`, `$AWS`, `$Azure` or `$GCP`.

4 Results

The following results were generated using the test environment described in section 3. The file tests ran across 22 Amazon Web Services regions,[87], 32 Microsoft Azure locations,[90] and 23 Google Cloud Platform regions,[93] depicted in figure 1. OCR tests were performed on all Google Cloud Platform regions and a subset of Amazon Web Services (12[99]) and Microsoft Azure (28[100]).

4.1 File Transfer

As described in section 3.3 and shown in figure 3, 9 different files with 5 different sizes where used in the test. `Test-CloudFileTransfer`[85] was invoked with all possible combinations of regions, and at three different times: 12:00 Central European Time, 12:00 Japan Standard Time and 12:00 Eastern Standard Time. All $9 * (22 + 32 + 23)^2 * 3 = 160083$ data points are publicly available.[101]
Figure 4 shows the average transfer times over all different files, times and regions between

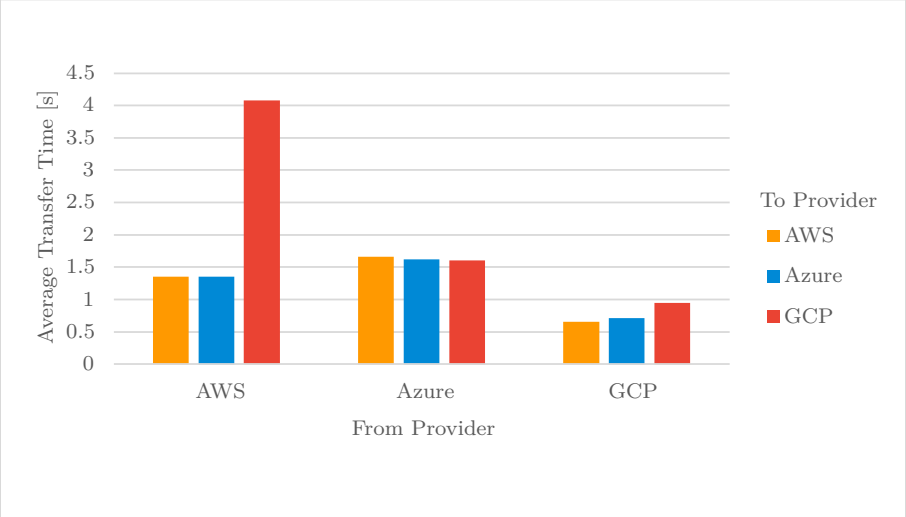
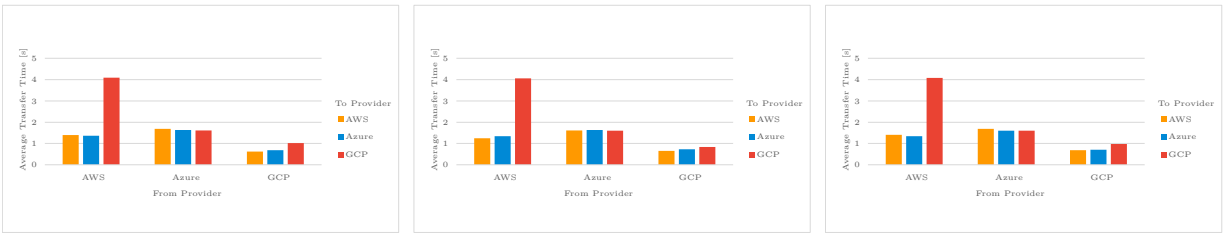


Figure 4: Average transfer time between providers.

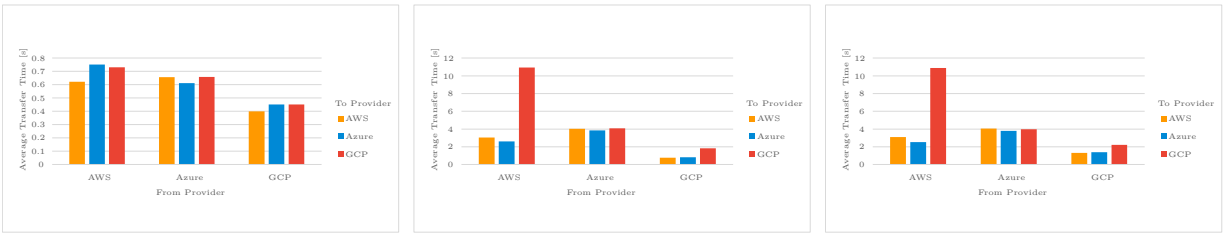
providers. While downloading from Microsoft Azure takes more or less the same time across all providers, transfer speeds between Amazon Web Services and Google Cloud Platform are definitely not symmetrical. Interestingly, transferring from Google Cloud Platform to itself takes almost double the time than to Microsoft Azure. Let's see if any of the components we averaged over can provide further explanation.
First we look at the same results but split up into the three different times of day in figure 5. Alas, but also unsurprisingly, these plots add no further insight, as the average transfer times



(a) 12:00 CET (b) 12:00 JST (c) 12:00 EST

Figure 5: Average transfer time between providers at different times.

remain almost the same. They are, after all, still averaged over all regions.
Next let's look at the different file types in figure 6. Here we can see the first significant difference to figure 4: Empty files, which can be seen as the equivalent to doing a *ping* at the storage level, take (almost) the same to every provider. Although 'pinging' Amazon Web Services storage takes longer than Google Cloud Platform's, there is no outlier like in figure 4. Yet the same outlier appears again for non-empty files. The difference between files containing random bytes and all zeroes seems to make no difference, except when downloading from Google Cloud Platform. Let's explore these two observations further.



(a) Empty 'ping' files. (b) Files containing all zeroes. (c) Files containing random bytes.

Figure 6: Average transfer time between providers using different file types.

Figure 7 combines the file types again, but shows the average transfer time by each file size. At

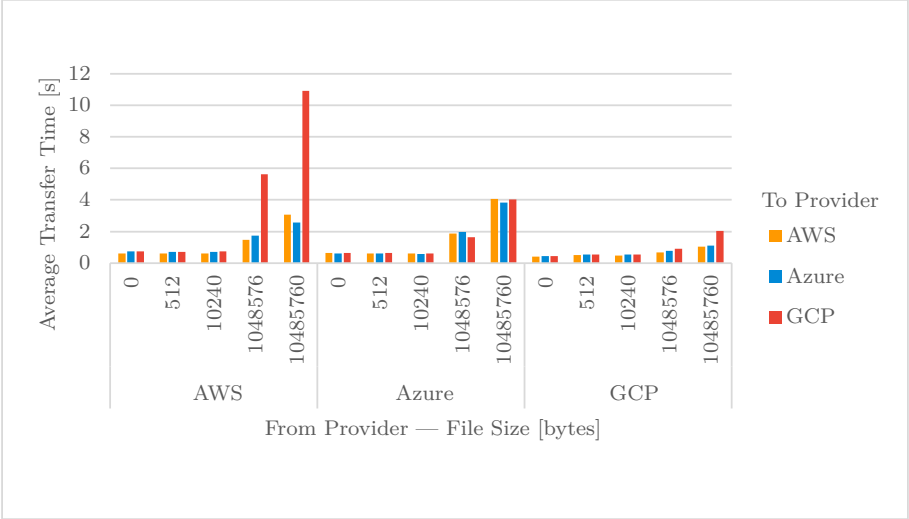
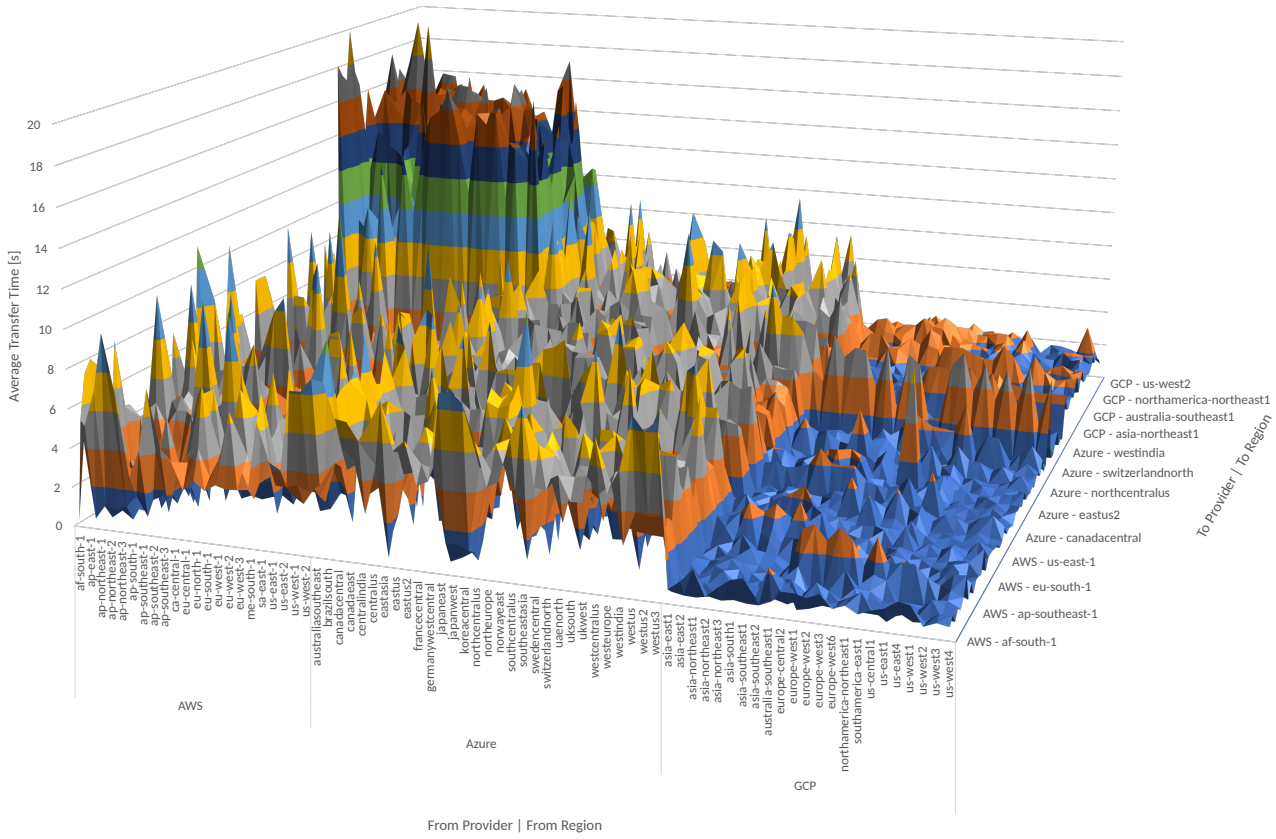


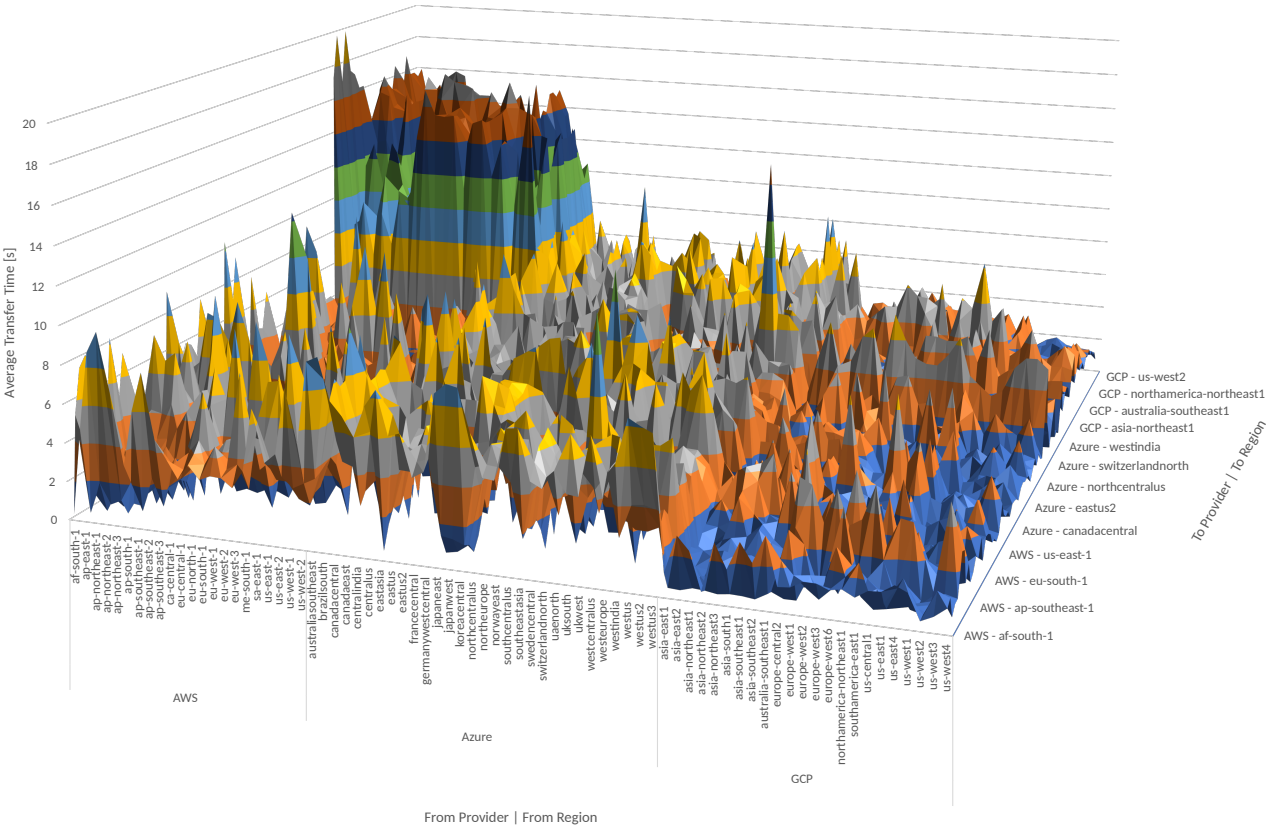
Figure 7: Average transfer time between providers by file size.

first glance it may appear as if transfers from Amazon Web Services to Google Cloud Platform actually scale exponentially in size, but at closer inspection, it becomes apparent that all other transfers just pick up speed faster (c.f. TCP slow start[102]). While blackbox testing cannot fully uncover the reason behind it, it is likely that some artificial rate limiting is employed, as downloading from Microsoft Azure to Google Cloud Platform doesn't show this behavior.
The difference between files of 0, 512 and 10240 bytes in size across all three providers is negligible, which makes sense since these amounts can be transferred in one packet[†], hence all act more or less as 'ping'. But let's have a closer look at the difference between files containing random data and all zeroes again.
To illustrate the difference, figure 8 shows a 3d surface of average transfer times between all regions,[‡] once for each file type (except empty files). While transfers from other providers seem unaffected, downloading files containing all zeroes - i.e. compressible files - from Google Cloud Platform results in shorter download times. If we consider section 2.2 and the fact that Google Cloud Platform transfers files internally over most of the path between the data center and the client, it is likely to assume some compression is indeed employed by the provider.
This concludes the file result section. Of course there are facets to explore (see section 5), let's move on to the OCR results nonetheless.

[†]With possible fragmentation, except for jumbograms.[103]
[‡]Except for *southafricanorth* and *australiaeast* on Microsoft Azure to improve visibility.



(a) Files containing all zeroes.



(b) Files containing random bytes.

Figure 8: Average transfer time between providers and regions.

4.2 OCR Transfer

As mentioned in section 2.4, measuring OCR transfer was done using the recognition service and storage of the same regions. `Test-CloudOcrTransfer`[85] was invoked with all 63 OCR-supporting regions twice, once at 00:00 UTC, and again at 12:00 UTC. Each invocations tests the file transfer and OCR execution speed of *image_ping.png* and *image_full.png* (see section 3.3) five times in a row, which we will call the five *requests* from here on out. All $63*2*2*2*5 = 2520$ data points are publicly available.[104] Figure 9 aggregates over regions, time and requests.
The values for OCR on *image_ping.png* show an expected higher overhead of invoking the service,

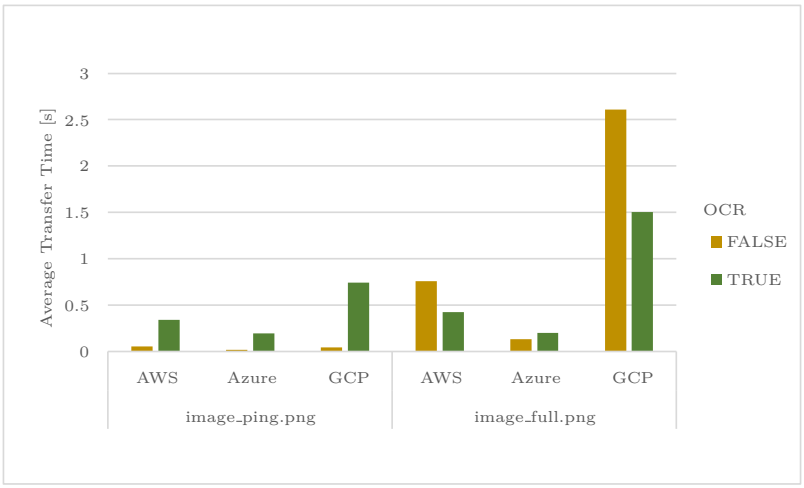
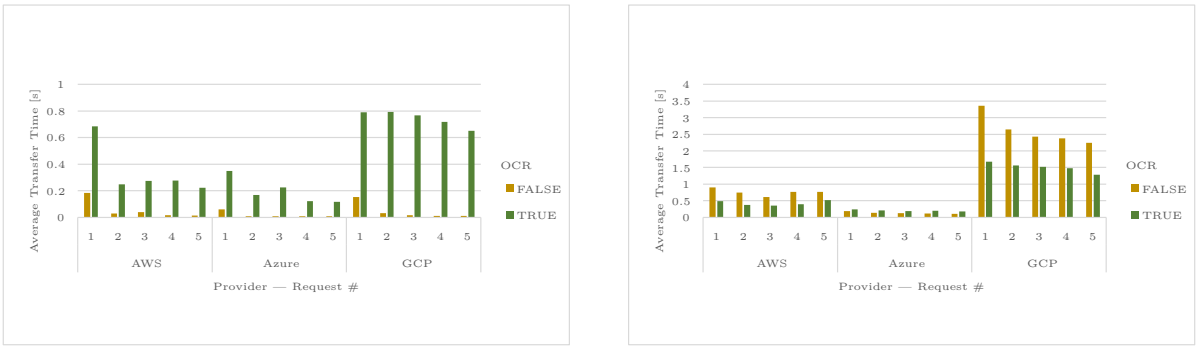


Figure 9: Average download time (OCR=FALSE) vs OCR execution time (OCR=TRUE).

while the non-OCR values correspond to the previous file tests. Looking at *image_full.png*, the first observation is that every provider offers faster transfers between the OCR service and the storage than between FaaS and the storage within each region. While not immediately obvious from figure 9, this also holds true for Microsoft Azure, as we have to subtract the OCR overhead and thus end up with a lower invocation time.
The second observation is that the download times returned by `Test-CloudOcrTransfer` are slightly higher than those of `Test-CloudFileTransfer` (c.f. section 4.1). This can be attributed to the fact that the former uses *fetch* and parses the results, while the latter uses the *node:https* module and discards downloaded chunks.[85] There is also a slight delay during the first request, as can be seen in figure 10 for *image_ping.png*. However, when fetching or running OCR on the full image, the delay becomes negligible.
While more comparisons - e.g. time of day or performance per region - could be shown, the



(a) *image_ping.png*

(b) *image_full.png*

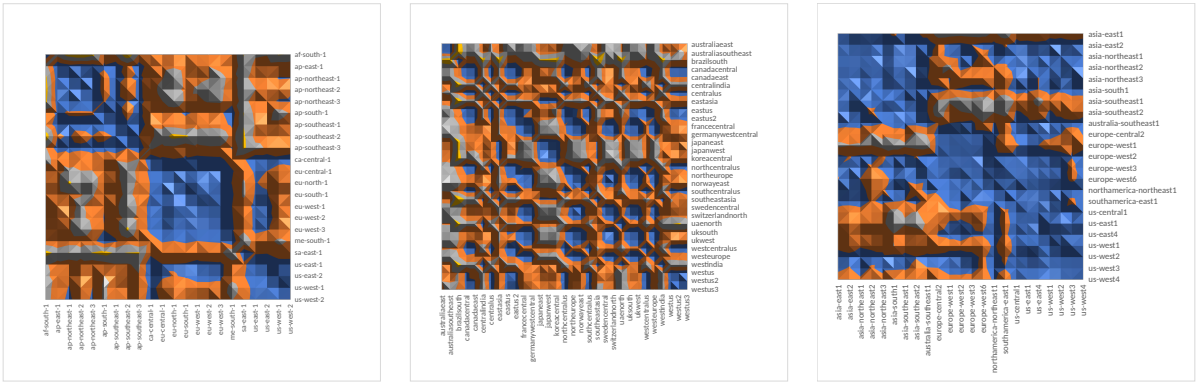
Figure 10: Average transfer time for each request.

results are similar to those presented in the previous section and therefore omitted for brevity.

5 Corollary and Future Work

The findings presented in section 4.1 are only a fraction of what can be gathered from the collected data. While we have shown that transfer speed between providers cannot be assumed to be symmetric, we only focused on the provider level. Yet, as figure 11 shows, there is clearly more to uncover at a regional level, since transferring between Google Cloud Platform regions is definitely not symmetrical either, unlike on Amazon Web Services and Microsoft Azure.
In multi-provider deployments, our OCR service results (section 4.2) can help decide which provider and region to target, depending on the image size. The next step here would be to run the tests across providers as well. While this won't work with Amazon Web Services, Microsoft Azure and Google Cloud Platform can download source images from anywhere, as we've discussed in section 2.4. Based on those results, schedulers could decide to either first transfer the file via storage and invoke the service within the same region, or let the service download it directly.
Another topic of interest not covered so far is the relation between geolocation and file transfers. While measuring transfer time and bandwidth has been used to calculate the location of a host before,[105;106] it would be interesting to see if enriching those methods with our approximately 6000 known triples of latency, bandwidth and distance can help improve accuracy.
Regarding bandwidth, we've seen that there is a stark difference in how transfers increase throughput with larger file size in section 4.1. However, since the largest files were only 10MiB in size, it is very likely that our tests did not uncovered the maximum bandwidth. The reason for not using larger files was simple economics, as transfers aren't cheap,[107] and, due to our full-mesh tests, scale exponentially in size. Running the tests again with a higher budget and simply changing one line (see section 3.3) would fill the gap.
Finally, all transfers in our tests so far have been downloads. And while figure 11 suggests that those are symmetrical within Microsoft Azure locations, initial tests have shown that this is not true for *uploading* data. That is to say that the speed with which region A downloads from region B, is not the same as region B uploads to region A. Especially in multi-regional deployments, knowing about these asymmetries can help schedulers decide **how** to move data, which until now just focused on **where** to move it to.

In any case, our Cloud Workbench can be used to implement and run future tests.



(a) Amazon Web Services (b) Microsoft Azure (c) Google Cloud Platform

Figure 11: Average transfer time between regions.

References

[1] Dana Petcu. Multi-cloud: Expectations and Current Approaches. In *Proceedings of the 2013 International Workshop on Multi-Cloud Applications and Federated Clouds*, pages 1–6, 2013.

[2] Aditi Rajan Khot. A Comparative Analysis of Public Cloud Platforms and Introduction of Multi-Cloud. *International Journal of Innovative Science and Research Technology*, 5: 448–454, 2020.

[3] Ataollah Fatahi Baarzi, George Kesidis, Carlee Joe-Wong, and Mohammad Shahrad. On Merits and Viability of Multi-Cloud Serverless. In *Proceedings of the ACM Symposium on Cloud Computing*, pages 600–608, 2021.

[4] Omar Alqaryouti, Nur Siyam, et al. Serverless Computing and Scheduling Tasks on Cloud: A Review. *American Academic Scientific Research Journal for Engineering, Technology, and Sciences*, 40(1):235–247, 2018.

[5] Minchen Yu, Tingjia Cao, Wei Wang, and Ruichuan Chen. Following the Data, Not the Function: Rethinking Function Orchestration in Serverless Computing. *arXiv preprint arXiv:2109.13492*, 2021.

[6] Jing Tai Piao and Jun Yan. A Network-Aware Virtual Machine Placement and Migration Approach in Cloud Computing. In *2010 Ninth International Conference on Grid and Cloud Computing*, pages 87–92. IEEE, 2010.

[7] Sobir Bazarbayev, Matti Hiltunen, Kaustubh Joshi, William H Sanders, and Richard Schlichting. Content-Based Scheduling of Virtual Machines (VMs) in the Cloud. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*, pages 93–101. IEEE, 2013.

[8] Sashko Ristov, Mika Hautz, Christian Hollaus, and Radu Prodan. SimLess: Simulate Serverless Workflows and Their Twins and Siblings in Federated FaaS. In *Proceedings of the 13th Symposium on Cloud Computing*, pages 323–339, 2022.

[9] Asif Imran, Md SQ Zulkar Nine, Kemal Guner, and Tevfik Kosar. A Vision for Cloud-Hosted Data Transfer Scheduling and Optimization as a Service, 2019.

[10] Liang Wang, Mengyuan Li, Yinqian Zhang, Thomas Ristenpart, and Michael Swift. Peeking Behind the Curtains of Serverless Platforms. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, pages 133–146, 2018.

[11] Oleksii Serhiienko and Josef Spillner. Systematic and Recomputable Comparison of Multi-Cloud Management Platforms. In *2018 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 107–114. IEEE, 2018.

[12] Amazon Web Services. Rekognition - DetectText. `https://docs.aws.amazon.com/rekognition/latest/APIReference/API_DetectText.html`, (Accessed on 01/02/2023).

[13] Microsoft Azure. Cognitive Services - Computer Vision - Recognize Printed Text. `https://learn.microsoft.com/en-us/rest/api/computervision/3.1/recognize-printed-text/recognize-printed-text`, (Accessed on 12/23/2022).

[14] Google Cloud Platform. Cloud Vision - AnnotateImageRequest. `https://cloud.google.com/vision/docs/reference/rest/v1/AnnotateImageRequest`, (Accessed on 12/23/2022).

[15] Amazon Web Services. Simple Storage Service - Objects. https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingObjects.html, (Accessed on 01/01/2023).

[16] Google Cloud Platform. Cloud Storage - About Objects. https://cloud.google.com/storage/docs/objects, (Accessed on 01/02/2023).

[17] Microsoft Azure. Blob Storage. https://azure.microsoft.com/en-us/products/storage/blobs/, (Accessed on 01/01/2023).

[18] Amazon Web Services. Simple Storage Service. https://docs.aws.amazon.com/AmazonS3/latest/userguide/s3-access-control.html, (Accessed on 01/01/2023).

[19] Microsoft Azure. Blob Storage - Authorize Operations for Data Access. https://learn.microsoft.com/en-us/azure/storage/common/authorize-data-access, (Accessed on 01/01/2023).

[20] Google Cloud Platform. Cloud Storage - Overview of Access Control. https://cloud.google.com/storage/docs/access-control, (Accessed on 01/01/2023).

[21] Amazon Web Services. Simple Storage Service - Organizing Objects Using Folders. https://docs.aws.amazon.com/AmazonS3/latest/userguide/using-folders.html, (Accessed on 01/02/2023).

[22] Google Cloud Platform. Cloud Storage - About Buckets. https://cloud.google.com/storage/docs/buckets, (Accessed on 01/02/2023).

[23] Amazon Web Services. Simple Storage Service - Object Storage Classes. https://aws.amazon.com/s3/storage-classes/, (Accessed on 01/01/2023).

[24] Microsoft Azure. Blob Storage - Access Tiers. https://learn.microsoft.com/en-us/azure/storage/blobs/access-tiers-overview, (Accessed on 01/01/2023).

[25] Google Cloud Platform. Cloud Storage - Storage classes. https://cloud.google.com/storage/docs/storage-classes, (Accessed on 01/01/2023).

[26] Amazon Web Services. Simple Storage Service - Methods for Accessing a Bucket. https://docs.aws.amazon.com/AmazonS3/latest/userguide/access-bucket-intro.html, (Accessed on 01/03/2023).

[27] Microsoft Azure. Introduction to Blob Storage. https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction, (Accessed on 01/03/2023).

[28] Microsoft Azure. Storage Account Overview. https://learn.microsoft.com/en-us/azure/storage/common/storage-account-overview, (Accessed on 01/03/2023).

[29] Google Cloud Platform. Cloud Storage - Access Public Data. https://cloud.google.com/storage/docs/access-public-data, (Accessed on 01/03/2023).

[30] Cloudflare. What is Anycast? https://www.cloudflare.com/learning/cdn/glossary/anycast-network/, (Accessed on 01/03/2023).

[31] Amazon Web Services. Elastic Compute Cloud - Regions and Zones. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html, (Accessed on 01/03/2023).

[32] Microsoft Azure. Choose the Right Azure Region for You. https://azure.microsoft.com/en-us/explore/global-infrastructure/geographies/, (Accessed on 01/03/2023).

[33] Google Cloud Platform. Compute Engine - Regions and Zones. https://cloud.google.com/compute/docs/regions-zones, (Accessed on 01/03/2023).

[34] Amazon Web Services. botocore Python SDK - ClientCreator. https://github.com/boto/botocore/blob/dddd199cd287999a249d9001077a62d5fab93d3a/botocore/client.py#L109, (Accessed on 12/23/2022).

[35] Microsoft Azure. Cognitive Services Python SDK - Computer Vision Client Operations. https://github.com/Azure/azure-sdk-for-python/blob/c45589ebb4e26d5d2fdd9be3961656dacc5aa927/sdk/cognitiveservices/azure-cognitiveservices-vision-computervision/azure/cognitiveservices/vision/computervision/operations/_computer_vision_client_operations.py, (Accessed on 12/23/2022).

[36] Google Cloud Platform. Cloud Vision Python SDK - Image Annotator. https://github.com/googleapis/python-vision/blob/ed9548afdca70b95b5f87b825c770c4569441aa6/google/cloud/vision_v1/services/image_annotator/transports/grpc.py, (Accessed on 12/23/2022).

[37] Amazon Web Services. Data Lakes and Analytics on AWS. https://aws.amazon.com/big-data/datalakes-and-analytics/, (Accessed on 01/02/2023).

[38] Microsoft Azure. Introduction to Azure Files. https://learn.microsoft.com/en-us/training/modules/introduction-to-azure-files/, (Accessed on 01/02/2023).

[39] Google Cloud Platform. Cloud Storage - Transfer Service. https://cloud.google.com/storage-transfer/docs/overview, (Accessed on 01/02/2023).

[40] Amazon Web Services. Rekognition - Endpoints and Quotas. https://docs.aws.amazon.com/general/latest/gr/rekognition.html, (Accessed on 01/02/2023).

[41] Microsoft Azure. Cognitive Services APIs. https://centraluseuap.dev.cognitive.microsoft.com/docs/services/computer-vision-v3-2/operations/5d986960601faab4bf452005, (Accessed on 01/02/2023).

[42] Google Cloud Platform. Cloud Vision - Detect Text in Images. https://cloud.google.com/vision/docs/ocr, (Accessed on 01/02/2023).

[43] Amazon Web Services. Translate - TranslateText. https://docs.aws.amazon.com/translate/latest/APIReference/API_TranslateText.html, (Accessed on 01/02/2023).

[44] Microsoft Azure. Cognitive Services - Translator Translate Method. https://learn.microsoft.com/en-us/azure/cognitive-services/translator/reference/v3-0-translate, (Accessed on 01/02/2023).

[45] Google Cloud Platform. Cloud Translation - TranslateText. https://cloud.google.com/translate/docs/reference/rest/v3/projects/translateText, (Accessed on 01/02/2023).

[46] Microsoft Azure. Cognitive Services - Speech service - Speech-To-Text REST API for Short Audio. https://learn.microsoft.com/en-us/azure/cognitive-services/speech-service/rest-speech-to-text-short, (Accessed on 01/02/2023).

[47] Amazon Web Services. Transcribe - Actions. https://docs.aws.amazon.com/transcribe/latest/APIReference/API_Operations.html, (Accessed on 01/02/2023).

[48] Amazon Web Services. Rekognition - S3Object. https://docs.aws.amazon.com/rekognition/latest/APIReference/API_S3Object.html, (Accessed on 01/02/2023).

[49] Amazon Web Services. Simple Storage Service - S3 Object Lock Retention. https://docs.aws.amazon.com/AmazonS3/latest/userguide/batch-ops-retention-date.html, (Accessed on 01/04/2023).

[50] Amazon Web Services. Backup - Restoring S3 Data. https://docs.aws.amazon.com/aws-backup/latest/devguide/restoring-s3.html, (Accessed on 01/04/2023).

[51] Amazon Web Services. Simple Storage Service - Restoring Previous Versions. https://docs.aws.amazon.com/AmazonS3/latest/userguide/RestoringPreviousVersions.html, (Accessed on 01/04/2023).

[52] Microsoft Azure. Storage - Retention Policies. https://learn.microsoft.com/en-us/azure/storage/blobs/immutable-time-based-retention-policy-overview, (Accessed on 01/04/2023).

[53] Microsoft Azure. Storage - Restore Container. https://learn.microsoft.com/en-us/rest/api/storageservices/restore-container, (Accessed on 01/04/2023).

[54] Microsoft Azure. Storage - Undelete Blob. https://learn.microsoft.com/en-us/rest/api/storageservices/undelete-blob, (Accessed on 01/04/2023).

[55] Google Cloud Platform. Cloud Storage - Retention Policies. https://cloud.google.com/storage/docs/bucket-lock, (Accessed on 01/04/2023).

[56] Google Cloud Platform. Cloud Storage - Object Versioning. https://cloud.google.com/storage/docs/object-versioning, (Accessed on 01/04/2023).

[57] Amazon Web Services. Simple Storage Service - CopyObject. https://docs.aws.amazon.com/AmazonS3/latest/API/API_CopyObject.html, (Accessed on 01/04/2023).

[58] Microsoft Azure. Storage - Copy Blob. https://learn.microsoft.com/en-us/rest/api/storageservices/copy-blob, (Accessed on 01/04/2023).

[59] Google Cloud Platform. Cloud Storage - Rewrite. https://cloud.google.com/storage/docs/json_api/v1/objects/rewrite, (Accessed on 01/04/2023).

[60] Microsoft Azure. Storage - Copy Blob From URL. https://learn.microsoft.com/en-us/rest/api/storageservices/copy-blob-from-url, (Accessed on 01/04/2023).

[61] Google Cloud Platform. Cloud Storage - Copy. https://cloud.google.com/storage/docs/json_api/v1/objects/copy, (Accessed on 01/04/2023).

[62] Amazon Web Services. Simple Storage Service - Uploading and Copying Objects Using Multipart Upload. https://docs.aws.amazon.com/AmazonS3/latest/userguide/mpuoverview.html, (Accessed on 01/04/2023).

[63] Amazon Web Services. Simple Storage Service - UploadPartCopy. https://docs.aws.amazon.com/AmazonS3/latest/API/API_UploadPartCopy.html, (Accessed on 01/04/2023).

[64] Microsoft Azure. Storage - Put Block From URL. https://learn.microsoft.com/en-us/rest/api/storageservices/put-block-from-url, (Accessed on 01/04/2023).

[65] Microsoft Azure. Storage - Put Page From URL. https://learn.microsoft.com/en-us/rest/api/storageservices/put-page-from-url, (Accessed on 01/04/2023).

[66] Google Cloud Platform. Cloud Storage - Multipart Uploads. https://cloud.google.com/storage/docs/multipart-uploads, (Accessed on 01/04/2023).

[67] Google Cloud Platform. Cloud Storage - Perform Resumable Uploads. https://cloud.google.com/storage/docs/performing-resumable-uploads, (Accessed on 01/04/2023).

[68] Google Cloud Platform. Cloud Storage - Compose. https://cloud.google.com/storage/docs/json_api/v1/objects/compose, (Accessed on 01/04/2023).

[69] Amazon Web Services. Simple Storage Service - UploadPart. https://docs.aws.amazon.com/AmazonS3/latest/API/API_UploadPart.html, (Accessed on 01/04/2023).

[70] Microsoft Azure. Storage - Understanding Blobs. https://learn.microsoft.com/en-us/rest/api/storageservices/understanding-block-blobs--append-blobs--and-page-blobs, (Accessed on 01/04/2023).

[71] Microsoft Azure. Storage - Put Block List. https://learn.microsoft.com/en-us/rest/api/storageservices/put-block-list, (Accessed on 01/04/2023).

[72] Google Cloud Platform. Cloud Storage - Upload Part. https://cloud.google.com/storage/docs/xml-api/put-object-multipart, (Accessed on 01/04/2023).

[73] Google Cloud Platform. Cloud Storage - Quotas & Limits. https://cloud.google.com/storage/quotas#requests, (Accessed on 01/04/2023).

[74] Google Cloud Platform. Cloud Storage - Resumable Uploads. https://cloud.google.com/storage/docs/resumable-uploads, (Accessed on 01/04/2023).

[75] Cloud Workbench. A Tool for Evaluating AWS, Azure and GCP. https://github.com/meitinger/CloudWorkbench, (Accessed on 01/10/2023).

[76] Amazon Web Services. Serverless Computing. https://aws.amazon.com/lambda/, (Accessed on 01/10/2023).

[77] Microsoft Azure. Serverless Functions in Computing. https://azure.microsoft.com/en-us/products/functions/, (Accessed on 01/10/2023).

[78] Google Cloud Platform. Cloud Functions. https://cloud.google.com/functions, (Accessed on 01/10/2023).

[79] Google Cloud Platform. Deploy a Cloud Function. https://cloud.google.com/functions/docs/deploy, (Accessed on 01/10/2023).

[80] Cloud Workbench. Amazon Web Services Function. https://github.com/meitinger/CloudWorkbench/blob/3b4e64839796633f7965501196f6c76607b32205/aws/function/index.mjs, (Accessed on 01/10/2023).

[81] Cloud Workbench. Microsoft Azure Function. https://github.com/meitinger/CloudWorkbench/blob/3b4e64839796633f7965501196f6c76607b32205/azure/function/v1/index.js, (Accessed on 01/10/2023).

[82] Cloud Workbench. Google Cloud Platform Function. https://github.com/meitinger/CloudWorkbench/blob/3b4e64839796633f7965501196f6c76607b32205/gcp/function/index.js, (Accessed on 01/10/2023).

[83] Microsoft Azure. Securing Azure Functions. https://learn.microsoft.com/en-us/azure/azure-functions/security-concepts?tabs=v4, (Accessed on 01/10/2023).

[84] Hugo Krawczyk, Mihir Bellare, and Ran Canetti. RFC2104: HMAC: Keyed-Hashing for Message Authentication, 1997.

[85] Cloud Workbench. PowerShell Invoke Script. https://github.com/meitinger/CloudWorkbench/blob/0294c33753d8de36098d1a32210c73fb1eafbb00/invoke.ps1, (Accessed on 01/11/2023).

[86] Cloud Workbench. Prepare Script - Amazon Web Services. https://github.com/meitinger/CloudWorkbench/blob/3b4e64839796633f7965501196f6c76607b32205/aws/prepare.sh, (Accessed on 01/12/2023).

[87] Cloud Workbench. Regions - Amazon Web Services. https://github.com/meitinger/CloudWorkbench/blob/3b4e64839796633f7965501196f6c76607b32205/aws/regions.txt, (Accessed on 01/12/2023).

[88] Cloud Workbench. Prepare Script - Microsoft Azure. https://github.com/meitinger/CloudWorkbench/blob/3b4e64839796633f7965501196f6c76607b32205/azure/prepare.sh, (Accessed on 01/12/2023).

[89] Microsoft Azure. Resources Organization. https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/azure-setup-guide/organize-resources, (Accessed on 01/12/2023).

[90] Cloud Workbench. Regions - Microsoft Azure. https://github.com/meitinger/CloudWorkbench/blob/3b4e64839796633f7965501196f6c76607b32205/azure/regions.txt, (Accessed on 01/12/2023).

[91] Microsoft Azure. Storage Account Overview. https://learn.microsoft.com/en-us/azure/storage/common/storage-account-overview, (Accessed on 01/13/2023).

[92] Cloud Workbench. Prepare Script - Google Cloud Platform. https://github.com/meitinger/CloudWorkbench/blob/3b4e64839796633f7965501196f6c76607b32205/gcp/prepare.sh, (Accessed on 01/12/2023).

[93] Cloud Workbench. Regions - Google Cloud Platform. https://github.com/meitinger/CloudWorkbench/blob/3b4e64839796633f7965501196f6c76607b32205/gcp/regions.txt, (Accessed on 01/13/2023).

[94] Google Cloud Platform. Project Resources. https://cloud.google.com/resource-manager/docs/cloud-platform-resource-hierarchy#projects, (Accessed on 01/13/2023).

[95] Cloud Workbench. Update Script - Amazon Web Services. https://github.com/meitinger/CloudWorkbench/blob/0294c33753d8de36098d1a32210c73fb1eafbb00/aws/update.sh, (Accessed on 01/15/2023).

[96] Cloud Workbench. Update Script - Microsoft Azure. https://github.com/meitinger/CloudWorkbench/blob/0294c33753d8de36098d1a32210c73fb1eafbb00/azure/update.sh, (Accessed on 01/15/2023).

[97] Cloud Workbench. Update Script - Google Cloud Platform. https://github.com/meitinger/CloudWorkbench/blob/0294c33753d8de36098d1a32210c73fb1eafbb00/gcp/update.sh, (Accessed on 01/15/2023).

[98] Thomas Boutell. PNG (Portable Network Graphics) Specification Version 1.0, 1997.

[99] Cloud Workbench. Rekognition Regions - Amazon Web Services. https://github.com/meitinger/CloudWorkbench/blob/0294c33753d8de36098d1a32210c73fb1eafbb00/aws/rekognition-regions.txt, (Accessed on 01/16/2023).

[100] Cloud Workbench. Cognitive Regions - Microsoft Azure. https://github.com/meitinger/CloudWorkbench/blob/0294c33753d8de36098d1a32210c73fb1eafbb00/azure/cognitive-regions.txt, (Accessed on 01/16/2023).

[101] Cloud Workbench. File Transfer Results. https://github.com/meitinger/CloudWorkbench/blob/0294c33753d8de36098d1a32210c73fb1eafbb00/fileresults.csv, (Accessed on 01/16/2023).

[102] Mark Allman, Vern Paxson, and Ethan Blanton. TCP Congestion Control, 2009.

[103] David Borman, Steve Deering, and Robert Hinden. IPv6 Jumbograms, 1999.

[104] Cloud Workbench. OCR Transfer Results. https://github.com/meitinger/CloudWorkbench/blob/0294c33753d8de36098d1a32210c73fb1eafbb00/ocrresults.csv, (Accessed on 01/16/2023).

[105] Minzheng Jia, Yuanzhong Zhu, and Hongyu Yang. An IP Geolocation Algorithm Using Bandwidth Measurement and Distance Estimation. In *Proceedings of the 2015 International Conference on Electrical and Information Technologies for Rail Transportation*, pages 519–527. Springer, 2016.

[106] Shichang Ding, Xiangyang Luo, Dengpan Ye, and Fenlin Liu. Delay-Distance Correlation Study for IP Geolocation. *Wuhan University Journal of Natural Sciences*, 22(2):157–164, 2017.

[107] Microsoft Azure. Pricing - Bandwidth. https://azure.microsoft.com/en-us/pricing/details/bandwidth/, (Accessed on 01/18/2023).