Skyplane

The Intercloud Broker for Data

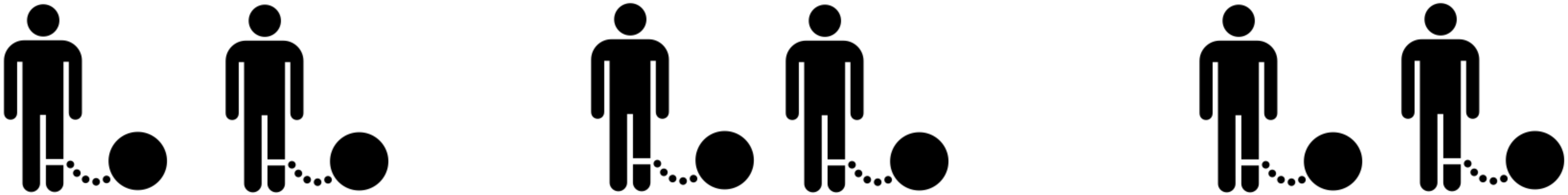
skyplane.org

Presented by Sarah Wooders
Sky Computing Lab, UC Berkeley

Cloud Data Today: Partitioned by Region, Provider, & Services

Cloud Data Today: Partitioned by Region, Provider, & Services

Locked in by *data gravity*



aws

EC2, S3, RDS, etc…

Google Cloud

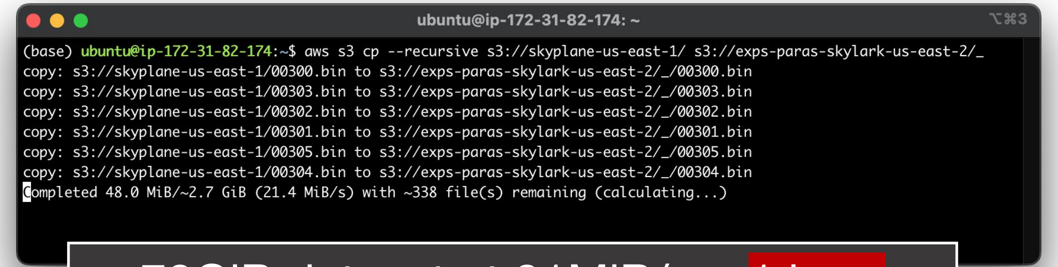
VMs, GCS, BigQuery, etc…

Azure

VMs, Blob, Synapse, etc…

The problem of **data gravity**

1. Slow transfers lock in data



70GiB dataset at 21MiB/s = 1 hour

2. Expensive egress fees = $$$

Cost to move 70GB dataset = running **34 instances** (m5.xlarge)

3. Painful integrations

Working with data in the cloud is painful

» bash [paras@Paras-M1-Mac.local] ⌥⌘2

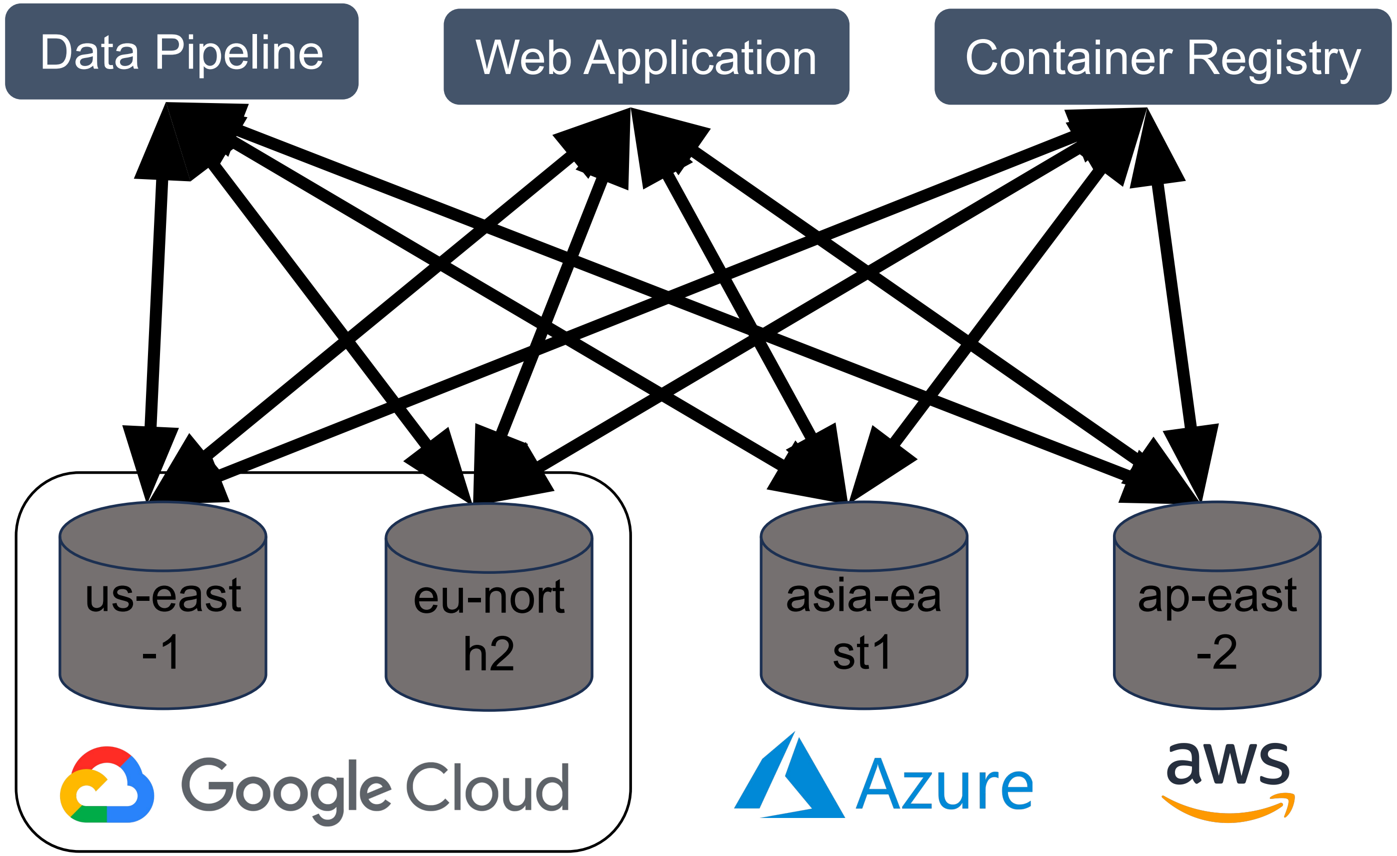
» ip-172-31-82-174 ⌘1 | » ip-172-31-82-174 ⌘2

```
(base) ubuntu@ip-172-31-82-174:~$ aws s3 cp --recursive s3://skyplane-us-east-1/ s3://exps-paras-sky
lark-us-east-2/ordered/
```

What is Skyplane?

Problem: Managing data across regions and across clouds is **slow** and **expensive**

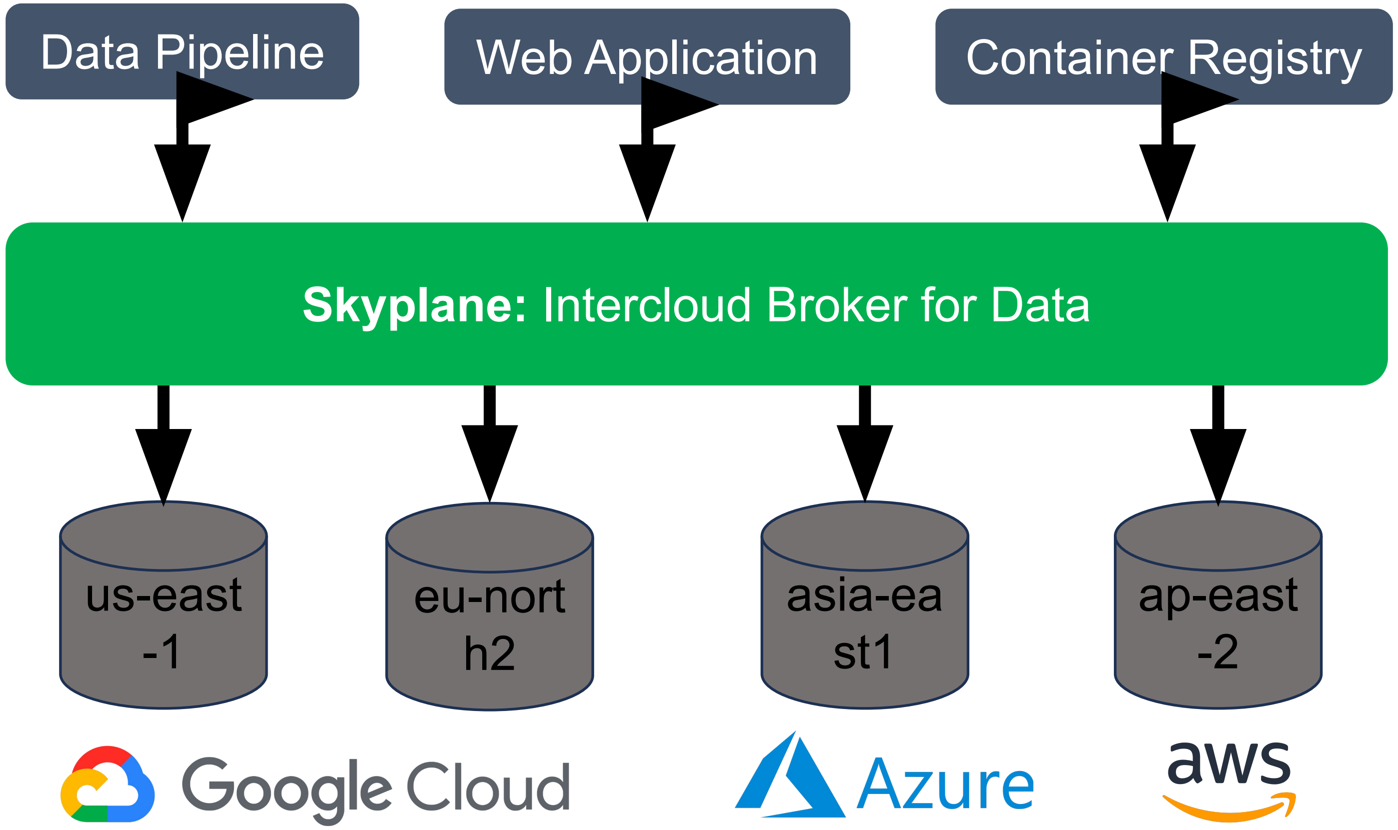
Life in the cloud today



What is Skyplane?

Problem: Managing data across regions and across clouds is **slow** and **expensive**

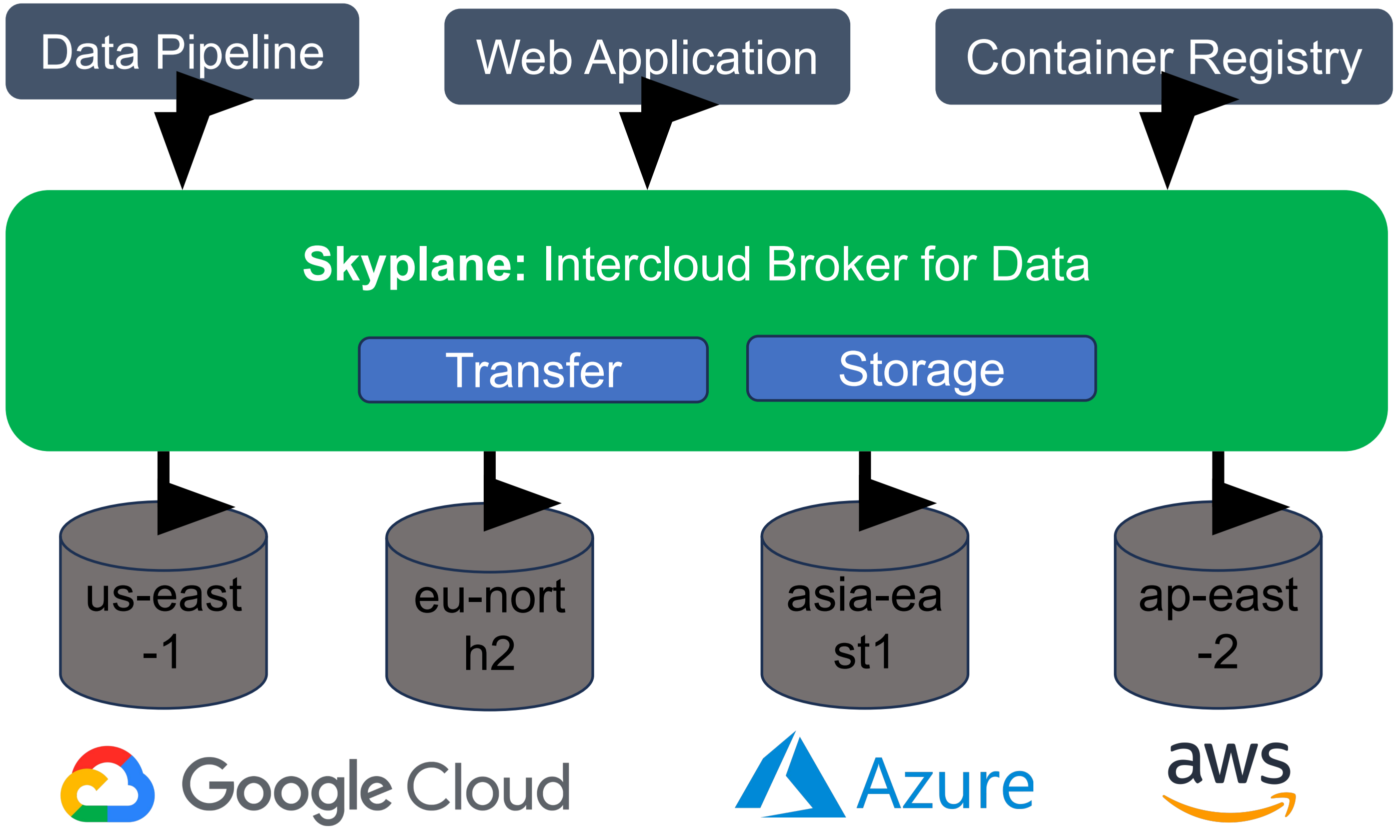
Skyplane is an Intercloud Broker for managing data across cloud providers.



What is Skyplane?

Problem: Managing data across regions and across clouds is **slow** and **expensive**

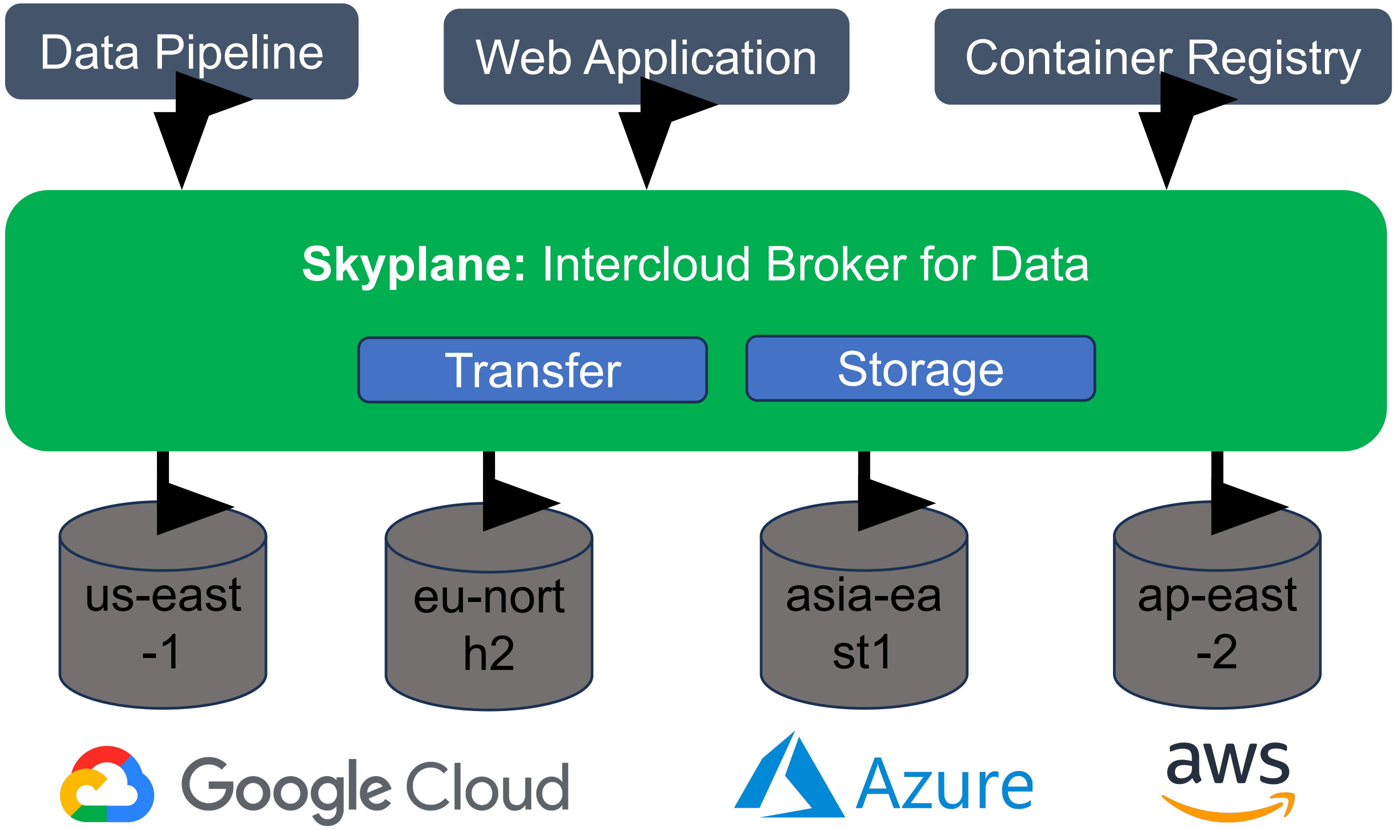
Skyplane is an Intercloud Broker for managing data across cloud providers.



What is Skyplane?

Problem: Managing data across regions and across clouds is **slow** and **expensive**

Skyplane is an Intercloud Broker for managing data across cloud providers.



1: Cross-cloud transfer broker

```
skyplane cp {s3,gs,az}://... {s3,gs,az}://...
```

2: Cross-cloud storage broker

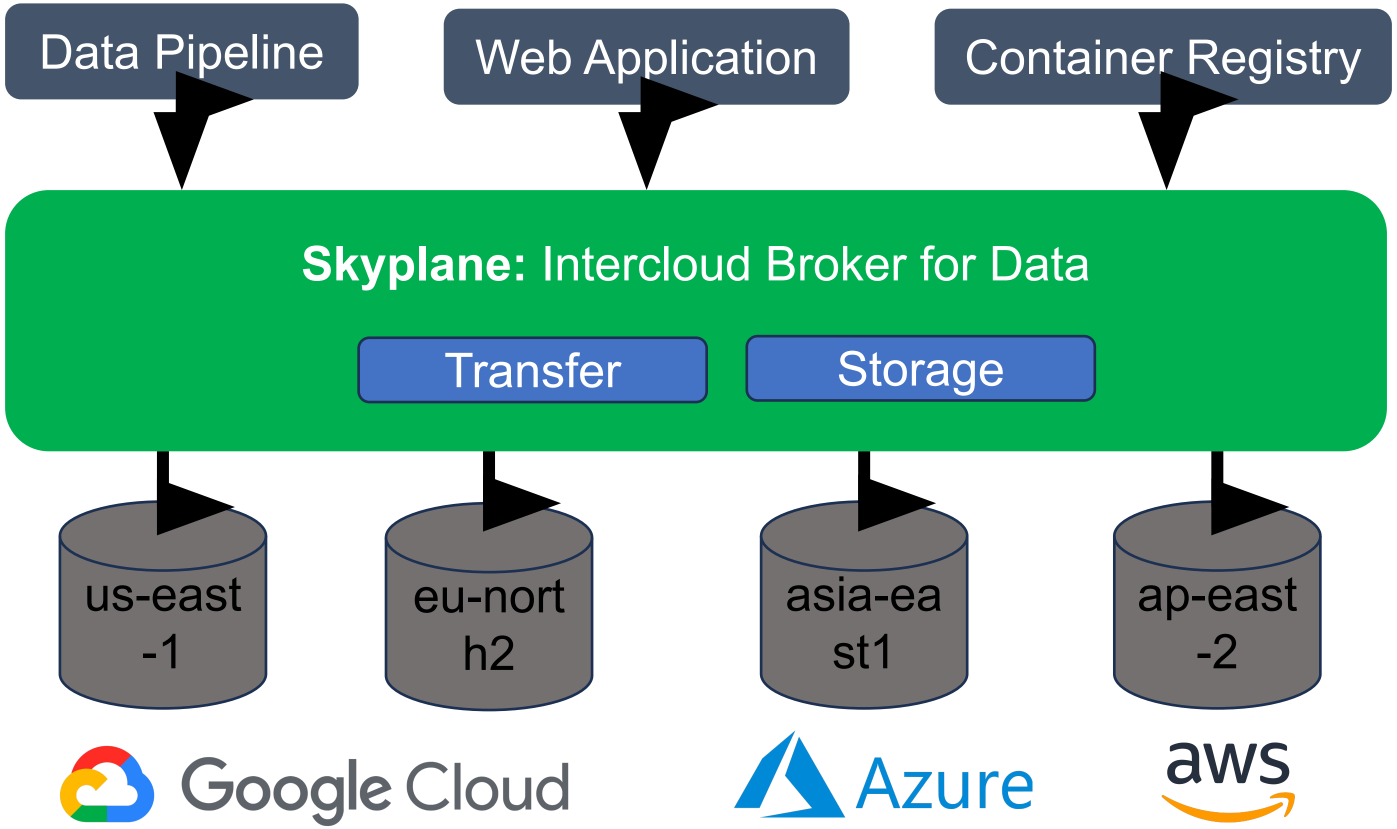
```
boto3.download_file("sky://imagenet/0001.tfrecord")
```

skyplane.org

What is Skyplane?

Problem: Managing data across regions and across clouds is **slow** and **expensive**

Skyplane is an Intercloud Broker for managing data across cloud providers.



1: Cross-cloud transfer broker

```
skyplane cp {s3,gs,az}://... {s3,gs,az}://...
```

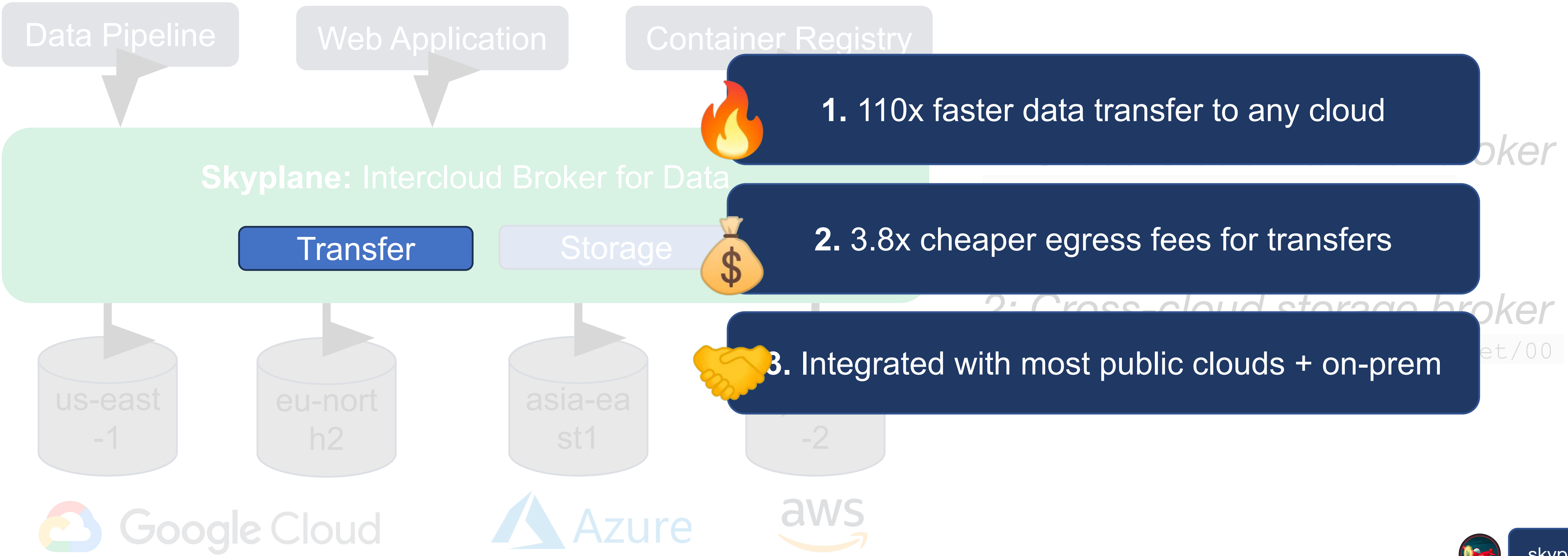
2: Cross-cloud storage broker

```
boto3.download_file("sky://imagenet/0001.tfrecord")
```

What is Skyplane?

Problem: Managing data across regions and across clouds is **slow** and **expensive**

Skyplane is an Intercloud Broker for managing data across cloud providers.



1. 110x faster data transfer to any cloud
2. 3.8x cheaper egress fees for transfers
3. Integrated with most public clouds + on-prem

Transfer Broker: Optimizing Cloud Networking

Unicast: Point-to-point replication

"Move 5TB from AWS to GCP"

Multicast: One-to-many replication

"Publish 300GB model weights to 10 regions"

Service catalog:
- Map of cloud WAN
- Throughput grid
- Price grid

→

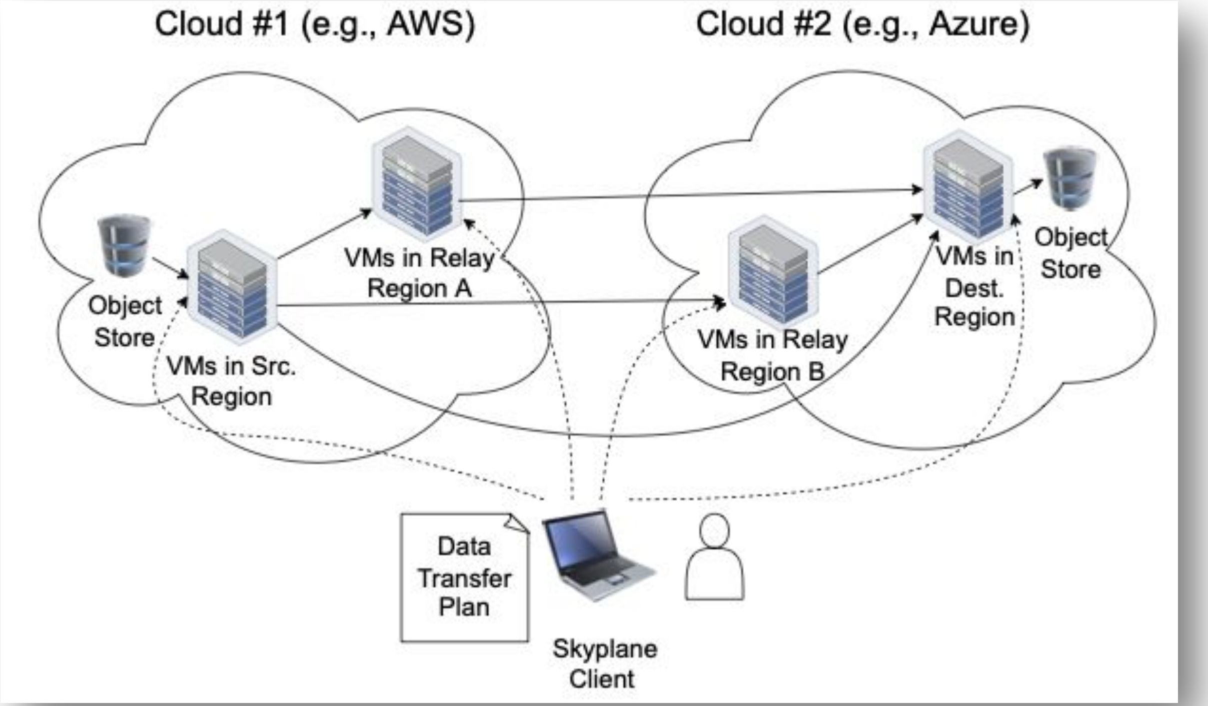
Skyplane optimizer
- Min cost
- Max throughput

→

Execute over Skyplane overlay network

CLOUDFLARE · aws · Google Cloud · Azure · IBM Cloud · On prem

Skyplane Overlay Network



No cooperation required from clouds!

Skyplane only uses public APIs + runs in your cloud VPC

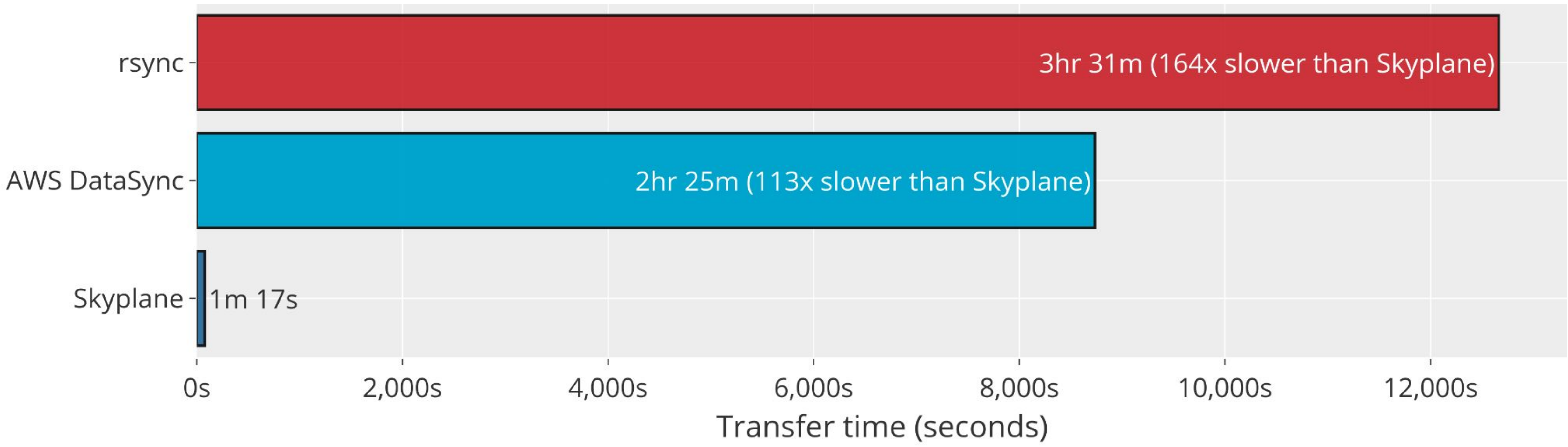
110x faster data transfers with the Skyplane transfer broker

» bash [paras@Paras-M1-Mac.local] ⌥⌘2

» ip-172-31-82-174 ⌘1 | » ip-172-31-82-174 ⌘2

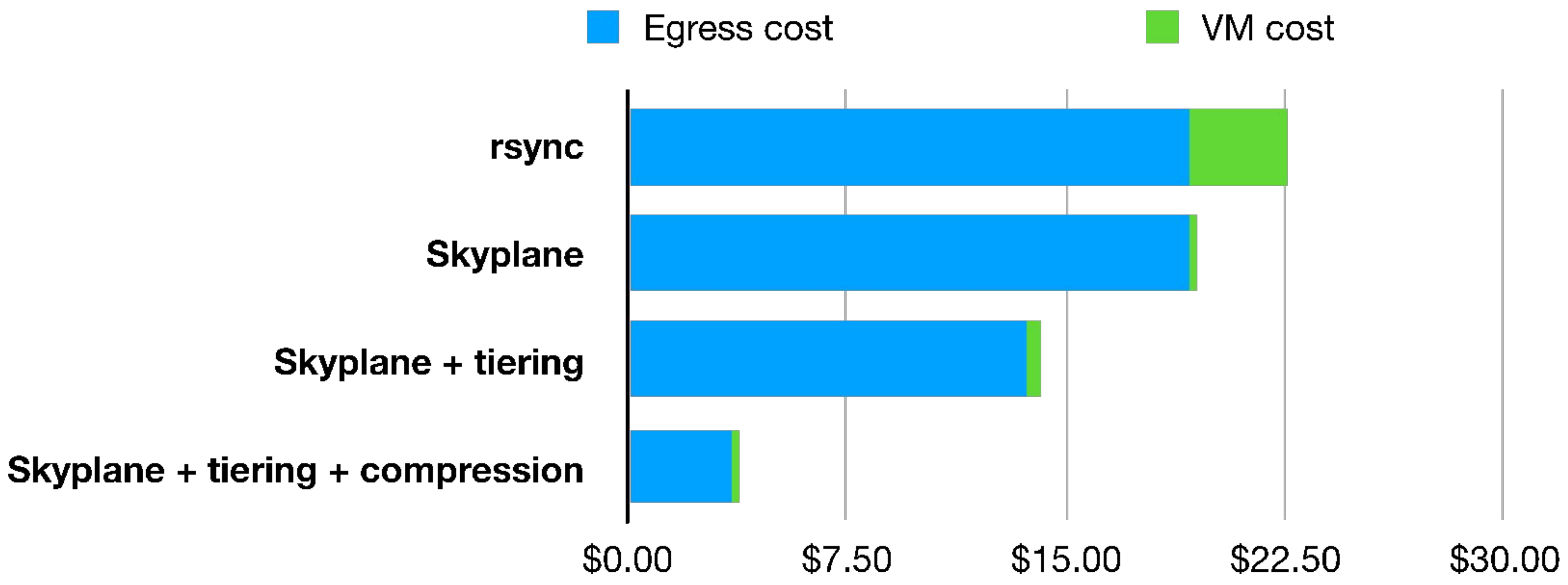
```
(base) ubuntu@ip-172-31-82-174:~/skylark$ skyplane cp s3://exps-paras-skylark-us-east-1/fake_imagenet/ s3://skyplan
e-demo-us-east-1/imagenet
```

All together: 110x faster transfers with Skyplane



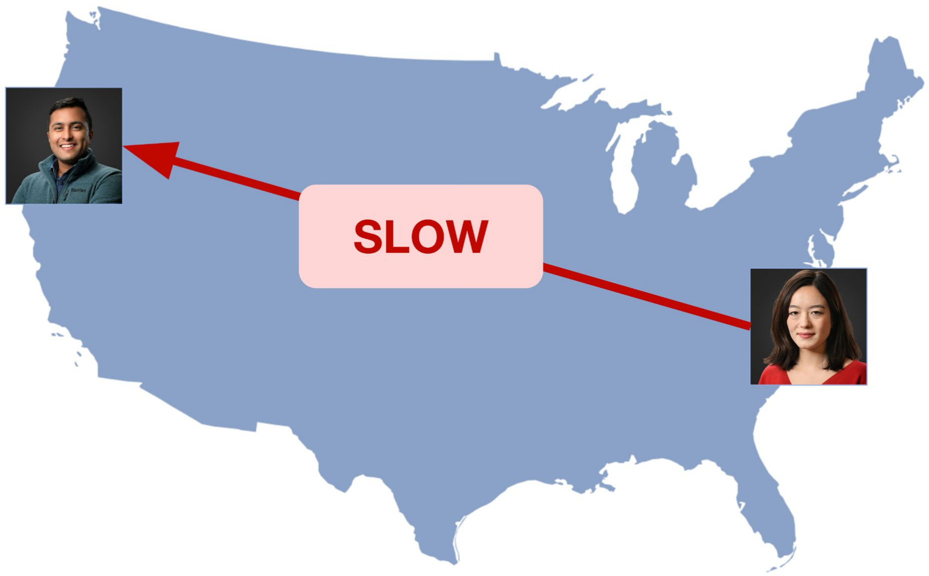
Full benchmarks online at https://skyplane.org/en/latest/benchmark.html

All together: 3.8x cheaper transfers with Skyplane



Full benchmarks online at https://skyplane.org/en/latest/benchmark.html

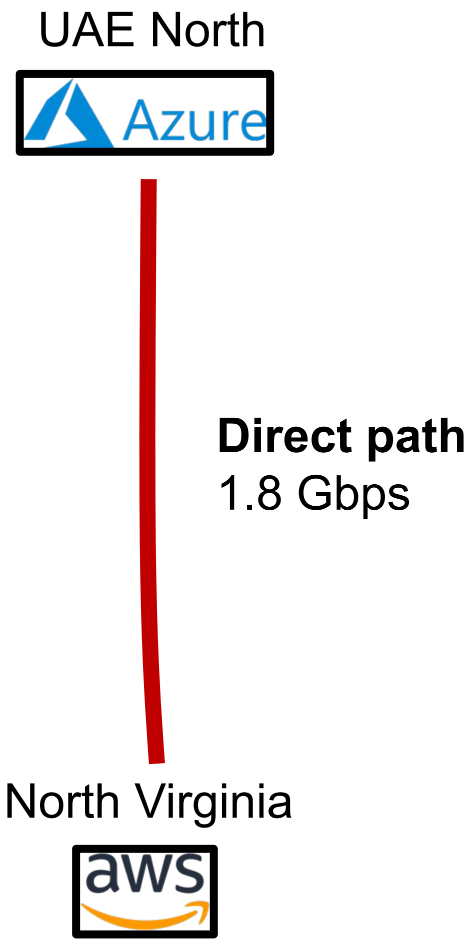
Direct internet path between clouds are often slow



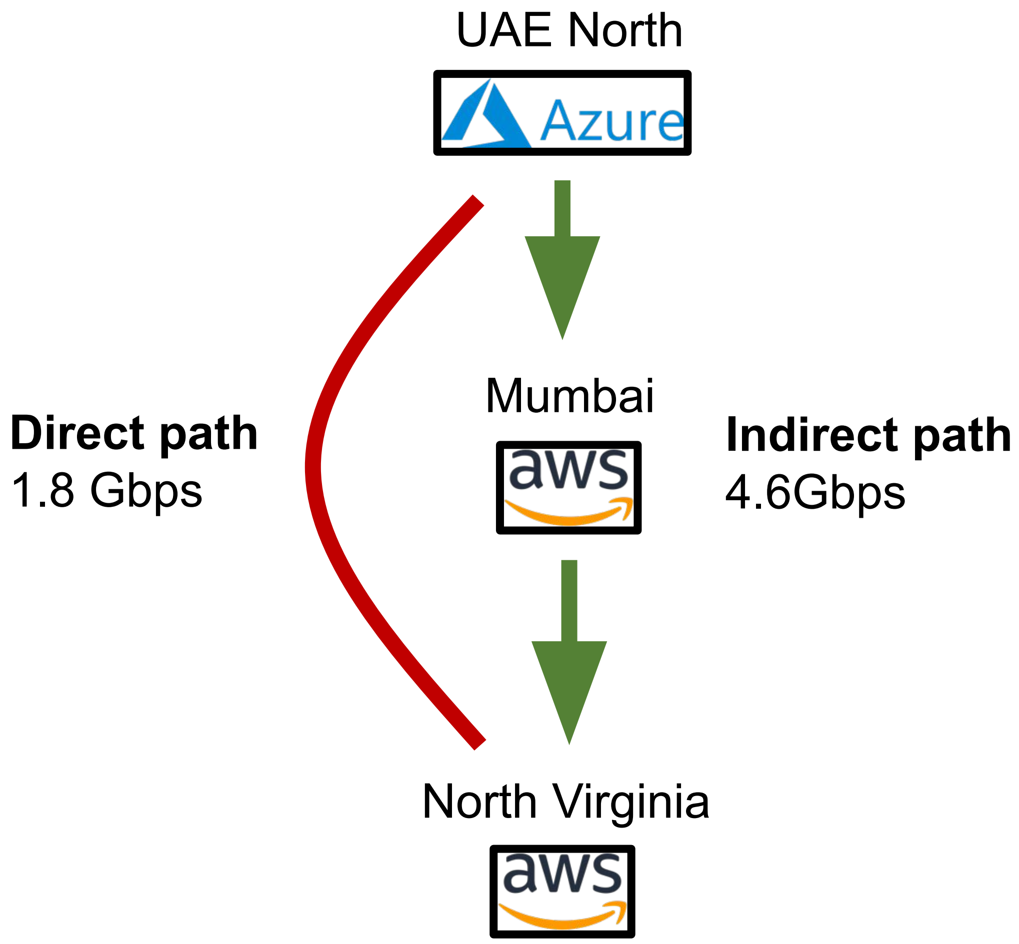
Reasons for slow transfers

1. Congestion along direct path
2. Poor peering between providers
3. Packet loss from the physical layer

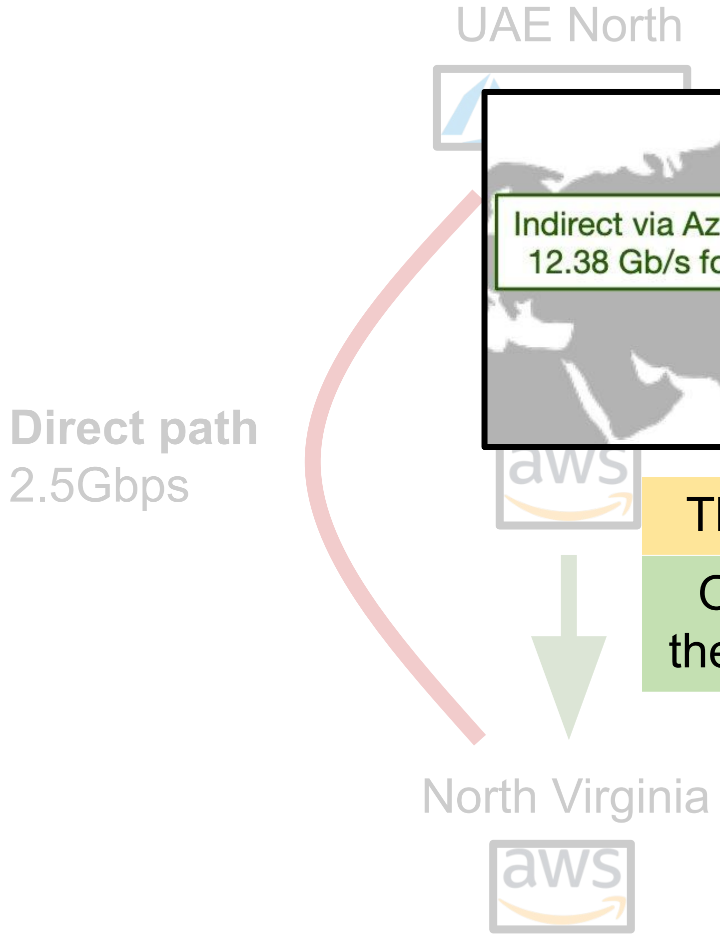
Insight #1: overlay routing to circumvent slow/expensive links

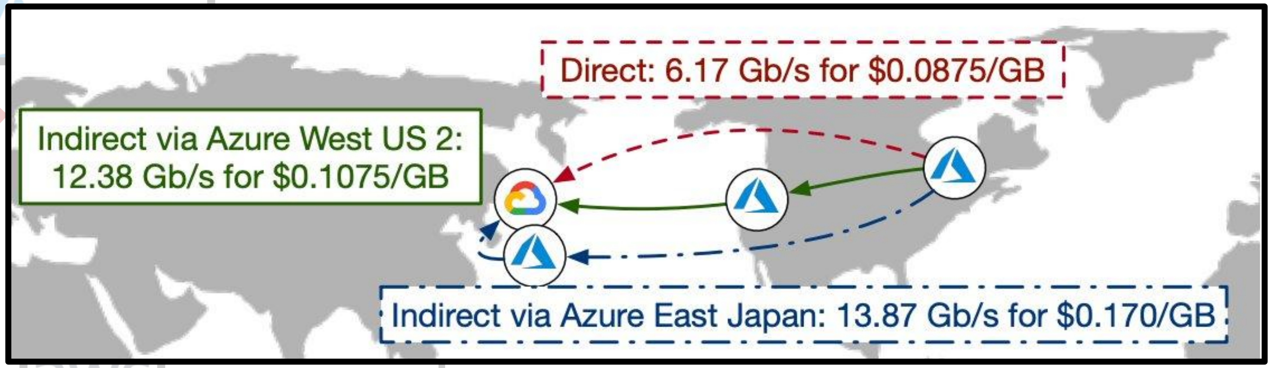


Insight #1: overlay routing to circumvent slow/expensive links



Insight #1: overlay routing to circumvent slow/expensive links

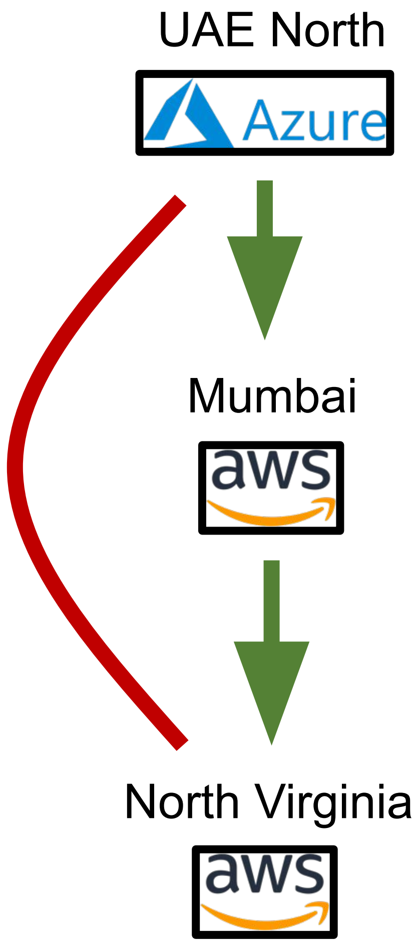




There are $O(2^{200})$ possible routing paths

Our planner searches the full space for the optimal plan in **less than 10 seconds**

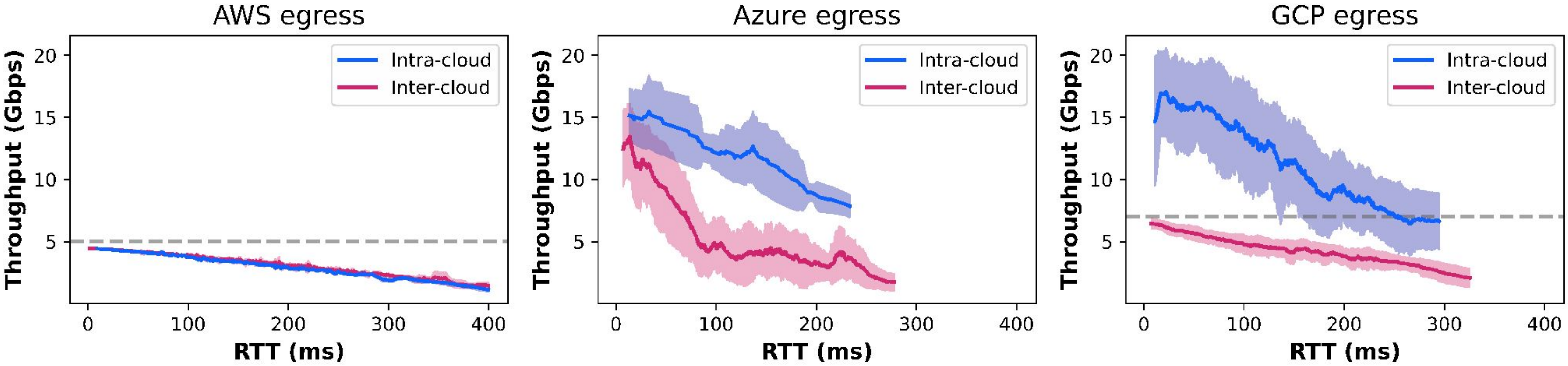
Insight #1: overlay routing to circumvent slow/expensive links



Overlay routing

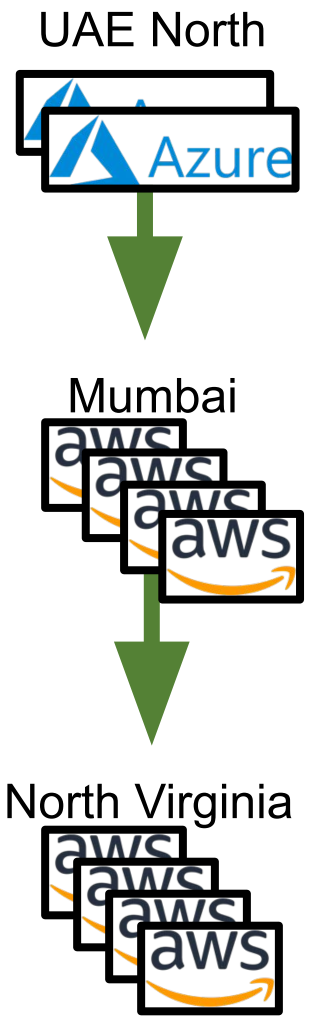
Indirect paths to avoid slow or expensive links

Insight #2: parallel VMs per region to avoid egress limits



Clouds throttle egress speeds!

Insight #2: parallel VMs per region to avoid egress limits



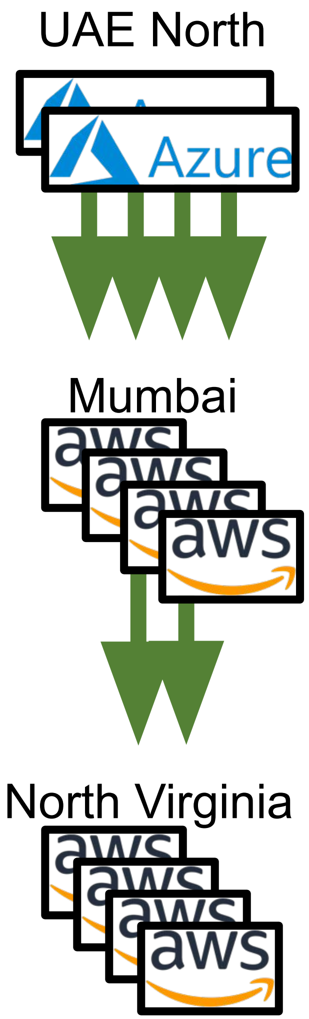
Overlay routing

Indirect paths to avoid slow or expensive links

of VMs per region

Access throughput beyond NIC, AWS and GCP throttle egress

Insight #3: parallel TCP connections to improve goodput



Overlay routing

Indirect paths to avoid slow or expensive links

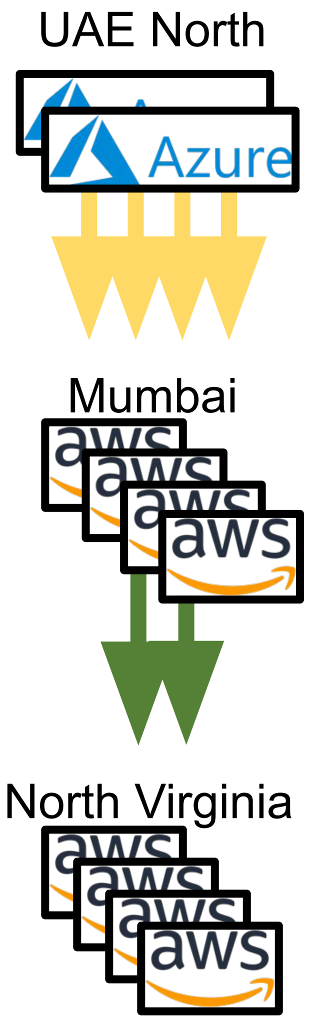
of VMs per region

Access throughput beyond NIC, AWS and GCP throttle egress

of parallel TCP connections

Unlike internet, fairness is a provider-level concern due to egress fees

Insight #4: cut cost with compression + network tiers



Overlay routing

Indirect paths to avoid slow or expensive links

of VMs per region

Access throughput beyond NIC, AWS and GCP throttle egress

of parallel TCP connections

Unlike internet, fairness is a provider-level concern due to egress fees

Network tiering + compression

Hot potato routing up to 40% cheaper than cold potato

All techniques explained in our NSDI 2023 paper

Overlay routing

Indirect paths to avoid slow or expensive links

of VMs per region

Access throughput beyond NIC, AWS and GCP throttle egress

of parallel TCP connections

Unlike internet, fairness is a provider-level concern due to egress fees

Network tiering + compression

Hot potato routing up to 40% cheaper than cold potato

Skyplane: Optimizing Transfer Cost and Throughput Using Cloud-Aware Overlays

Paras Jain, Sam Kumar, Sarah Wooders, Shishir G. Patil, Joseph E. Gonzalez, and Ion Stoica
University of California, Berkeley

Abstract

Cloud applications are increasingly distributing data across multiple regions and cloud providers. Unfortunately, wide-area bulk data transfers are often slow, bottlenecking applications. We demonstrate that it is possible to significantly improve inter-region cloud bulk transfer throughput by adapting network overlays to the cloud setting—that is, by routing data through indirect paths at the application layer. However, directly applying network overlays in this setting can result in unacceptable increases in cloud egress prices. We present Skyplane, a system for bulk data transfer between cloud object stores that uses cloud-aware network overlays to optimally navigate the trade-off between price and performance. Skyplane's planner uses mixed-integer linear programming to determine the optimal overlay path and resource allocation for data transfer, subject to user-provided constraints on price or performance. Skyplane outperforms public cloud transfer services by up to 4.6× for transfers within one cloud and by up to 5.0× across clouds.

1 Introduction

Increasingly, cloud applications transfer data across datacenter boundaries, both across multiple regions within a cloud provider (multi-region) and across multiple cloud providers (multi-cloud). This is in part due to privacy regulations, the availability of specialized hardware, and the desire to prevent vendor lock-in. In a recent survey [26], more than 86% of 727 respondents had adopted a multi-cloud strategy across diverse workloads. Thus, support for fast, cross-cloud bulk transfers is increasingly important.

Applications transfer data between datacenters for batch processing (e.g. ETL [9], Geo-Distributed Analytics [54]), and production serving (e.g. search indices [34]). Extensive prior work optimizes the throughput of bulk data transfers between datacenters within application-defined minimum performance constraints [34, 36, 38, 64]. All major clouds offer services for bulk transfers such as AWS DataSync [5], Azure AzCopy [22], and GCP Storage Transfer Service [31].

From the perspective of a cloud customer, transfer throughput and cost (price) are the two important metrics of transfers in the cloud. Thus we ask *how can we optimize network cost and throughput for cloud bulk transfers?* We study this question in the context of designing and implementing Skyplane, an open-source cloud object transfer system.

A seemingly natural approach is to optimize the routing protocols in cloud providers internal networks to support higher-throughput data transfers. Unfortunately, this is not feasible for two reasons. First, rearchitecting the IP layer routing protocol to optimize for high-throughput bulk transfer could be negatively impact other applications that are sensitive to network latency. Second, cloud providers lack a strong incentive to optimize data transfer to other clouds. Indeed, AWS DataSync [5], AzCopy [22], GCP Storage Transfer [31], AWS Snowball [62], and Azure Data Box Disk [12], all support data transfer *into*, but not out of, their respective clouds. Improvements to cross-cloud peering must be achieved with the cooperation of both the source and destination providers.

Skyplane's key observation is that we can instead identify *overlay paths*—paths that send data via intermediate regions—that are faster than the direct path. The throughput of the direct path from Azure's `Central Canada` region to GCP's `asia-northeast1` region is 6.2 Gbps. Instead, Skyplane can route the transfer via an intermediate stop at Azure's `US West 2` with a throughput of 12.4 Gbps for a 2.0× speedup (Fig. 1). Crucially, this can be implemented on top of the cloud providers' services without their explicit buy-in.

We are not the first to propose the use of overlay networks on the public Internet [8]. However, these techniques ignore two key considerations of public clouds: **price** and **elasticity**.

First, the highest-bandwidth overlay path may have an unacceptably high **price**. Cloud providers charge for data egress separately for each hop along the overlay path. To reduce the cost of the overlay, it is essential to transfer data along cheap paths to trade off price and performance. For example, in Fig. 1, one can achieve 13.9 Gbps by instead using Azure's `East Japan` region as the relay, but the cost would be 1.9× that of transferring data directly. In contrast, using Azure's

2 Background

Network overlays In the early 2000s, network overlays emerged as a technique for *application-level routing* with-

https://paper.skyplane.org

skyplane.org

Sky Storage: Data Availability Across the Sky

Data often must need to be accessed from different cloud **regions** and **providers**:

- Geo-distributed model serving
- Cross-organization dataset sharing
- Cross-cloud applications

Sky Storage: Data Availability Across the Sky

Access data across cloud regions or even cloud providers is **slow** and **expensive**.

- Cross-region egress: **$0.02-0.09/GB**
- Cross-cloud egress: **Up to $0.19/GB**

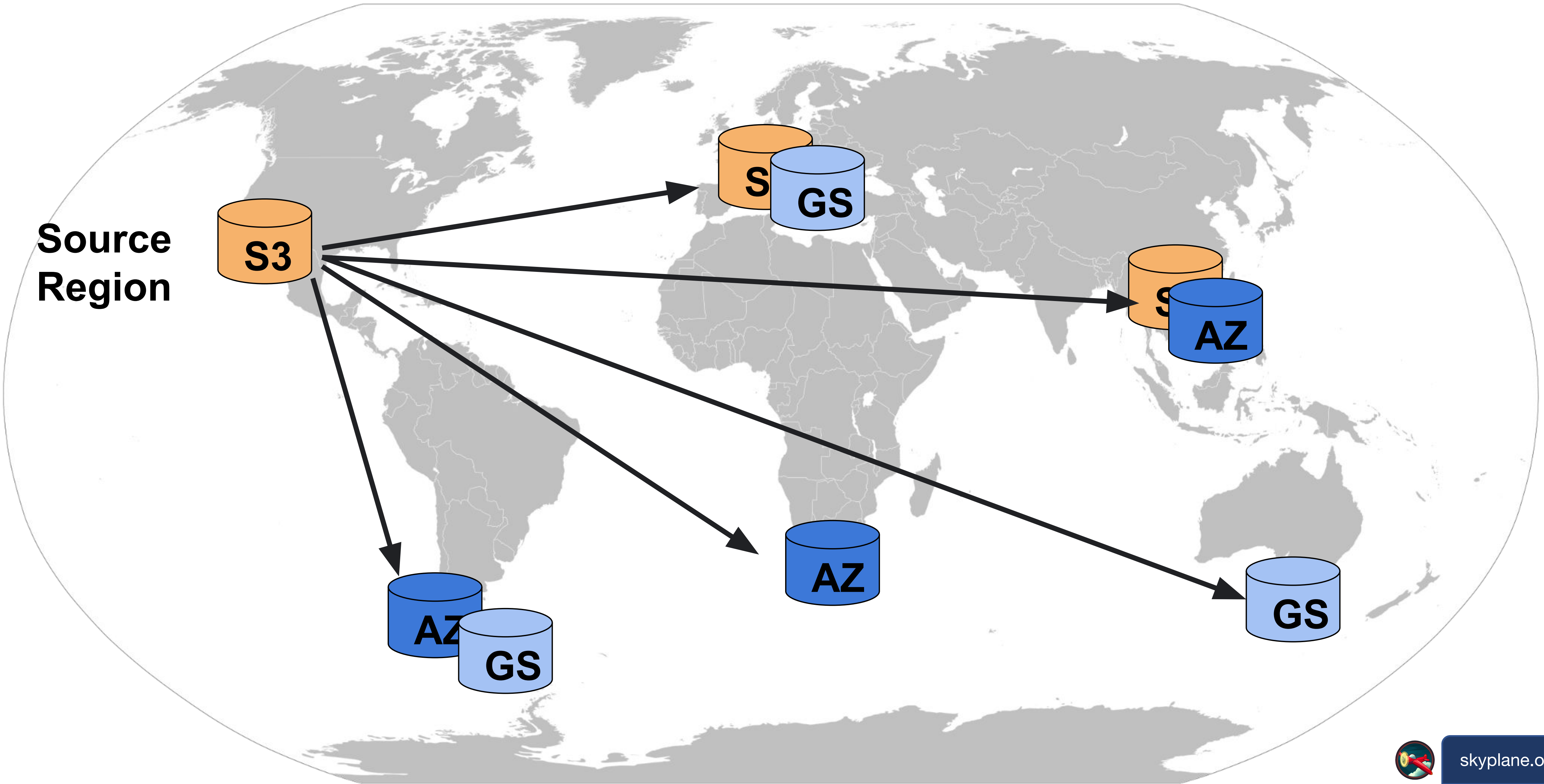
Sky Storage: Data Availability Across the Sky

Access data across cloud regions or even cloud providers is **slow** and **expensive**.

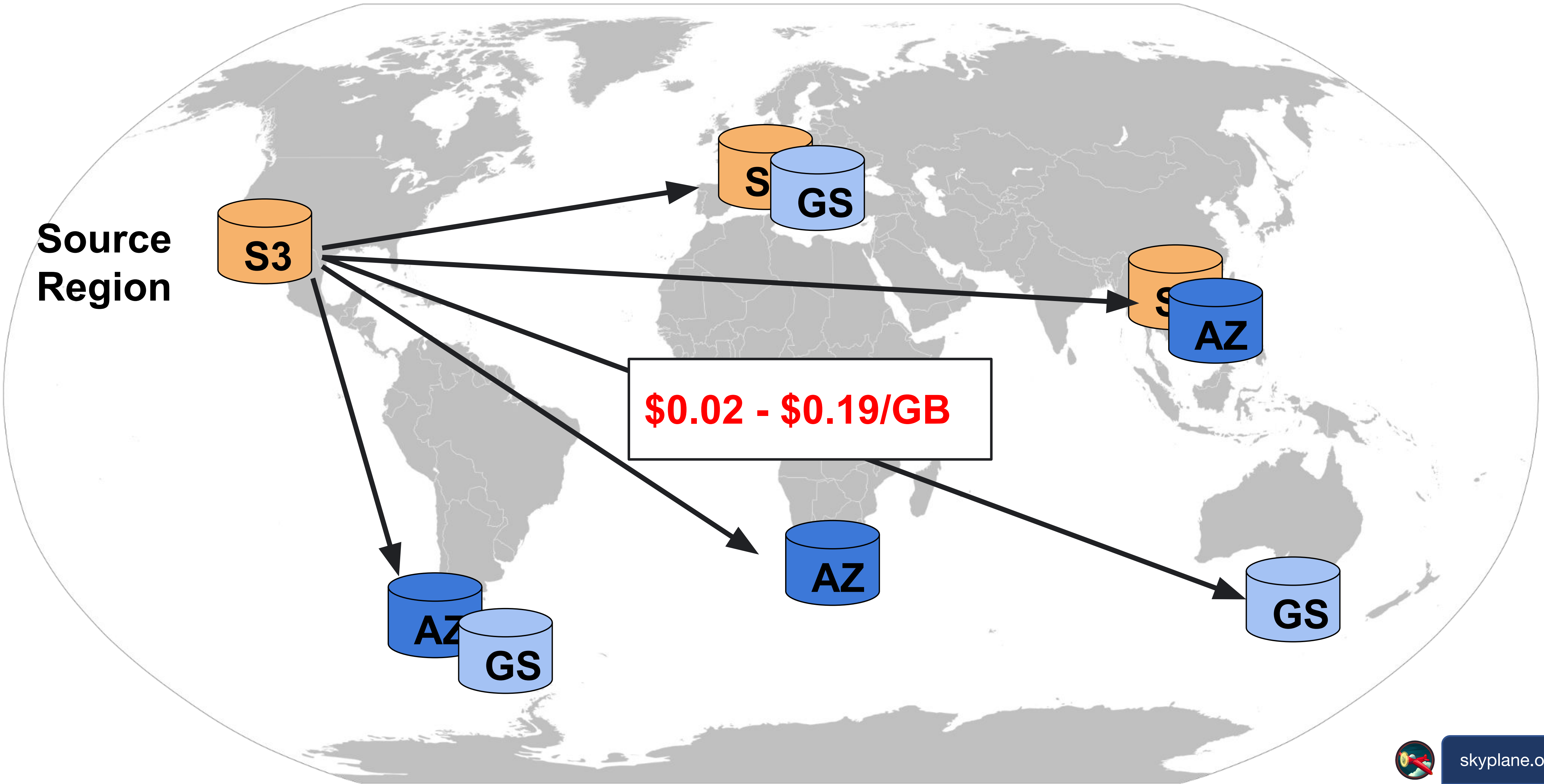
- Cross-region egress: **$0.02-0.09/GB**
- Cross-cloud egress: **Up to $0.19/GB**

For repeatedly accessed data, **data is replicated across regions & providers** to reduce *access latency* and overall *egress cost*.

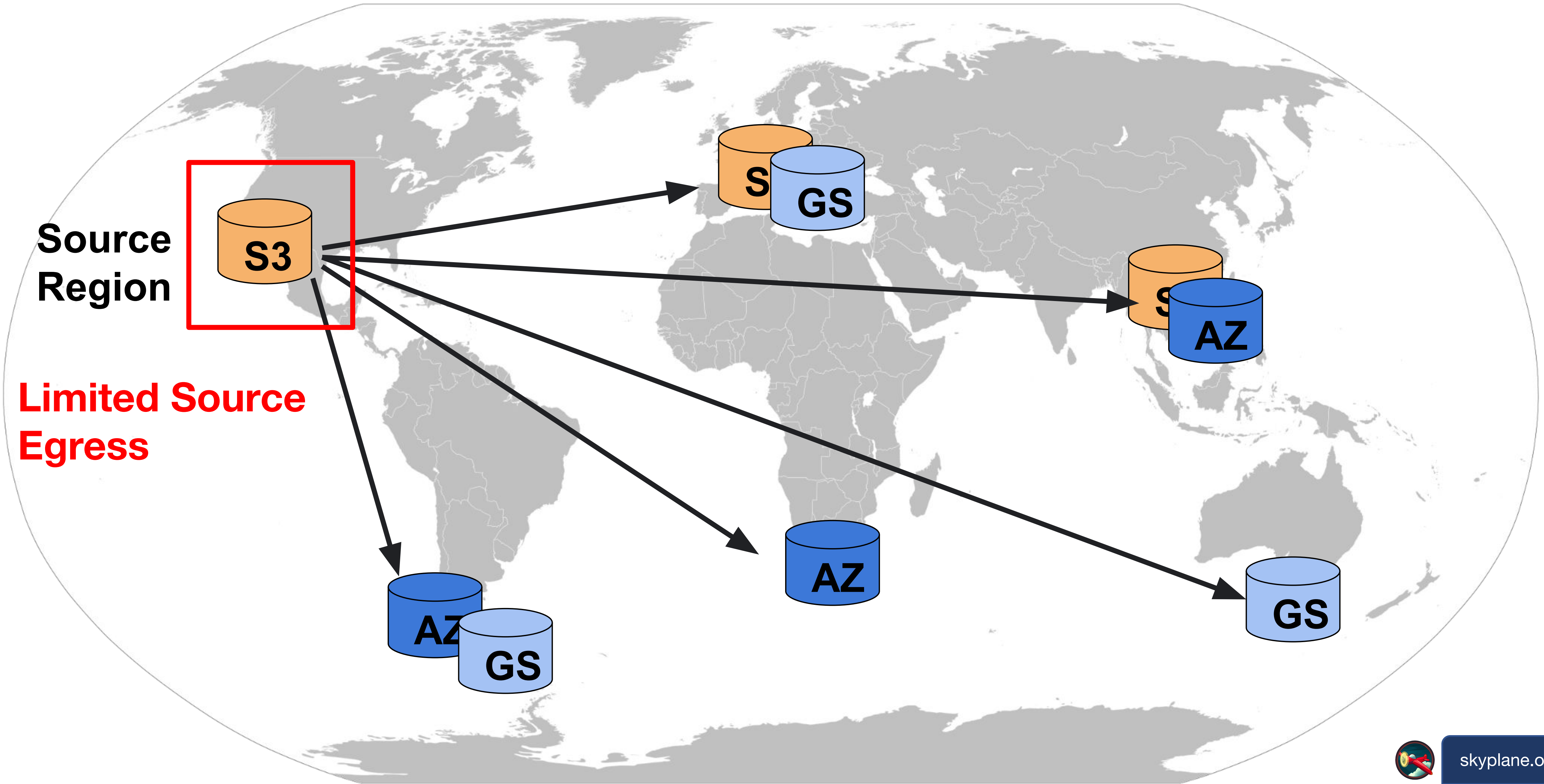
Simple Approach: **Direct Replication**



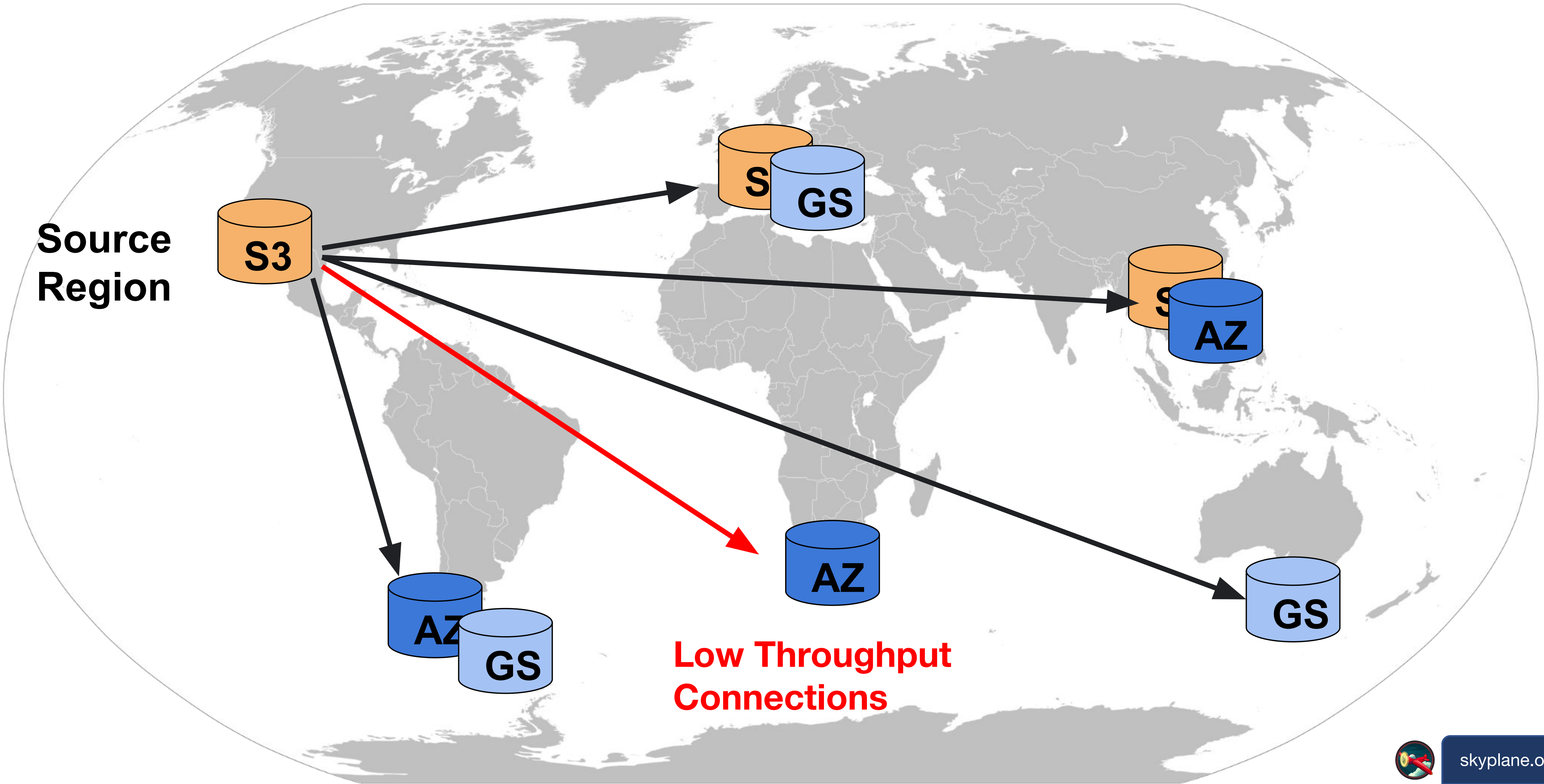
Problem: Replication is expensive



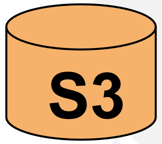
Problem: Bottlenecked replication throughput



Problem: Bottlenecked replication throughput



Goal: Publish data across clouds & regions

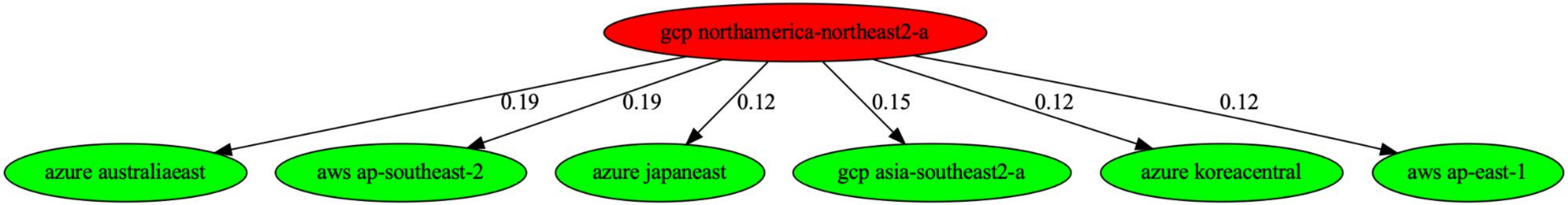


How can we publish data across multiple clouds and regions in a way that is:

- **Fast** (high throughput)
- **Cost effective** (low network costs)

Direct Transfers

Transferring from GCP source region to 6 Azure, GCP, and AWS destinations with **6 Direct Transfers**:



Replication Cost: 0.19 + 0.12 + 0.15 + 0.12 + 0.12 = **$0.90 / GB**

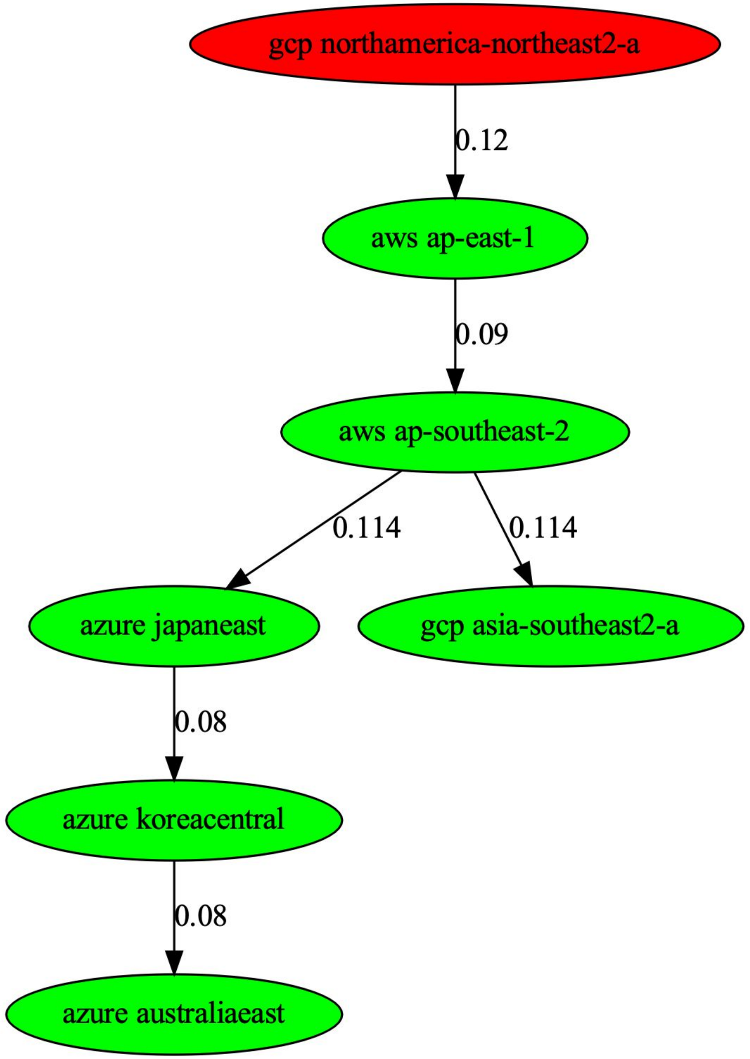
$900 / TB
or
$900,000 / PB

Minimizing Egress Cost: MST

Minimum Spanning Tree (MST): Create a graph that minimizes the total cost of edges but connects all nodes.

<u>Replication Cost:</u> 0.12 + 0.09 + 0.114 + 0.114 + 0.08 + 0.08 = **$0.60 / GB**

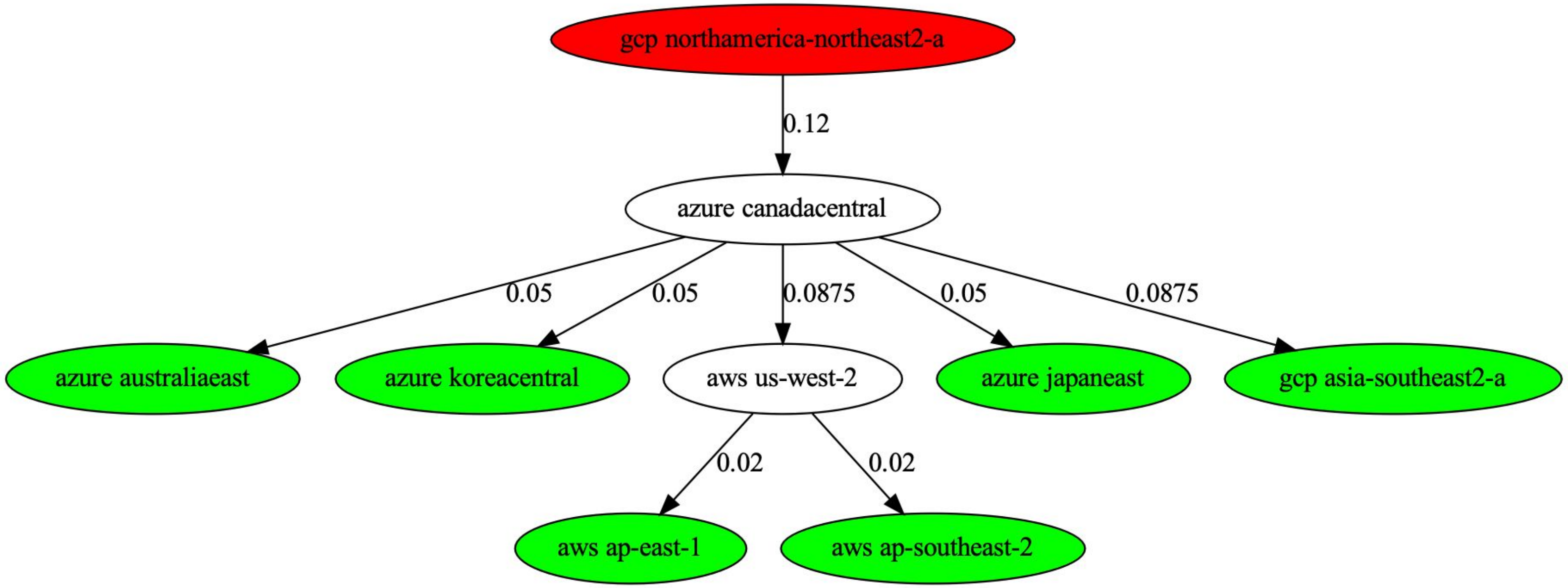
30% Cost Reduction



Minimizing Cost Further

Steiner Tree: MST with the option to use additional *non-destination nodes* (i.e. "overlay" nodes).

Replication Cost: 0.12
+ 0.0875 + 0.05 + 0.05
+ 0.0875 + 0.05 + 0.02
+ 0.02 = **$0.485 / GB**



47% Cost Reduction

0.6X Throughput (compared to MST)

What about throughput?

Ideally, we can minimize cost while meeting some throughput SLO

Solution: Define an **Integer Linear Program** (ILP) which minimizes replication cost given a replication time SLO:

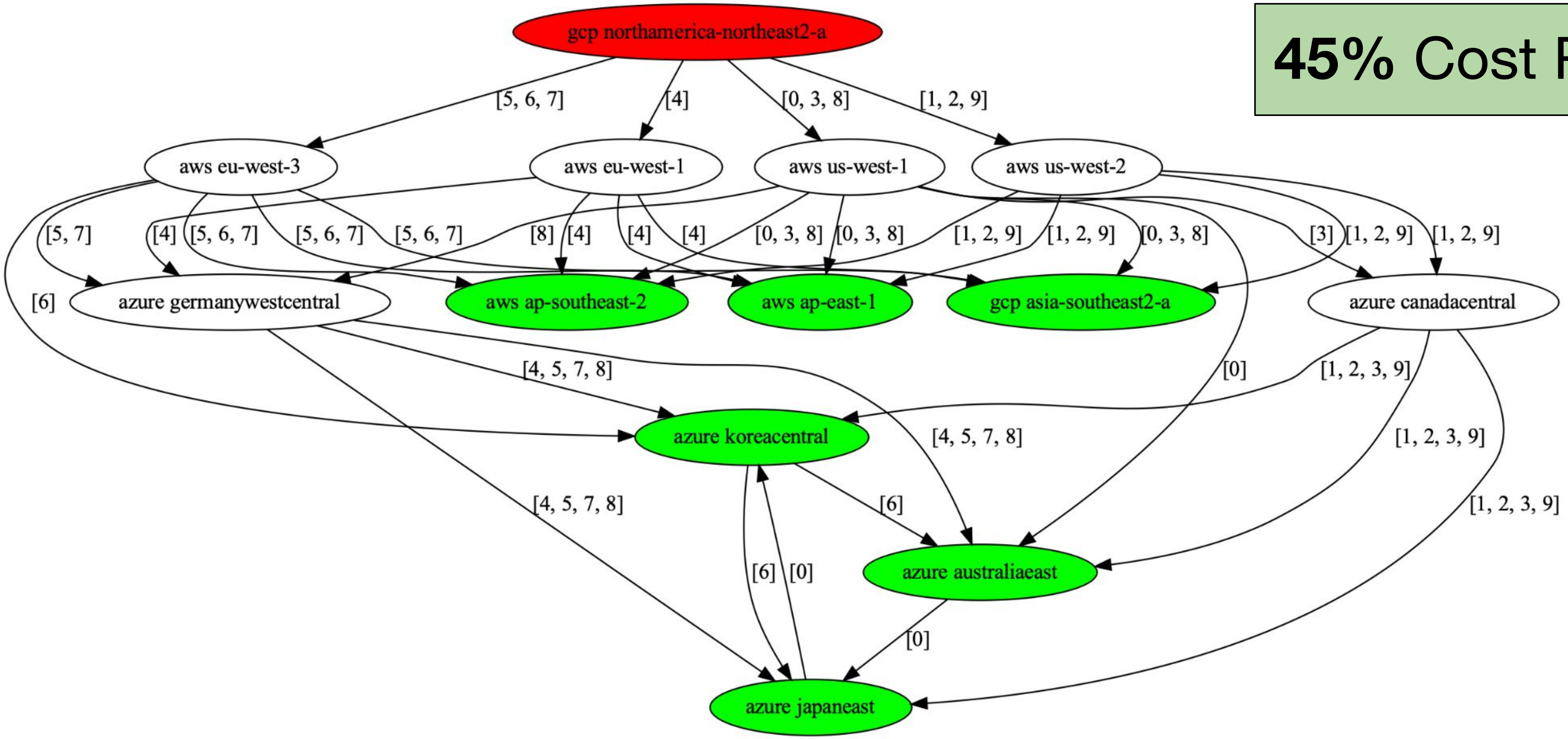
- Topology of replication graph
- # VM instances per region
- Allocation of data across topology

Meeting Throughput Requirements

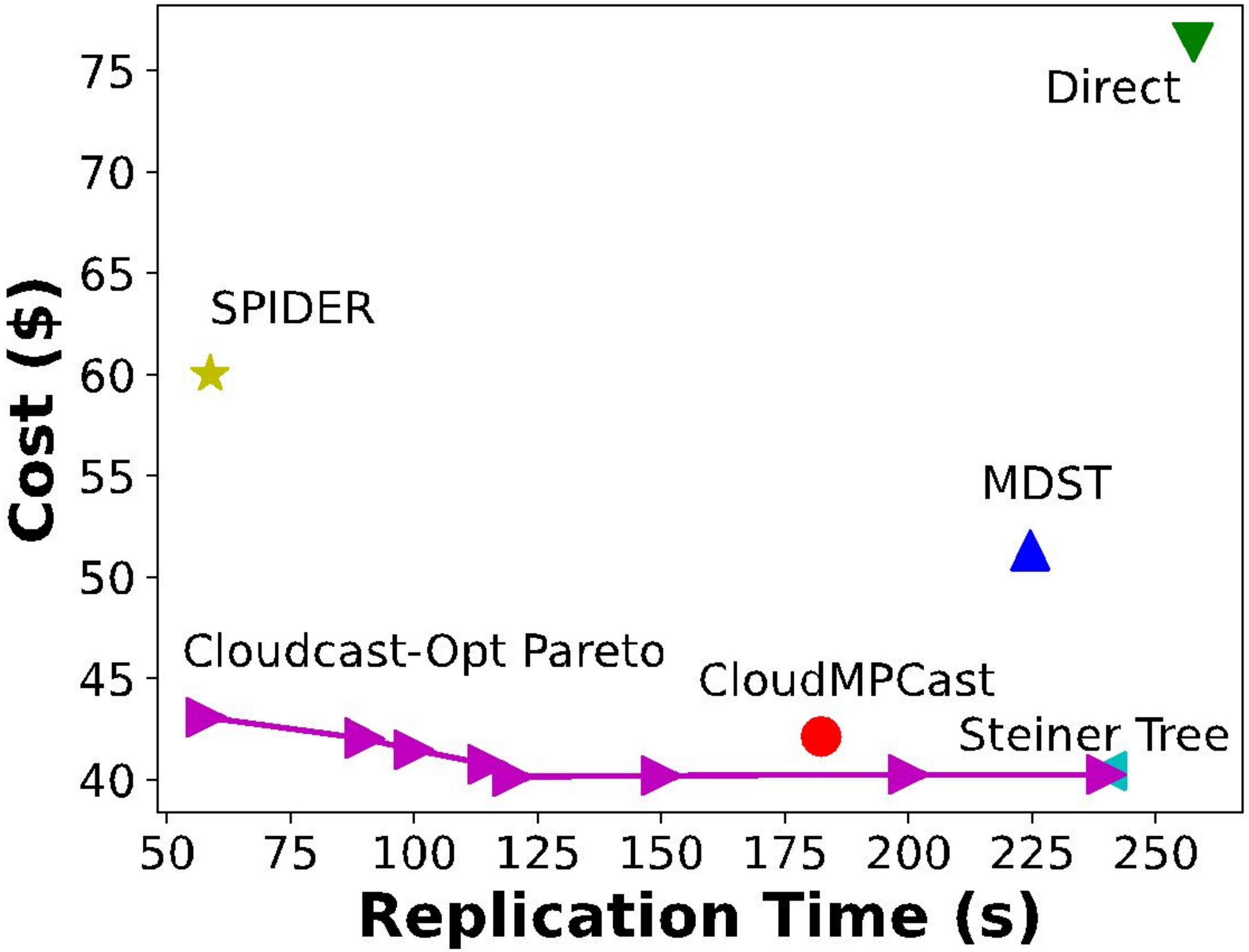
ILP solver output (*10 data partitions*)

6X Throughput

45% Cost Reduction

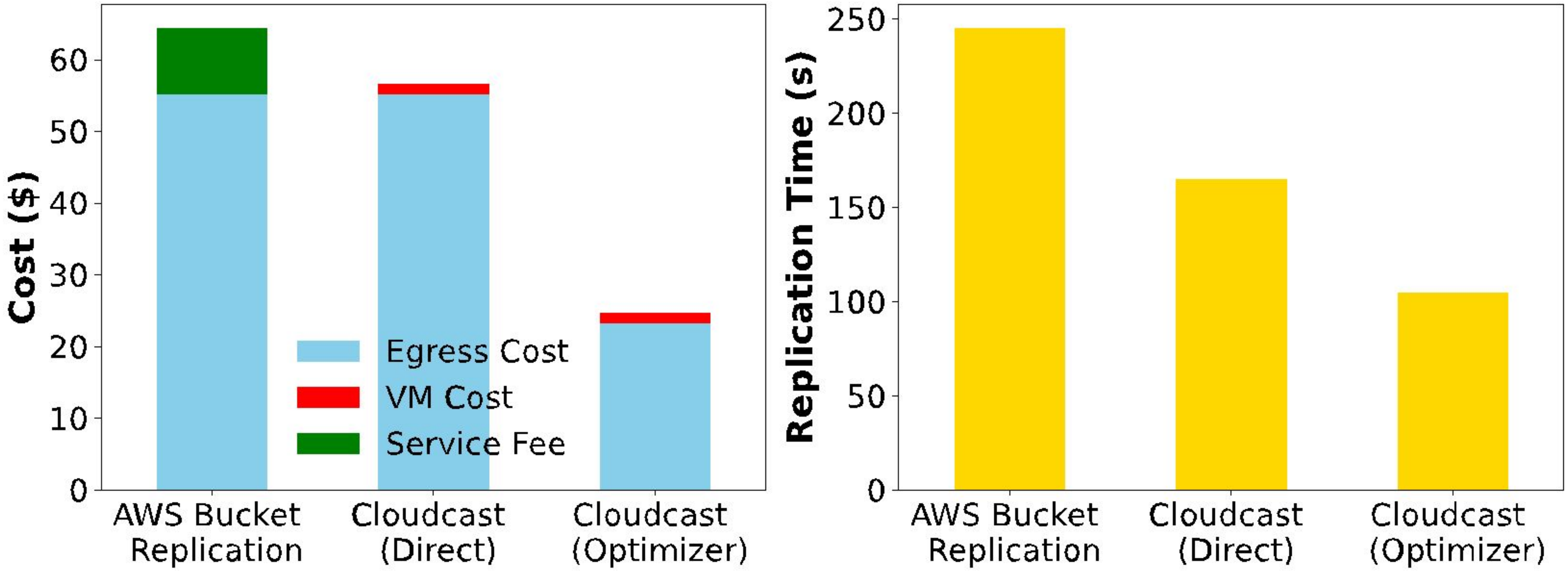


Cost & Throughput Tradeoff Curve



Up to **5X** replication speedup and **53%** cost savings from direct transfers

Comparison to S3 Multi-Bucket Replication



2.3X replication speedup and **61.5%** cost savings

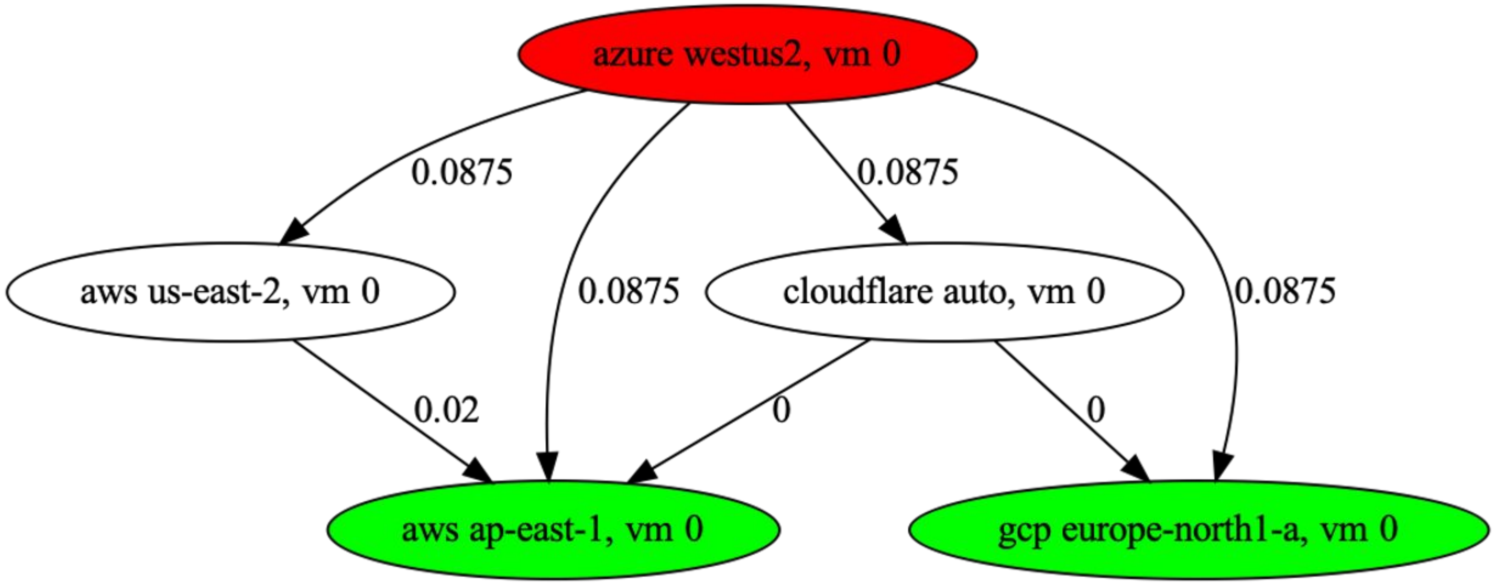
Impact of Incumbent Clouds

Incoming clouds offer reduced or **free egress** (e.g. Cloudflare)

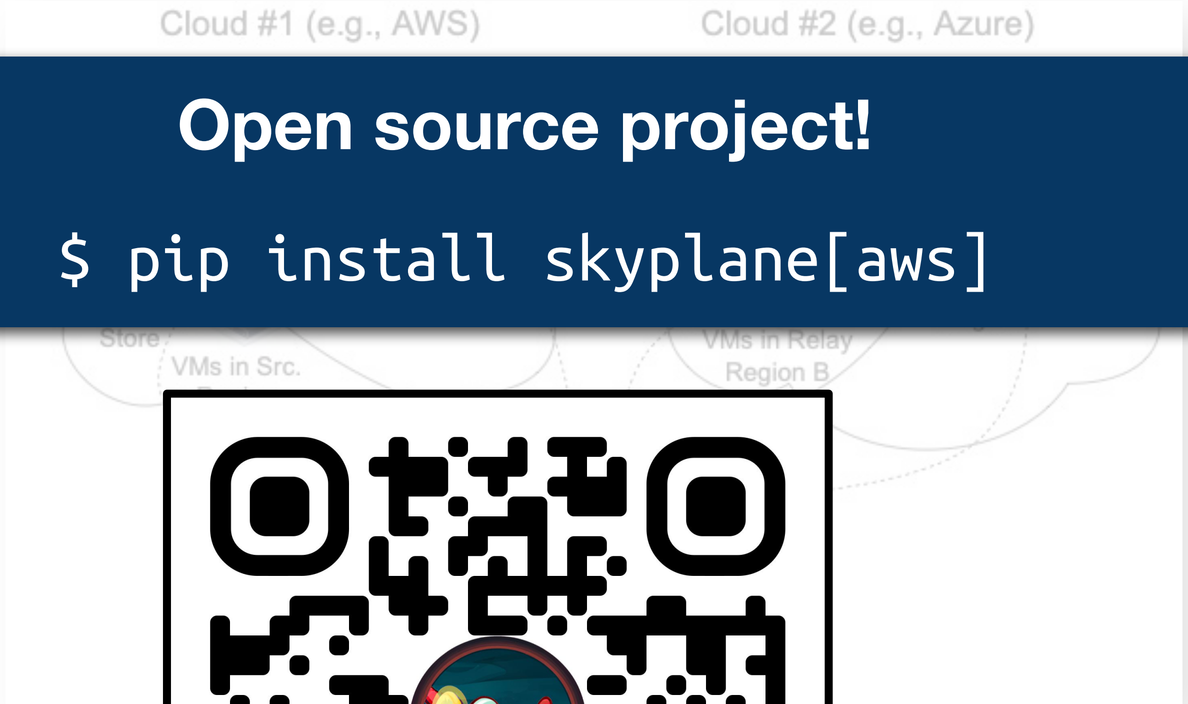
Impact of Incumbent Clouds

Incoming clouds offer reduced or **free egress** (e.g. Cloudflare)

For multi-cloud broadcasts, we can route through free egress clouds to reduce costs *even if the cloud is neither a source/destination*



Skyplane Overlay Network



Open source project!

```
$ pip install skyplane[aws]
```

skyplane.org

No coo… rom clouds!

Skyplane only u… ns in your cloud VPC

Open-source adoption

skyplane-project / **skyplane** Public

Starred 844

Andy Pavlo @andy_pavlo · 23h
This is a clever way of moving dat
reminds me of oldschool parallel
more cloud friendly. Plus they ma

Sourav
@srvmshr
This is so cool. Egress is a pain & Skyplane looks promising for sure. (I guess I might end up using it sometime to move our ~22TB to AWS)

Andrey Cheptsov
Replying to @_parasj
That's amazing! Very cool idea.

Michael Galarnyk @GalarnykMichael · Oct 5
This could be huge for hybrid cloud!

Approaching 1 PiB transferred!

Gridware
Berkeley UNIVERSITY OF CALIFORNIA
BROAD INSTITUTE
NASA EARTHDATA Powered by EOSDIS
VOXEL
Stanford University
ATLASSIAN
kaliberai

Skyplane

Intercloud Broker for Data Apps

Problem: cross-region and cross-cloud transfers are slow and expensive

Skyplane accelerates cloud transfers while reducing egress costs

Open-source tool – please share feedback, use cases or collaborations!

```
$ pip install skyplane[aws,azure,gcp]
$ skyplane init
$ skyplane cp -r s3://… gcs://…
```

skyplane.org

wooders@berkeley.edu

Skyplane team

Shu Liu

Sam Kumar

Sarah Wooders

Paras Jain

Shishir Patil

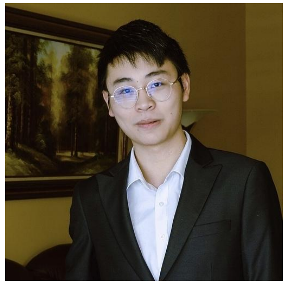
Ion Stoica

Joey Gonzalez

Vincent Liu

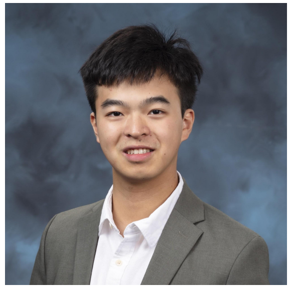
Daniel Kang

Asim Biswal



Jason Ding

Anton Zabreyko



Xuting Liu

Hailey Jang

Simon Mo