

Looking ahead: What comes after Airflow 2.0?

...

Aizhamal Nurmamat kyz
Ash Berlin-Taylor

The Road so Far...

Airflow Timeline



Airflow Today



Active

Top 5 Apache project
repositories by commits
in 2020



Collaborative

+1,500 code authors



Supportive

Continued active
releases for v1 and v2



Popular

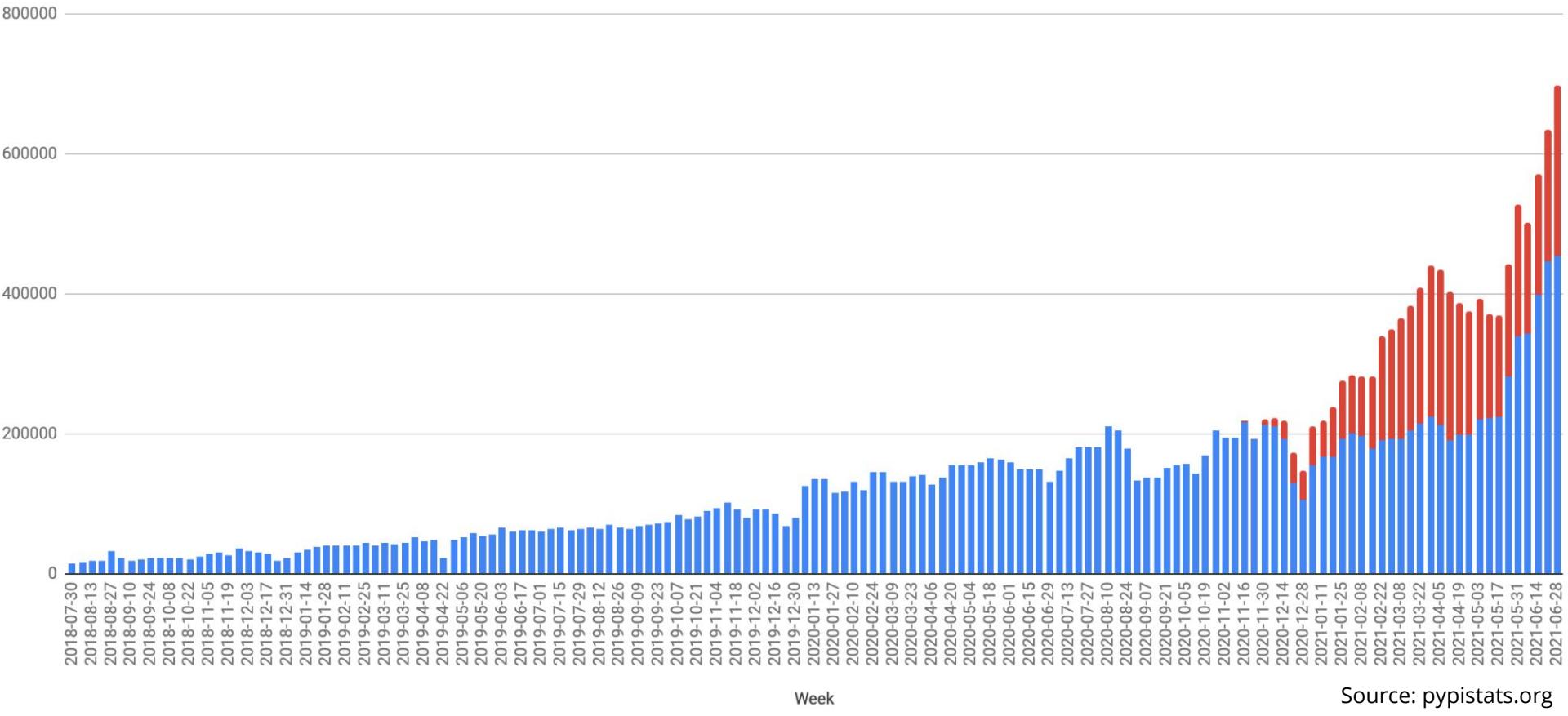
Hundreds of companies
using Airflow in the
wild

Airflow Today



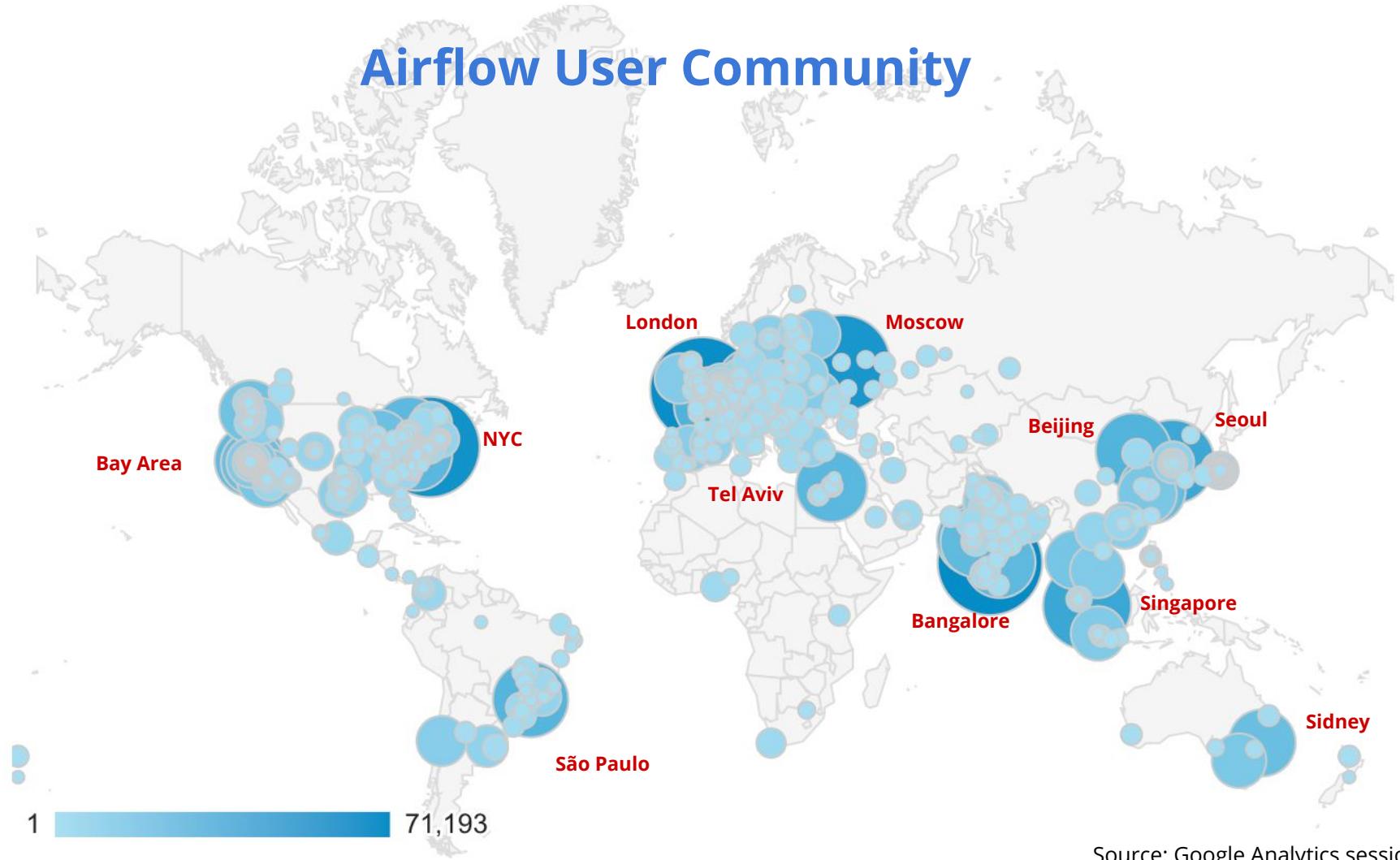
Downloads of Airflow packages per week

v2 v1



Source: pypistats.org

Airflow User Community



Community > Code

Community > Code



Airflow Summit 2020

~6000 registrations

X Sponsors

31 Committers

21 PMC members

Airflow Summit 2021

~9000 registrations

XX sponsors

44 Committers

24 PMC members

Transition

Apache Airflow 2.2 and beyond

We build our computer
(systems) the way we build
our cities: over time, without
a plan, on top of ruins

— Ellen Ulman



A photograph of a person walking away from the camera on a paved path through a dense forest of tall, massive redwood trees. The trees have thick trunks and green foliage. Sunlight filters through the canopy, creating bright rays. The person is wearing a backpack and blue jeans. The path is flanked by wooden railings.

Roadmap: A possible future

Making DAGs a joy to write

Airflow should be the go to
orchestrator for *every* data
workflow job

Airflow should be easier to
operate confidently

Roadmap Concepts

- Making DAGs a joy to write
- Airflow should be the go to orchestrator for every data workflow job
- Airflow should be easier to operate confidently

Airflow 2.2



AIP-39: Run DAGs on customizable schedules

Why AIP-39

AIP-40: Any operator can
"defer" itself

Why AIP-40



The near future

Where's the new UI?

AIP-38

Adding missing API endpoints *first*

Don't want to ask people to not contribute new UI features

Nor do we want to chase a moving target

airflowctl: CLI over the API

Untrusted workers

DAG/task lifecycle hooks and easier notifications

Dynamic DAGs

```
@task
def get_files_from_s3():
    ...
my_files = get_files_from_s3()
s3_delete_files = S3DeleteObjectsOperator.partial(
    aws_conn_id="my-aws-conn-id",
    bucket="my-bucket"
).map(key=my_files)
```

Add screenshot for previous

```
@dag
def my_dag(markets: list[str], campaigns: dict[str, list[int]]):
    @task
    def ingest(market):
        ...
    @task
    def calculate_roi(market, campaign):
        ...
    @task
    def aggregate_rois(market, campaign_rois):
        total = 0
        n = 0
        for campaign_roi in campaign_rois:
            n += 1
            total += campaign_roi
        return campaign_roi / total

    data = ingest.map(markets)
    rois = calculate_roi.map(market, data)
    stats = aggregate_rois(market, rois)
```

A better cross-DAG story

Looking further ahead

A wide-angle photograph of a dark night sky. The horizon is filled with the silhouettes of tall evergreen trees. Above them, the sky is a vibrant display of the Aurora Borealis. On the left, the aurora is a deep red, transitioning through orange and yellow towards the center, where it becomes a bright green. To the right, the green aurora continues across the upper half of the frame, with small white stars visible against the dark background.

Event triggered DAGs

DAG versioning

Easier DAG deployment

Streaming

New concept: a Data object

Better support for Machine
Learning



Airflow 3.0?